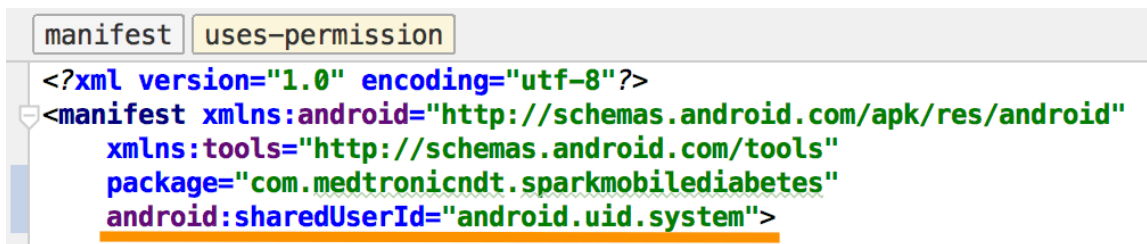# Android Change Locale and Testing with different locale

There are three **red text** in this note. These are ToDo items for future.

1. First of all, to allow app changing the locale, it needs to add the following in manifest file:

   `<uses–permission android:name="android.permission.CHANGE_CONFIGURATION"/>`

2. 因为android.permission.CHANGE_CONFIGURATION属于系统级别的权限，
   普通第三方应用时没有权限使用的，
   即使声明了该权限也无法获取该权限. 需要声明该应用为系统级别:需要在AndroidManifest.xml 添加 android:sharedUserId="android.uid.system".

```
manifest  uses–permission
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    xmlns:tools="http://schemas.android.com/tools"
    package="com.medtronicndt.sparkmobilediabetes"
    android:sharedUserId="android.uid.system">
```

refer : http://www.voidcn.com/article/p-pfxezqvu-bgn.html
refer : https://developer.android.com/guide/topics/manifest/manifest-element.html#uid

2.1 But this "sharedUserId" will make app NOT installable in my test, where the "incompatible" is reported constantly. No solution yet.

The following article suggests to install "system app" to specific folder. **Will try it later.**
http://blog.csdn.net/gengyichao/article/details/6586447

3. Beside above manifest file change, there are other ways to get permission to change locale.

- Root devices;
- Application signed by same key of ROM;
- Application native (inside ROM package);
- Using adb to give access (step-by-step below);

Refer the link https://github.com/jordansilva/Android-ChangeLocaleExample

3.1 I tried the "adb" solution, works:
`adb shell pm grant com.jordan.location.app android.permission.CHANGE_CONFIGURATION`

After running above command line command line, I put breakpoint in the following line, and debug, the checkSelfPermission() return "GRANTED"

```
if (ContextCompat.checkSelfPermission(this,
        Manifest.permission.CHANGE_CONFIGURATION) != PackageManager.PERMISSION_GRANTED) {
```

3.2 After the above permission, **will try launching** app from command line with locale setting. (how to put locale setting in the command line ?)
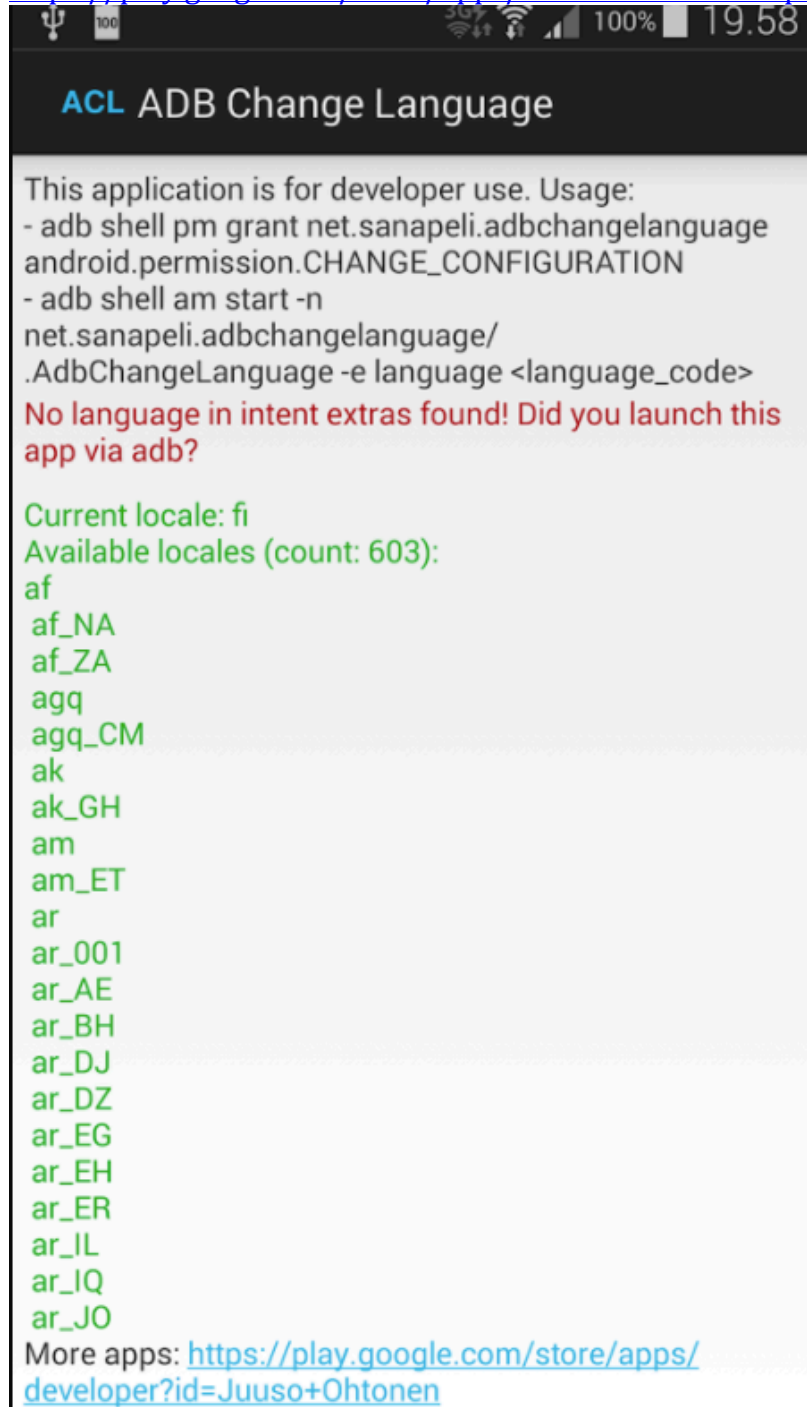

Run activity directly

adb shell am start -n com.package.name/com.package.name.ActivityName

You can also specify actions to be filter by your intent-filters:

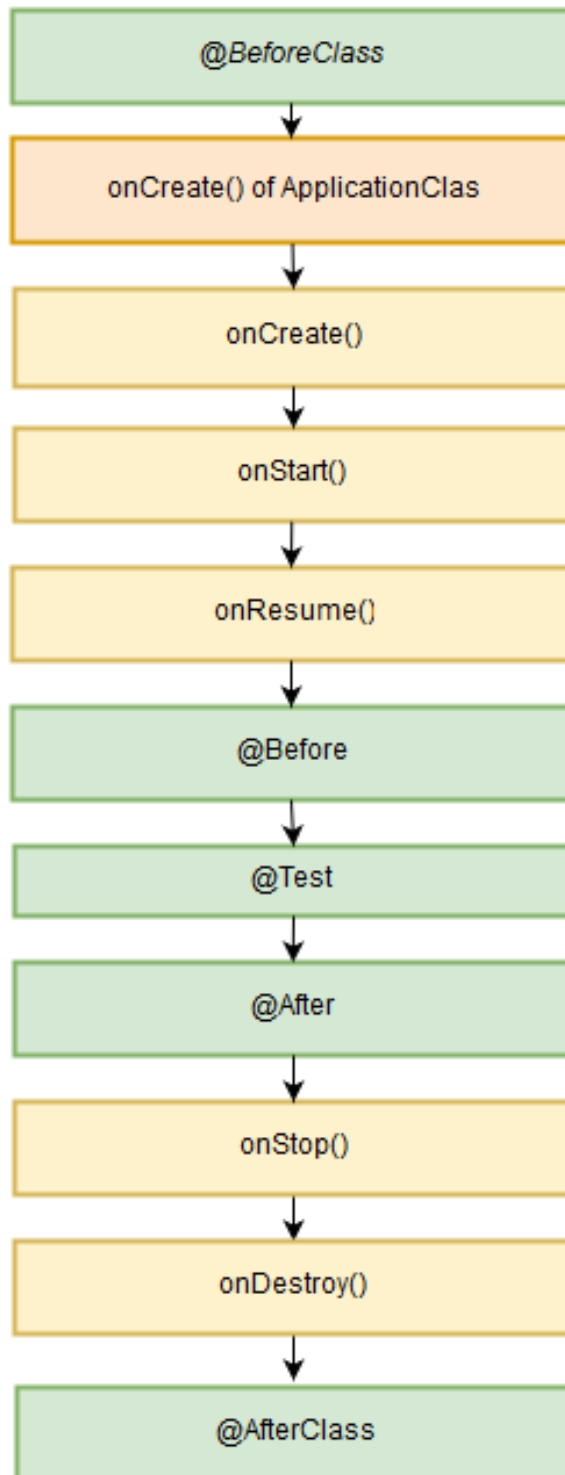am start -a com.example.ACTION_NAME -n com.package.name/com.package.name.ActivityName

---

Also refer
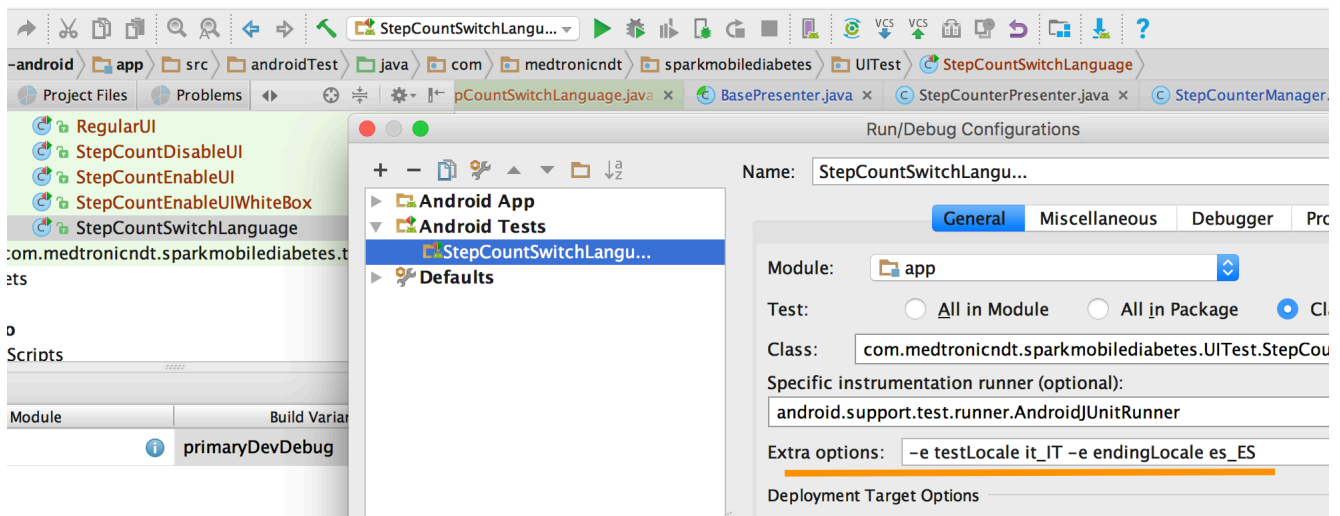https://play.google.com/store/apps/details?id=net.sanapeli.adbchangelanguage&hl=en

4. Run android UI test with espresso

4.1 The espresso runs based on activity. The pre-defined functions are:
   Refer : https://android.jlelse.eu/ui-testing-with-espresso-f5fab5fe3884

```
@BeforeClass
   ↓
onCreate() of ApplicationClas
   ↓
onCreate()
   ↓
onStart()
   ↓
onResume()
   ↓
@Before
   ↓
@Test
   ↓
@After
   ↓
onStop()
   ↓
onDestroy()
   ↓
@AfterClass
```

4.2 When running UITest, it can add command line arguments from UI:

4.3 When running the UI test with fastlane locale setting, the test will fail if run from android studio IDE. The test will pass if running from command line.

Run failed in IDE:

```
D/TestExecutor: Adding listener android.support.test.internal.runner.listener.ActivityFinisherRunListener
I/TestRunner: run started: 1 tests
E/LocaleUtil: Failed to change device locale to de_DE
        java.lang.reflect.InvocationTargetException <1 internal calls>
            at tools.fastlane.screengrab.locale.LocaleUtil.changeDeviceLocaleTo(LocaleUtil.java:41)
            at tools.fastlane.screengrab.locale.LocaleTestRule$1.evaluate(LocaleTestRule.java:30)
            at org.junit.rules.RunRules.evaluate(RunRules.java:20) <11 internal calls>
            at android.support.test.internal.runner.TestExecutor.execute(TestExecutor.java:59)
            at android.support.test.runner.AndroidJUnitRunner.onStart(AndroidJUnitRunner.java:262)
            at android.app.Instrumentation$InstrumentationThread.run(Instrumentation.java:1959)
        Caused by: java.lang.SecurityException: Permission Denial: updateConfiguration() from pid=21
            at android.os.Parcel.readException(Parcel.java:1620)
            at android.os.Parcel.readException(Parcel.java:1573)
            at android.app.ActivityManagerProxy.updateConfiguration(ActivityManagerNative.java:4728)
```

Run passed in command line:

```
[22:07:24]: ▶ 23
[22:07:24]: Granting the permissions necessary to access device external storage
[22:07:24]: $ adb -s 98866733584b573750 shell pm grant com.medtronicndt.envisionp
[22:07:24]: $ adb -s 98866733584b573750 shell pm grant com.medtronicndt.envisionp
[22:07:24]: Running tests for locale: en-US
[22:07:24]: $ adb -s 98866733584b573750 shell am instrument --no-window-animation
-e testLocale en_US \
-e endingLocale en_US \
com.medtronicndt.envisionpro.test/android.support.test.runner.AndroidJUnitRunner
[22:07:51]: ▶ com.medtronicndt.sparkmobilediabetes.UITestSuite.LoginUITest:.
[22:07:51]: ▶ Time: 21,728
[22:07:51]: ▶ OK (1 test)
[22:07:51]: Pulling captured screenshots from the device
[22:07:51]: Make sure you've used Screengrab.screenshot() in your tests and that
```

From the log, it seems the fastline uses reflection to change the locale.

4.4

To save screenshot with espresso, it also needs to (A) declare the read/write external storage permission in manifest file (B) pop up the dialog to let user confirm (> android 6.0)

Other Notes:

1. Offical document for test command arguments:
   https://developer.android.com/studio/test/command-line.html#AMOptionsSyntax

e.g. :
am instrument –w –r –e xxxxx

2. Run espresso UI test from command line. (Note: the 'orange' text for locale setting doesn't work.)

   adb shell am instrument -w -r **-e testLocale it_IT -e endingLocale es_ES** -e debug false -e class com.medtronicndt.sparkmobilediabetes.UITest.StepCountSwitchLanguage com.medtronicndt.envisionpro.test/android.support.test.runner.AndroidJUnitRunner

3. Two ways to set locale under fastlane tests:
   3.1 https://docs.fastlane.tools/actions/screengrab/
   ```
   locales ['en-US', 'fr-FR', 'it-IT']
   ```

   3.2 use @ClassRule

   ```
   https://docs.fastlane.tools/getting-started/android/screenshots/#configuring-your-ui-tests-for-screengrab
   ```

   ```
   @ClassRule public static final LocaleTestRule localeTestRule = new
   LocaleTestRule(Locale.GERMANY, Locale.ENGLISH);
   ```

4. It seems fastlane can NOT capture the pop up alert/dialog. Espresso works in this point.
   https://github.com/fastlane-old/screengrab/issues/90

5. It seems UiAutomator framework can do screenshot and language switch together well.
   Give it a try.
   http://flavienlaurent.com/blog/2014/12/05/screenshot_automation/

6. To make app change language at beginning, the app needs to switch resource and re-start the activity, which has UI to display.
   ```
   http://blog.csdn.net/android_it/article/details/51828456
   ```