



I'm not robot



Continue

Change android region

Since there is no complete answer for the current way to solve this problem, I try to provide instructions for complete decisions. If anything is missing or could be done better, please comment. Are common. At first, there are several libraries that want to fix it, but they all seem outdated or deprived of certain functions. I think the record library can't be a good/simple way to solve this problem, because it doesn't take much to do, and that's what needs to be done. Performance is a change in existing code, not the use of something completely separate.

Therefore, I passed the following instructions which must be executed. My solution is mainly based on (like localhost). This is the best code I have found that guides me. Some notes: if needed, provide multiple implementations to change the national environment for Android N (and above) and use the Refresh() method for manually updating all strings after changing the national environment (in using the usual GetString(ID) has getString(Id)) which is not needed in the following approach, which only supports languages and not full locations (which also include region (country) and the option codes), I changed it a bit, separating the part that remains from the part from the part from the part from the part National environment chosen (as it would be done separately, as shown below). The solution consists of the following two steps: Forever change the national environment in which applications use applications and obtain application. Use your own national environment set without rebooting. Step 1: Edit the area. Add lists. For preferred convenience with accessible languages (need to save when languages are added later) Import android.annotation.targetapi; Import Android.Content.Context; Import android.content.sharedpreferences; Import android.content.res.configuration; Import android.content.res.resources; import android.os.build; Import android.preference.PreferenceManager; Import Since there is no detailed answer to the current solution to this problem, I am trying to provide a detailed guide on how to solve it.

Write if something is missing or could be done better. General Information First, there are libraries that try to solve the problem, but they all seem to be outdated or lack some features. I also think that writing a library might not be the right/easy way to solve this problem since it's not there. It's a lot and you need to change existing code rather than use something entirely. That is why I have developed this guide, which should be detailed. My solution is mostly based on (as already posted on LocalHost). This is the best code I have found for guidance. A few notes: the various setups are intended to replace the local Android N (and above) method on each update() action to update all strings manually with the local one (using a simple getString(id) if necessary). not required for the next method. It only supports speeches, not all locals (including region (country) codes and variants).

I modified it by thinning out the part that saves the location of the selected locale (since you can do it separately as suggested). The solved solution consists of the following two steps: To permanently replace the local program, the program will use the configured locale without publishing it (inserted later) Import android.annotation.targetapi; import android.content.context; import settings android.content.share; import android.content.res.configuration; import android.content.res.resources; import android.os.build; Import android.preference.PreferenceManager; Import on the line specified in the specified line. * * @param context * @param localeSpec local spec used for Android resources (note: not yet supported by codes and country capability); SECTIN LINE "SYSTEM" set *National environment in locale * @return * /public Static Control SetLocale(ContextContext, String LocalEspec) {locations; if (localEspec.equals("System")) {if.version.sdk_int>=build.version_codes.n}{locale=secours.getSystem(). Get configuration(). Get local(). Get (0); } Else { // noimpltion deberitung locale = Resources.getSystem().

Get configuration(). Local; } Else { locale = new locale (localEspec); } Locale.setDefault (local scheme); if (build.version.sdk_int> = build.version_codes.n) { return resource update (context, local); } Else { return updateSourcesLegacy (context, location); } } @Targetapi (build.version_codes.n) private static context -UpdateReResources (contextContext, local) {configuration config = context.getResources(). receive configuration(); Configuration.SetLocale (local); configuration.setLayoutDirection (local); Return context.CreateConfigurationContext(configuration); } @SuppressWarnings ("shutdown") private static context -uUpdateUrcGacy (contextContext, Area Shell) {Resources Resources = conCecorces (); Configuration Configuration = Resources.getConfiguration(); Configuration.local = local; if (build.version.sdk_int> = build.version_codes.jelly bean mr1) { configuration.setLayoutDirection (local schema); } Resources.UpdateConfiguration (configuration, Resources.getDisplayMetrics ()); return context; } } Create a parameter as follows: Import android.content.sharepferation; import android.os.bundle; import android.preference.PreferenceManager; Import Android.PreferenceManager; Import android.view.layoutinflater; Import android.view.view; Import android.view.viewgroup; Importing myPackage.Localehelper; Import MePackage.r; /*** Fragmentcontainer, Savedinststate pack) {View = Infantr.inflate (Year Layout.fragment_ettings, Container, False); Review; } @Override Public Void OnharedPreferencechanged (SharedPreferen SharedPreferences, Key) {Switch (Key) {Case Key_pref_slangage: Localehelper.Setlocale (GetContextExt ()), Preference.late () in outure () () {Super.onresume (); // Documentation requires the reference to the listener must be maintained for as long as possible, as it is only called Getpreferenreen () GetsharedPreferene (). (This); } @Override public voiduse () {super.onpouse () () (.



Synchronized with "Estas_langage_values" file in Strings.xml (votes must answer). = "Locals" >> @String/System_lale @String/Default_lale CREEN Settings You can take this section to allow the user to select available languages: