

# CS 473 - MDP

## Mobile Device Programming

© 2021 Maharishi International University

**All course materials are copyright protected by international copyright laws and remain the property of the Maharishi International University. The materials are accessible only for the personal use of students enrolled in this course and only for the duration of the course. Any copying and distributing are not allowed and subject to legal action.**



Maharishi International  
University

# CS 473 - MDP

## Mobile Device Programming

MS.CS Program  
Department of Computer Science  
**Renuka Mohanraj , Ph.D.**



Maharishi International  
University

# CS 473 – MDP

## Mobile Device Programming

### Lesson 12

# Localization



Maharishi International  
University

# Wholeness of the lesson

An android application can run on many devices in many different regions. In order to make your application more interactive, your application should handle text, numbers, files etc., in ways appropriate to the locales where your application will be used. *Similarly in Cosmic consciousness, established in inner fullness like localization in android, one acts to fulfill the needs of others and society with greater effectiveness.*

# Localization

- Localization is the process of making an app available in multiple languages.
- To sell an application worldwide successfully, you must target it to a broad and diverse audience.
- English enjoys a broad acceptance and practice, and therefore you can expect that an English language application will sell well globally, but why give up on sales opportunities to the broader non-English-speaking Android market?
- In its simplest approach, this means translation into multiple languages. But there's more! Beyond language translation, an application needs to properly handle dates and times, number and currency formats, and for some applications unit of measure.
- The reasons for localizing an application are manifold. Your application may be bound for some cultural reasons to a specific region.

# String resource

- Keeping your labels and other bits of text outside the main source code of your application is generally considered to be a very good idea. [strings.xml]
- It helps with Internationalization (I18N) and localization (L10N).
- Even if you are not going to translate your strings to other languages, it is easier to make corrections if all the strings are in one spot instead of scattered throughout your source code.

# Android Way

- The way Android currently handles this is by having multiple resource directories, with the criteria for each embedded in their names.
- Suppose, for example, you want to support strings in both English and Spanish.
- Normally, for a single-language setup, you would put your strings in a file named `res/values/strings.xml`.
- To support both English and Spanish, you would create two folders, `res/values-en/` and `res/values-es/`, where the value after the hyphen is the ISO 639–1 two-letter code for the language you want.
- Your English-language strings would go in `res/values-en/strings.xml` and the Spanish ones in `res/values-es/strings.xml`. Android will choose the proper file based on the user's device settings.

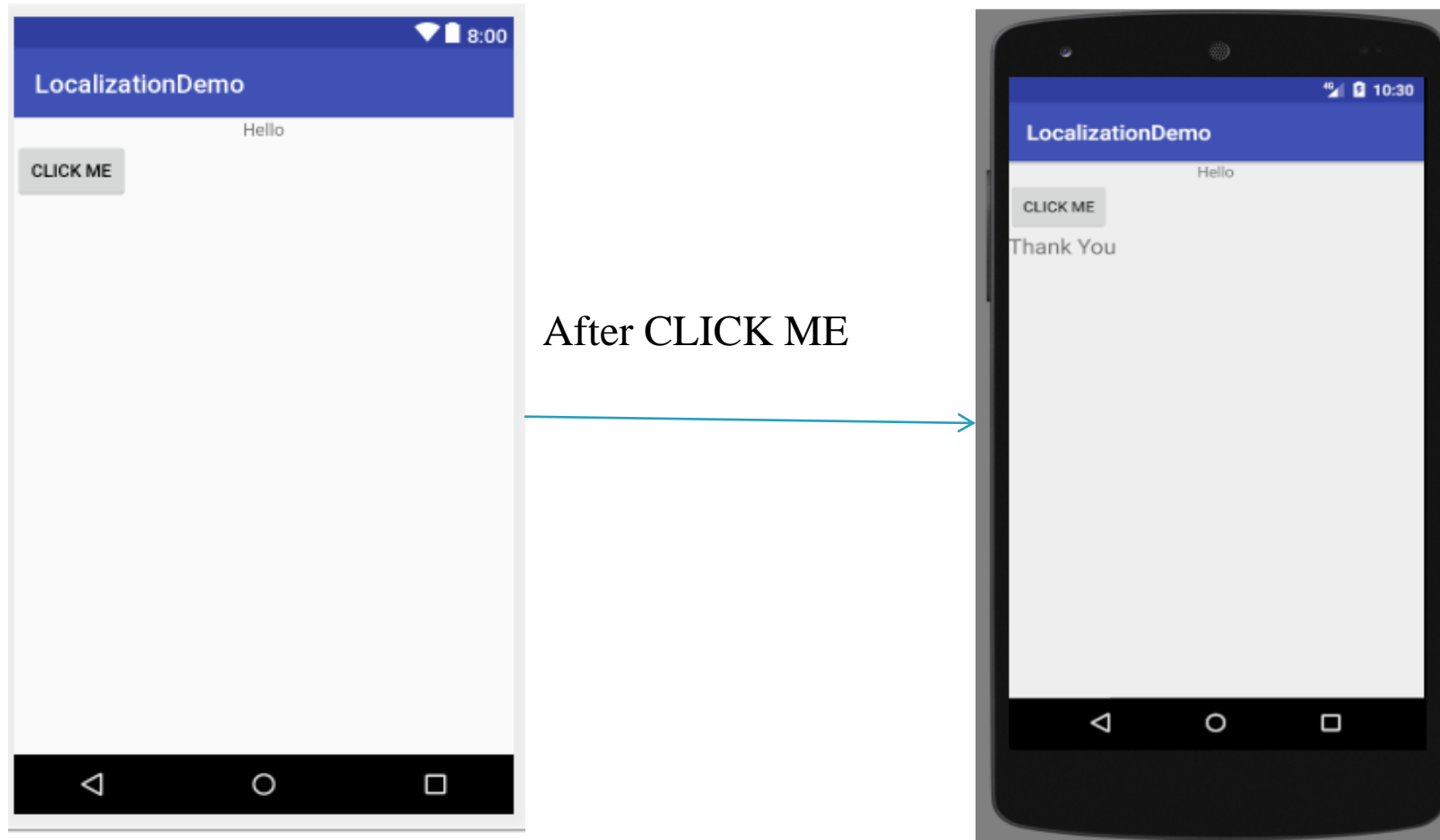
# Android Way

- An even better approach is for you to consider some language to be your default and put those strings in `res/values/strings.xml`.
- Then, create other resource directories for your translations (e.g., `res/values-es/strings.xml` for Spanish).
- Android will try to match a specific language set of resources; failing that, it will fall back to the default of `res/values/strings.xml`.
- This way, if your app winds up on a device with a language that you do not expect, you at least serve up strings in your chosen default language.
- Otherwise, if there is no such default, you will wind up with a `ResourceNotFoundException`, and your application will crash.



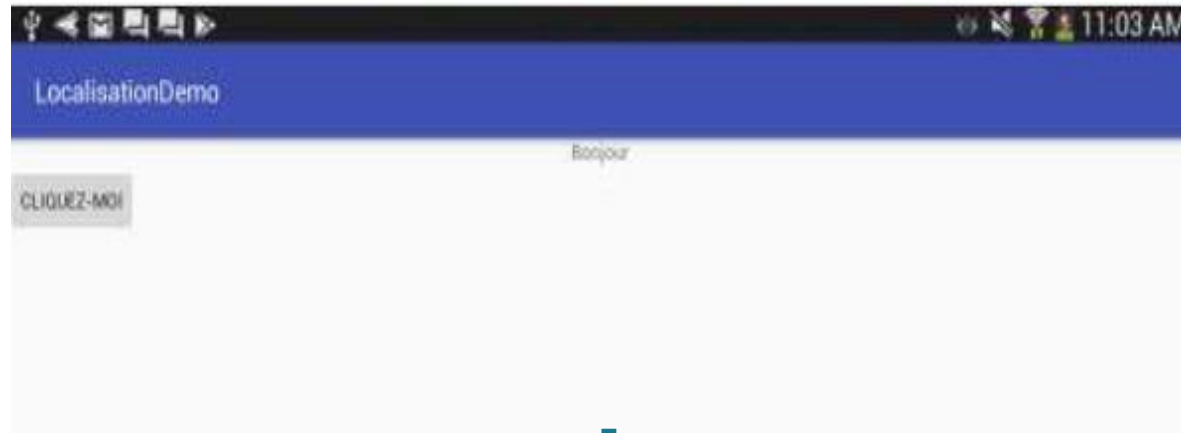
# Hands on Example(English)

This for the default Language(English), you have store everything in String Resources. To make localization here we use French.



# Hands on Example(France)

Once you changed the Language settings to France, you will get the following output



# activity\_main.xml

The screenshot displays the Android Studio IDE with the following components:

- Project Structure:** The left sidebar shows the project hierarchy for 'LocalizationDemo'. The 'app' folder contains 'manifests', 'java' (with 'com.example.rmohanraj.MainActivity'), 'res' (with 'drawable', 'layout', 'activity\_main.xml', 'mipmap', and 'values'), and 'Gradle Scripts'.
- Code Editor:** The main window shows the 'activity\_main.xml' file in the 'Text' tab. The XML code is as follows:

```
1 <?xml version="1.0" encoding="utf-8" ?>
2 <LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
3     android:layout_width="fill_parent"
4     android:layout_height="fill_parent"
5     android:orientation="vertical" >
6     <TextView
7         android:layout_width="fill_parent"
8         android:layout_height="wrap_content"
9         android:gravity="center"
10        android:text="Hello" />
11    <Button
12        android:id="@+id/button1"
13        android:layout_width="wrap_content"
14        android:layout_height="wrap_content"
15        android:onClick="displayResponse"
16        android:text="Click Me" />
17    <TextView
18        android:id="@+id/textView1"
19        android:layout_width="match_parent"
20        android:layout_height="wrap_content"
21        android:text=""
22        android:textSize="20sp" />
23 </LinearLayout>
```
- UI Preview:** On the right, a preview of the app's interface is shown. It features a blue header bar with the title 'LocalizationDemo' and a status bar at the top showing the time '8:00'. The main content area displays the text 'Hello' and a button labeled 'CLICK ME'. The bottom of the preview shows the standard Android navigation bar.
- Bottom Bar:** The bottom of the IDE contains a toolbar with icons for 'Logcat', 'Android Profiler', 'Run', 'TODO', 'Terminal', and 'Messages'.

# String Resources

## **strings.xml**

```
<resources>
  <string name="app_name">LocalizationDemo</string>
  <string name="title">Localization</string>
  <string name="hello">Hello </string>
  <string name="button"> Click Me</string>
  <string name="response">Thank You</string>
</resources>
```

## **strings.xml(fr)**

```
<resources>
  <string name="app_name">LocalisationDemo</string>
  <string name="title">Localisation</string>
  <string name="hello">Bonjour </string>
  <string name="button"> Cliquez-moi</string>
  <string name="response">Je vous remercie</string>
</resources>
```

# MainActivity.java

```
public class MainActivity extends AppCompatActivity {  
    @Override  
    protected void onCreate(Bundle savedInstanceState) {  
        super.onCreate(savedInstanceState);  
        setContentView(R.layout.activity_main);  
    }  
    public void displayResponse(View v) {  
        textView1.setText(R.string.response);  
    }  
}
```

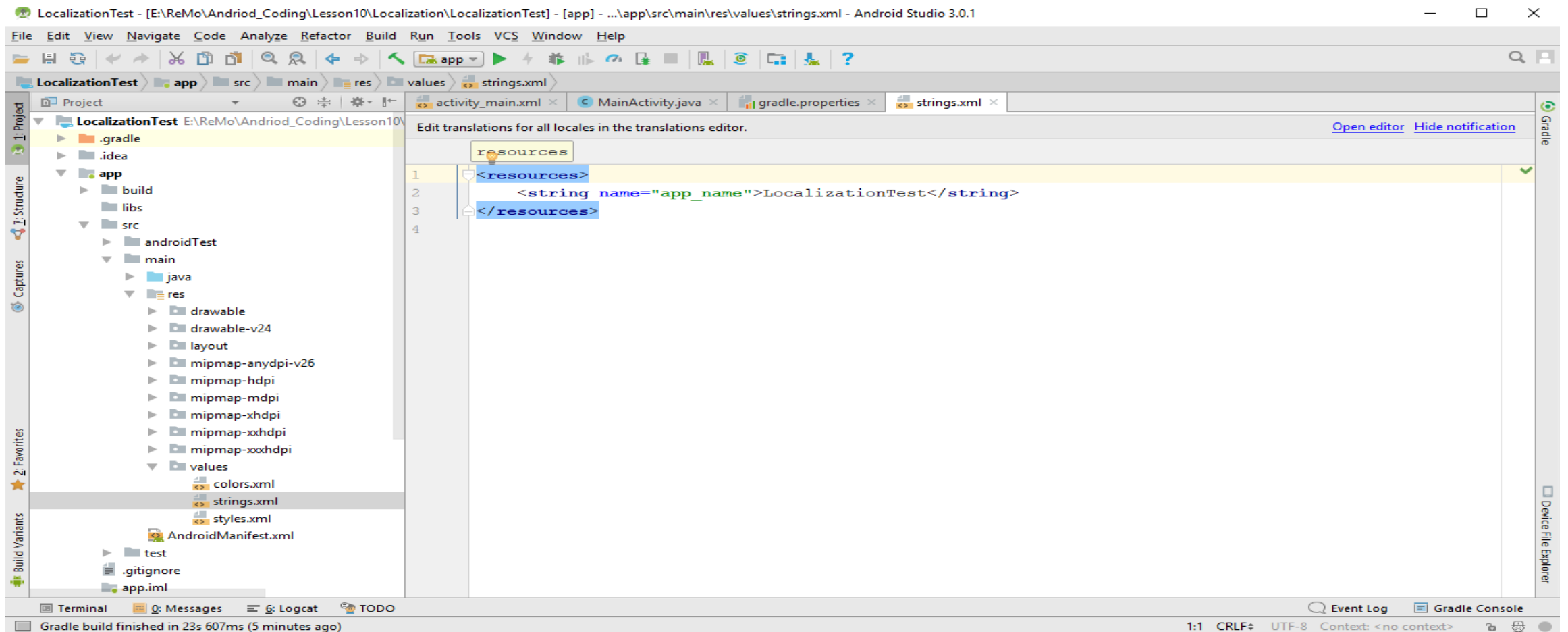
Demo Code: DaysApp, LocalizationDynammic

# Main Point 1

Localization is the process of making an app available in multiple languages for the benefit of everyone. *Science of Consciousness: In the state of refined cosmic consciousness, flow of universal love for everyone and everything*

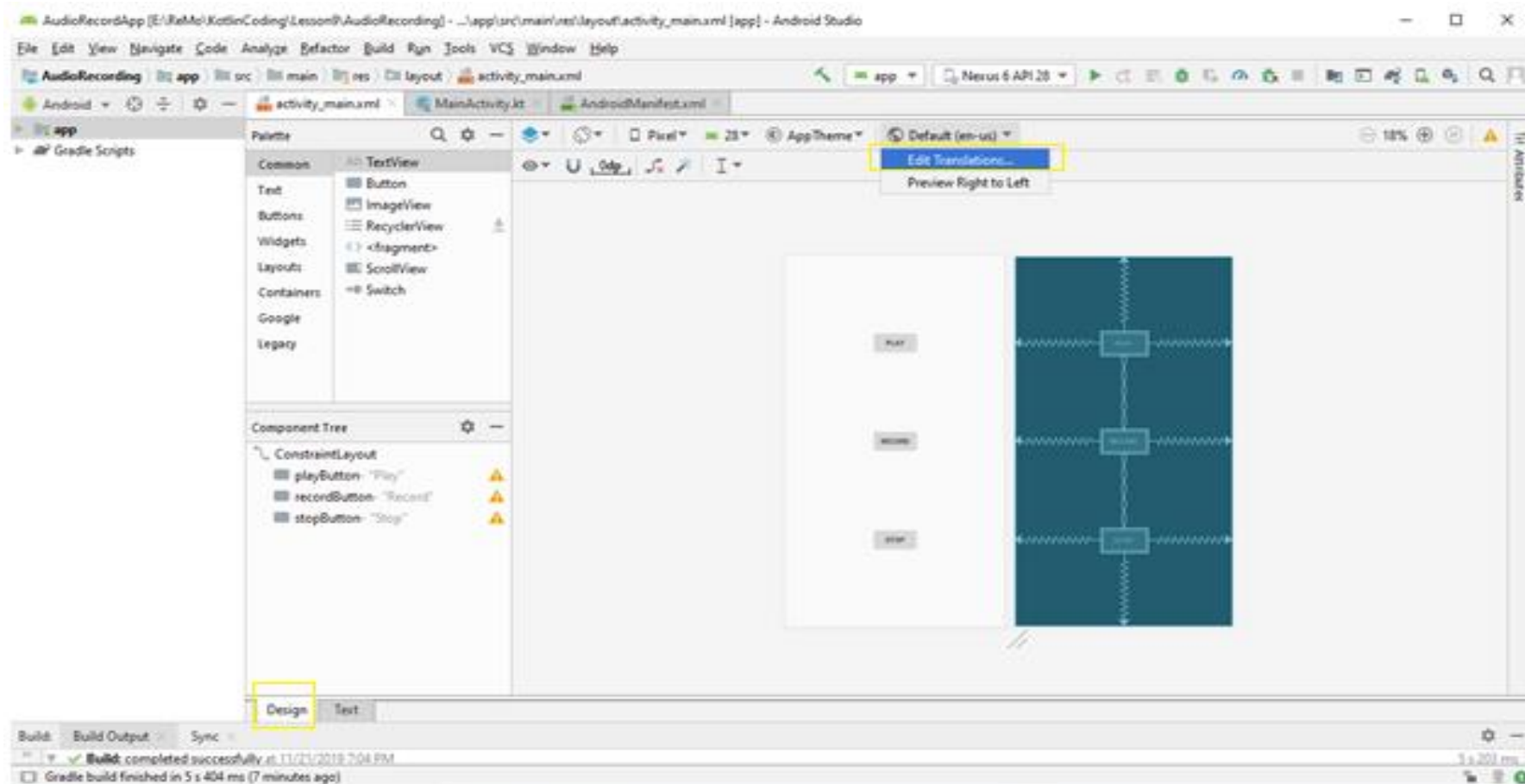
# Localization Step by Step Screens

Method 1 : Enter Data into String resource directly.



# Localization Step by Step Screens

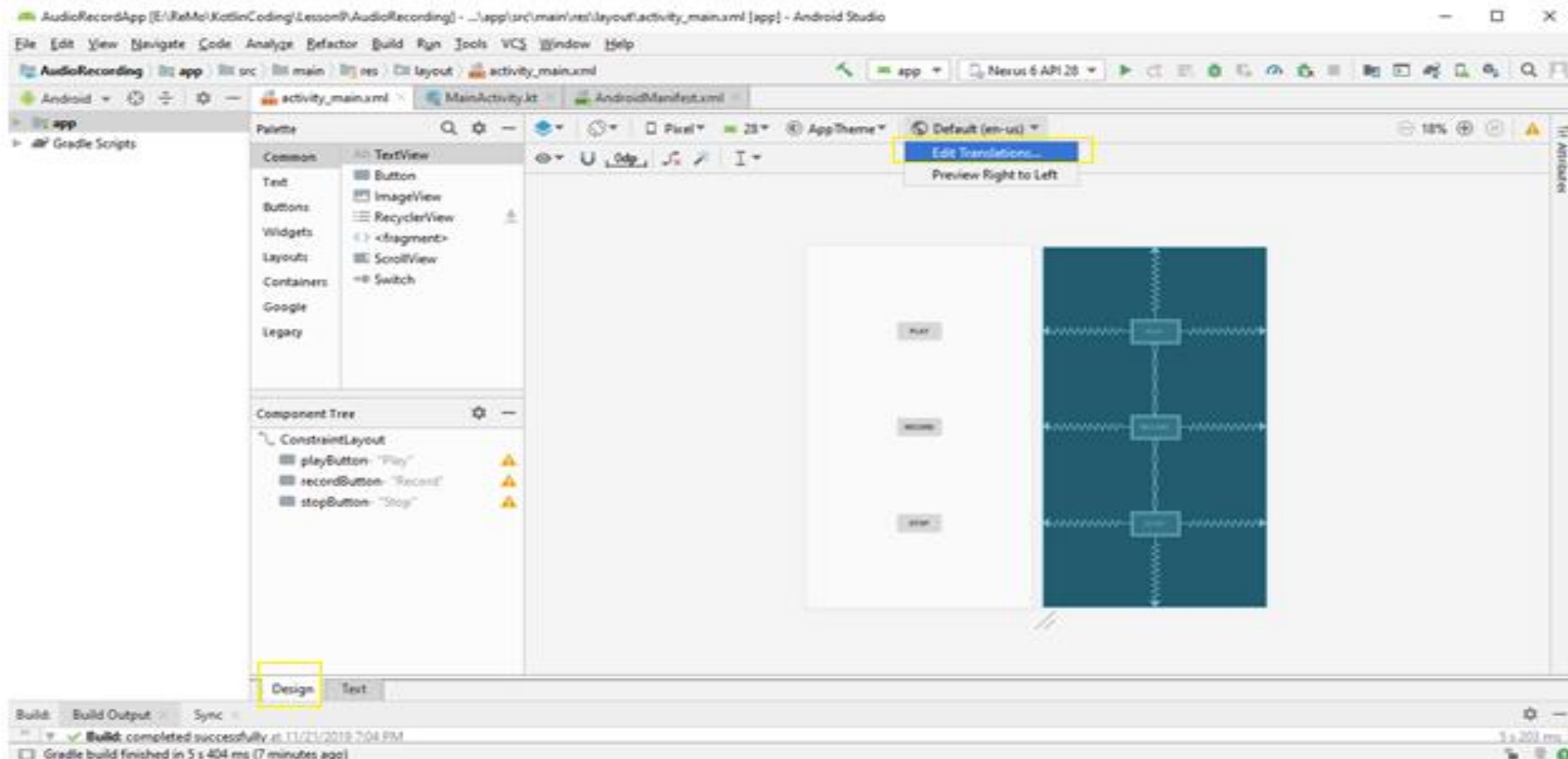
Method 2 : Step 1 : Go to your Layout Design View and  
Select Default(en-us) → Edit Translation





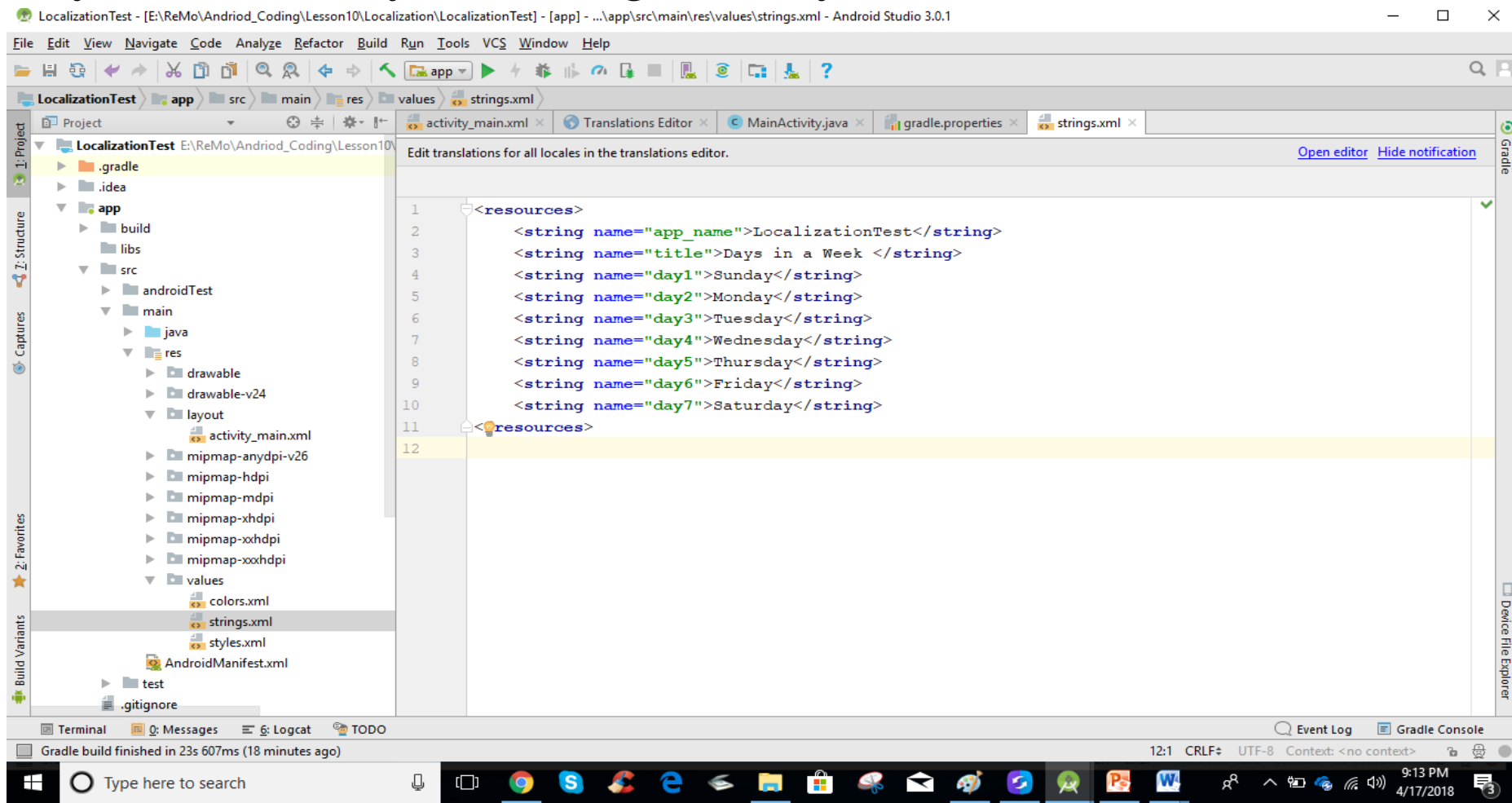
# Localization Step by Step Screens

Step 2 :Click + to add more String inputs as a Key/Value Pair into your resource.



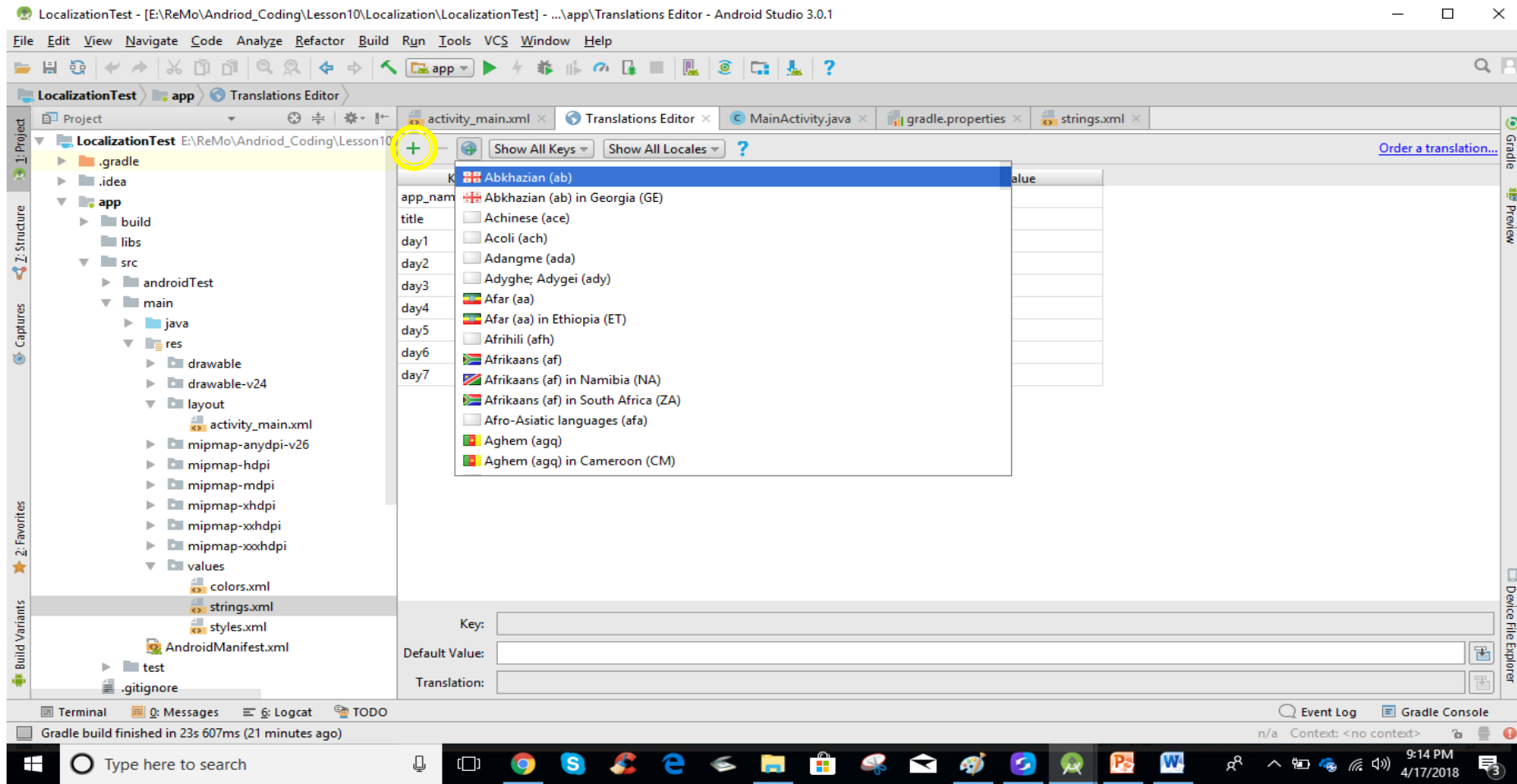
# Localization Step by Step Screens

Step 3 :If you click on your strings.xml, you will notice with all the values.



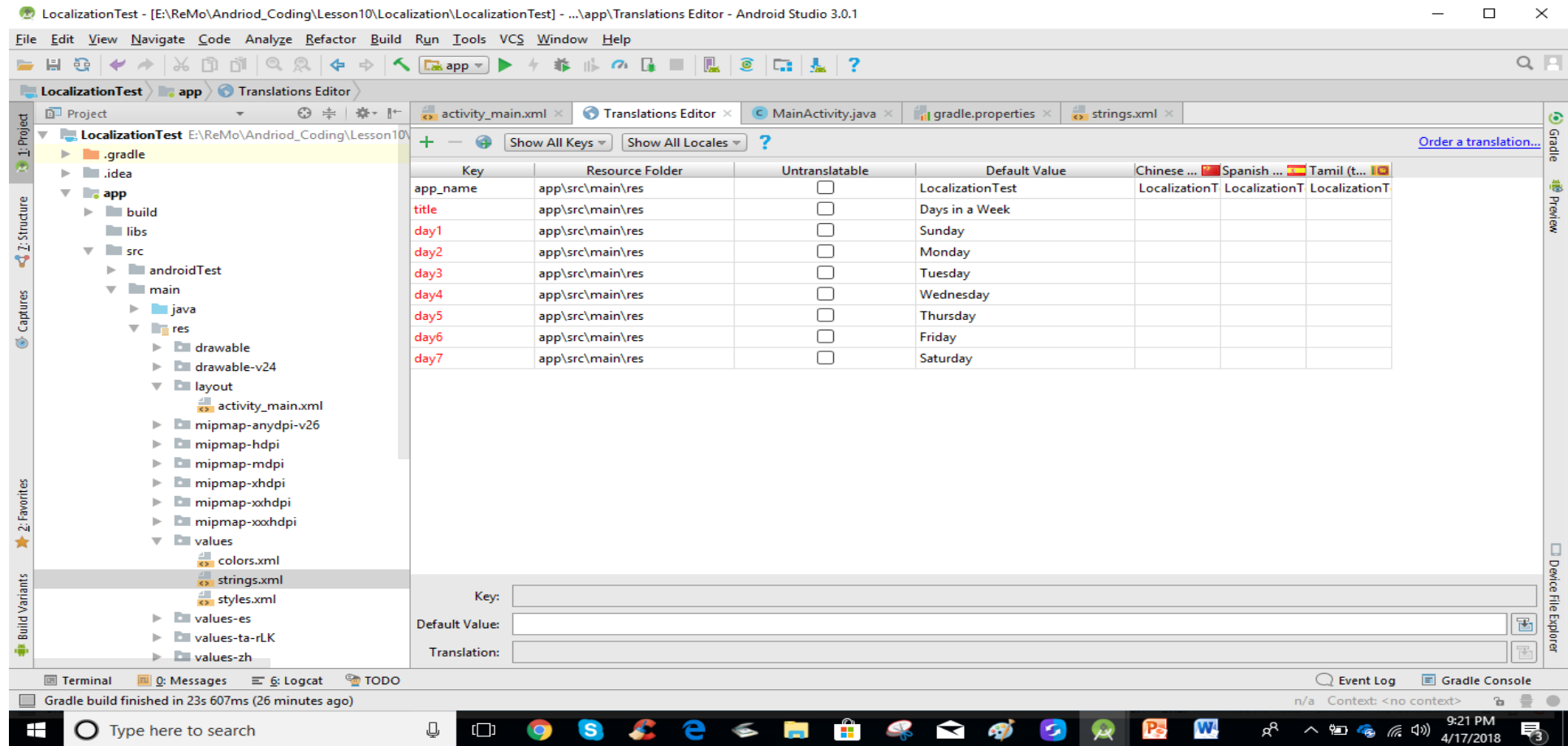
# Localization Step by Step Screens

**Step 4 :** To add Locale support, go back to your Translation Editor and click the (+) highlighted icon and choose the desired language.



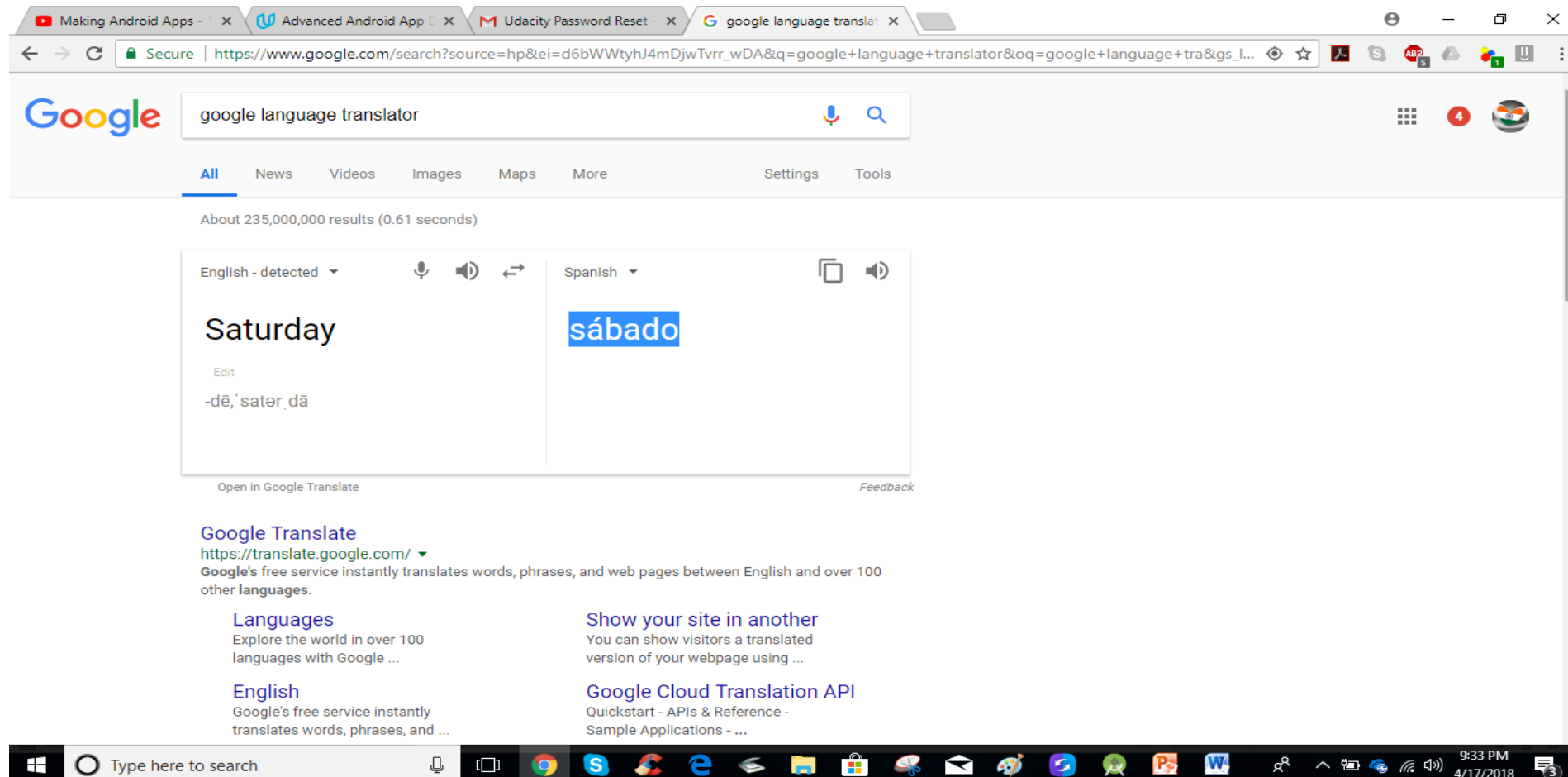
# Localization Step by Step Screens

**Step 5 :** After Choosing the Language, you will get screen looks like below. Red color indicates that need to provide terms for each Locale.



# Localization Step by Step Screens

**Step 6 :Use Google Language Translator to fill out the values in each locale or use <https://translate.google.com/>**



# Localization Step by Step Screens

Step 7 :Once everything is done, the screen looks below.

LocalizationTest - [E:\ReMo\Andriod\_Coding\Lesson10\Localization\LocalizationTest] - ...app\Translations Editor - Android Studio 3.0.1

File Edit View Navigate Code Analyze Refactor Build Run Tools VCS Window Help

Project LocalizationTest E:\ReMo\Andriod\_Coding\Lesson10

Translations Editor

activity\_main.xml Translations Editor MainActivity.java gradle.properties strings.xml

Show All Keys Show All Locales

| Key      | Resource Folder  | Untranslatable           | Default Value    | Spanish (es)       | Tamil (ta) in Sri Lanka (LK) |
|----------|------------------|--------------------------|------------------|--------------------|------------------------------|
| app_name | app\src\main\res | <input type="checkbox"/> | LocalizationTest | LocalizationTest   | LocalizationTest             |
| title    | app\src\main\res | <input type="checkbox"/> | Days in a Week   | Días en una semana | ஒரு வாரத்தில் நாட்கள்        |
| day1     | app\src\main\res | <input type="checkbox"/> | Sunday           | domingo            | ஞாயிறு                       |
| day2     | app\src\main\res | <input type="checkbox"/> | Monday           | lunes              | திங்கட்கிழமை                 |
| day3     | app\src\main\res | <input type="checkbox"/> | Tuesday          | martes             | செவ்வாய்க்கிழமை              |
| day4     | app\src\main\res | <input type="checkbox"/> | Wednesday        | miércoles          | புதன்கிழமை                   |
| day5     | app\src\main\res | <input type="checkbox"/> | Thursday         | jueves             | வியாழக்கிழமை                 |
| day6     | app\src\main\res | <input type="checkbox"/> | Friday           | viernes            | வெள்ளி                       |
| day7     | app\src\main\res | <input type="checkbox"/> | Saturday         | sábado             | சனிக்கிழமை                   |

Order a translation...

Key: Default Value: Translation:

Terminal Messages Logcat TODO

Gradle build finished in 23s 607ms (26 minutes ago)

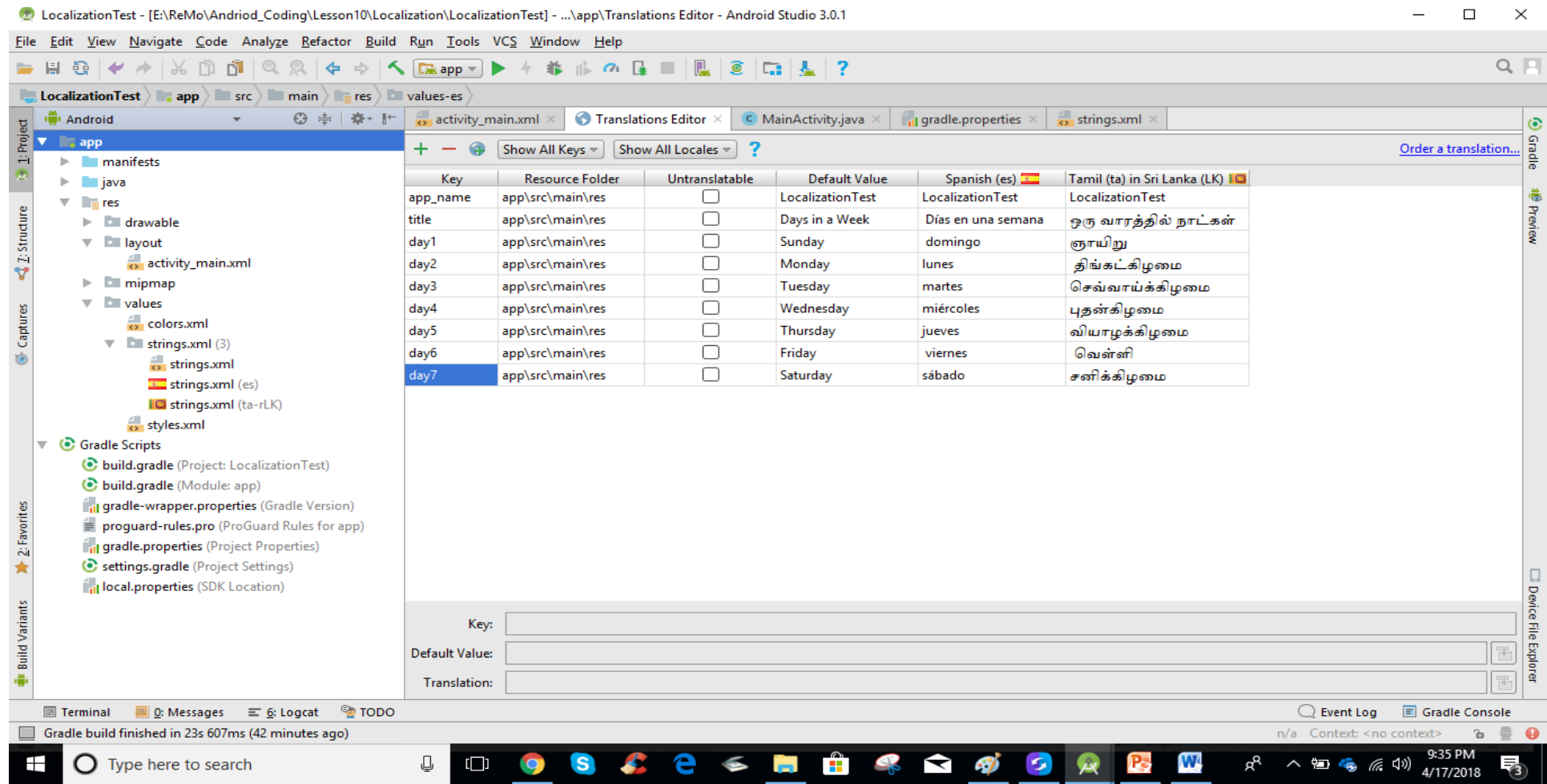
Event Log Gradle Console

n/a Context: <no context>

9:33 PM 4/17/2018

# Localization Step by Step Screens

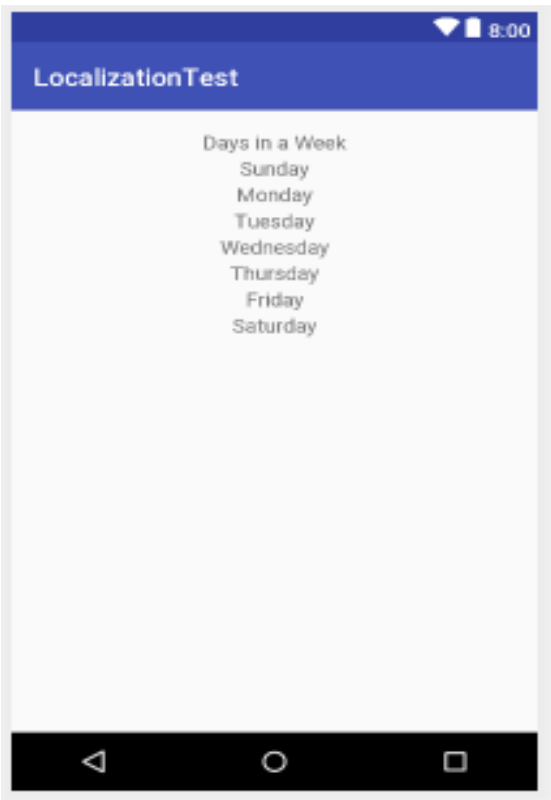
Step 8 : Open your res -> values->strings.xml, the screen as looks as below





# Localization Step by Step Screens

**Step 9 :Go to your activity\_xml and design layout as shown below. Add eight TextView components and make all component text attribute reference with its string resource.**



```
<TextView
```

```
    android:layout_width="wrap_content"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="@string/title"  
    android:textSize="16sp"/>
```

```
<TextView
```

```
    android:id="@+id/textView1"  
    android:layout_width="match_parent"  
    android:layout_height="wrap_content"  
    android:layout_gravity="center"  
    android:gravity="center"  
    android:text="@string/day1"  
    android:textSize="16sp"/>
```



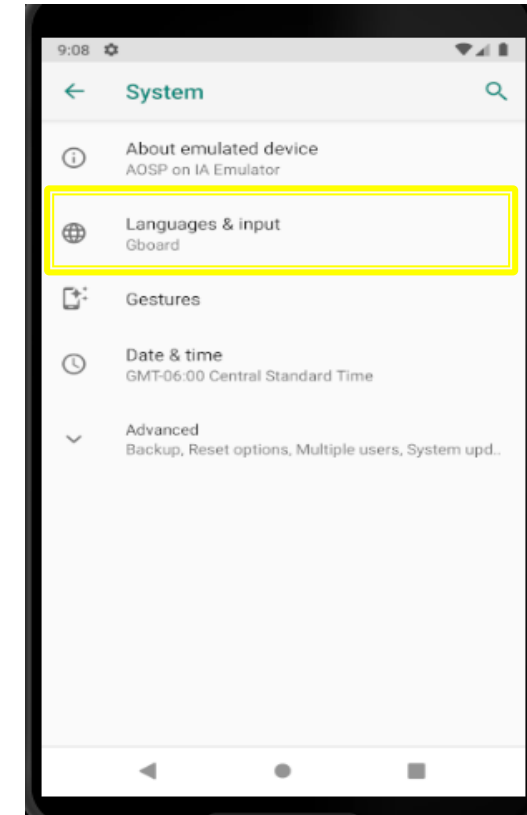
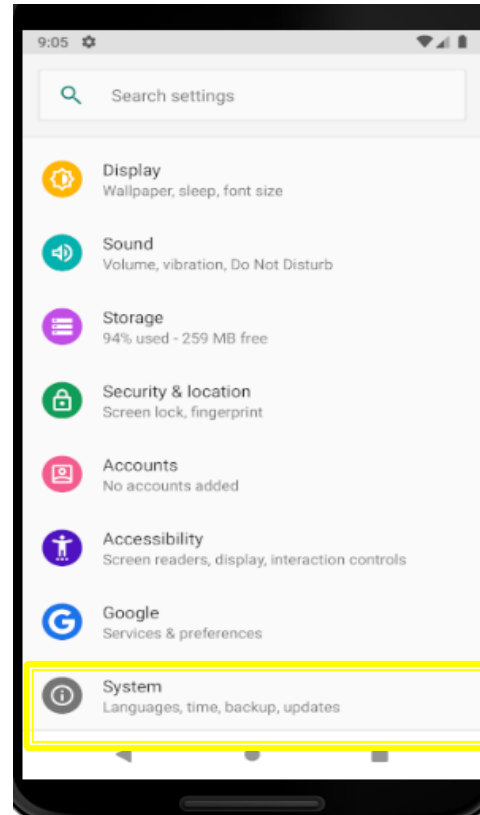
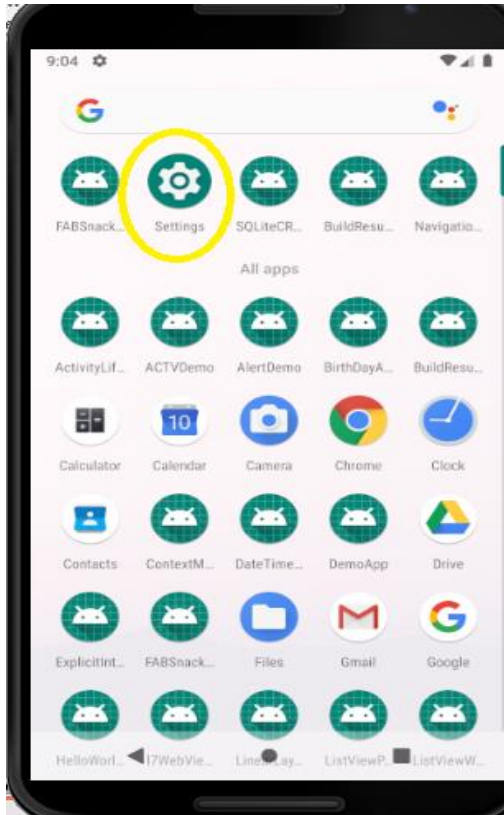
# How to change the Language Settings

From your emulator or real device,

Choose Settings

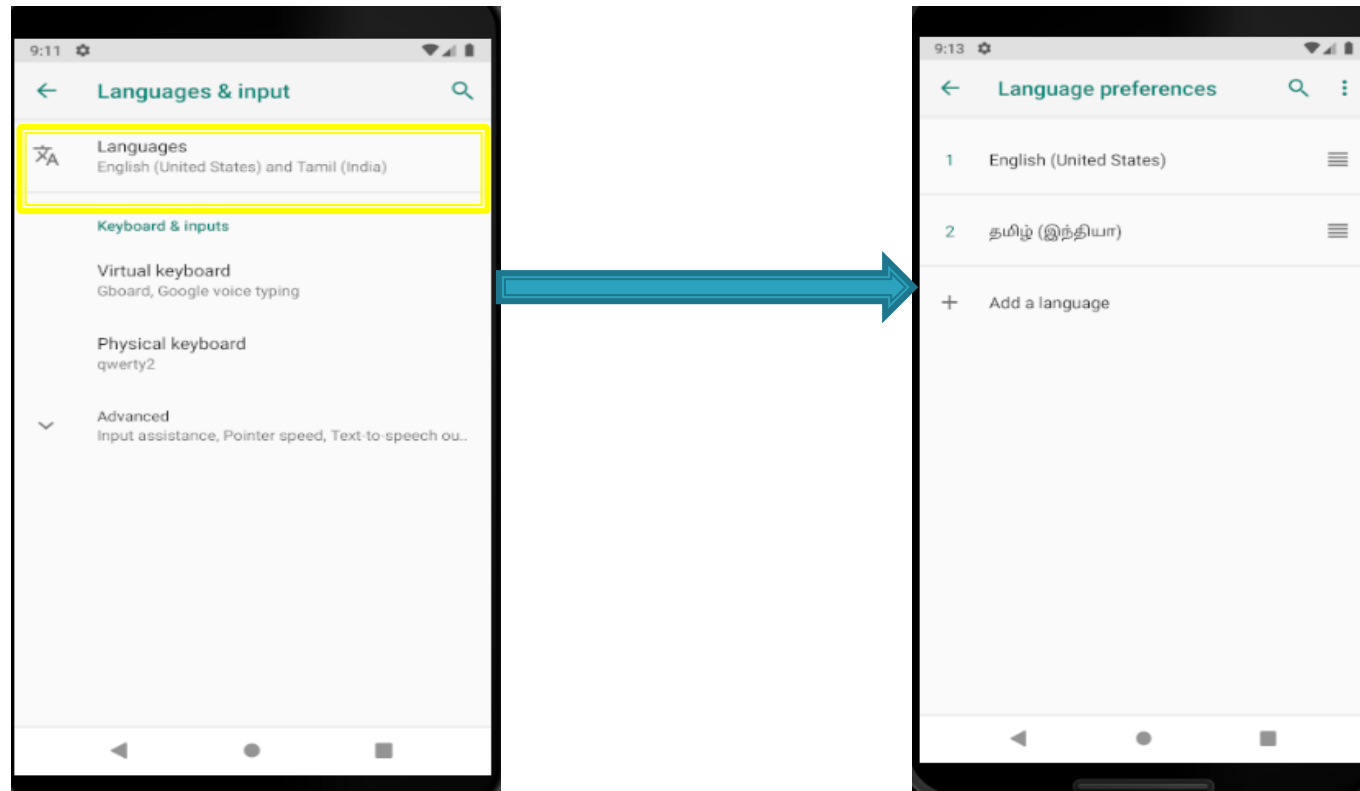
→ System

→ Language & Input



# How to change the Language Settings

Continuation of Previous slide Screenshots,  
Choose Languages, English is the first priority now. Drag your preferences and click Add a language to insert more languages. Make sure that, you should have the String resources for the selected language in your app.



# Online tools to get multilanguage strings.xml resource file

- Automatic String Resource Translation
- Upload the original strings.xml file and get the desired language.
- <https://asrt.gluege.boerde.de/>
- <https://localise.biz/free/converter>

# Change Language settings Programmatically

- To get localized data one should use Resources with the desired Locale set in their Configuration.
- Update the resources via updateConfiguration with a config that includes the desired locale display metrics.
- Recreate your activity
- **Refer: LocalizationDynammic**

# Code

```
// get the resource's current configuration.
val config = resources.configuration
//Construct a locale from language and country.
    val locale = Locale(code)
    Locale.setDefault(locale)
    config.locale = locale
// Update new locale settings
resources.updateConfiguration(config, resources.displayMetrics)
recreate()
```

## Main Point 2

In order to localize the strings used in your application, make a new folder under res with name of values-local where local would be the replaced with the region. Start by adding values directories to your resources so that Android knows you have localized your app completely to your broad range of audience. *Science of Consciousness: Similarly in Cosmic consciousness one can feel the completeness of inner and outer life.* The experience that activity is going on by itself, with the doer, the self-uninvolved.

# UNITY CHART

## CONNECTING THE PARTS OF KNOWLEDGE WITH THE WHOLENESS OF KNOWLEDGE

*Outer depends on inner*

1. To build multilingual Android apps you need to collect the texts into resource files and translate them.
2. You can provide any resource type that is appropriate for the language and culture of your users.  
The right thought at the right time for full effectiveness without strain.

- 
- 
3. **Transcendental Consciousness:** TC is the home of all knowledge. The Upanishads declare "Know that by which all else is known" – this is the field of pure consciousness.
  4. *Impulses within the Transcendental field:* The infinite diversity of these impulses within the transcendental field, is possible due to the infinite creativity and intelligence.
  5. *Wholeness moving within Itself:* In Unity Consciousness, one sees that the "key" to accessing complete knowledge of any object is the infinite value of that object, pure consciousness, which is known in this state to be one's own Self.

