

[RxJava \(http://developer.alexanderklimov.ru/android/rx/\)](http://developer.alexanderklimov.ru/android/rx/)

[Советы \(http://developer.alexanderklimov.ru/android/tips-android.php\)](http://developer.alexanderklimov.ru/android/tips-android.php)

[Статьи \(http://developer.alexanderklimov.ru/android/articles-android.php\)](http://developer.alexanderklimov.ru/android/articles-android.php)

[Книги \(http://developer.alexanderklimov.ru/android/books.php\)](http://developer.alexanderklimov.ru/android/books.php)

[Java \(http://developer.alexanderklimov.ru/android/java/java.php\)](http://developer.alexanderklimov.ru/android/java/java.php)

[Kotlin \(http://developer.alexanderklimov.ru/android/kotlin/\)](http://developer.alexanderklimov.ru/android/kotlin/)

[Дизайн \(http://developer.alexanderklimov.ru/android/design/\)](http://developer.alexanderklimov.ru/android/design/)

[Отладка \(http://developer.alexanderklimov.ru/android/debug/\)](http://developer.alexanderklimov.ru/android/debug/)

[Open Source \(http://developer.alexanderklimov.ru/android/opensource.php\)](http://developer.alexanderklimov.ru/android/opensource.php)

[Полезные ресурсы \(http://developer.alexanderklimov.ru/android/links.php\)](http://developer.alexanderklimov.ru/android/links.php)

Локализация. לשון חז"ל (хатуль мадан) - кот учёный

В статье, посвященной [ресурсам \(http://developer.alexanderklimov.ru/android/theory/resources.php#locale\)](http://developer.alexanderklimov.ru/android/theory/resources.php#locale), мы уже затрагивали эту тему с теоретической точки зрения. Разберем теперь практическую часть. Мы уже умеем пользоваться строковыми ресурсами и меняли строчки, например, *Hello, World* на *Здравствуй, Мир!* Но на самом деле это не совсем правильный подход. Для профессионально написанного приложения лучше использовать другой подход. А именно, локализацию.

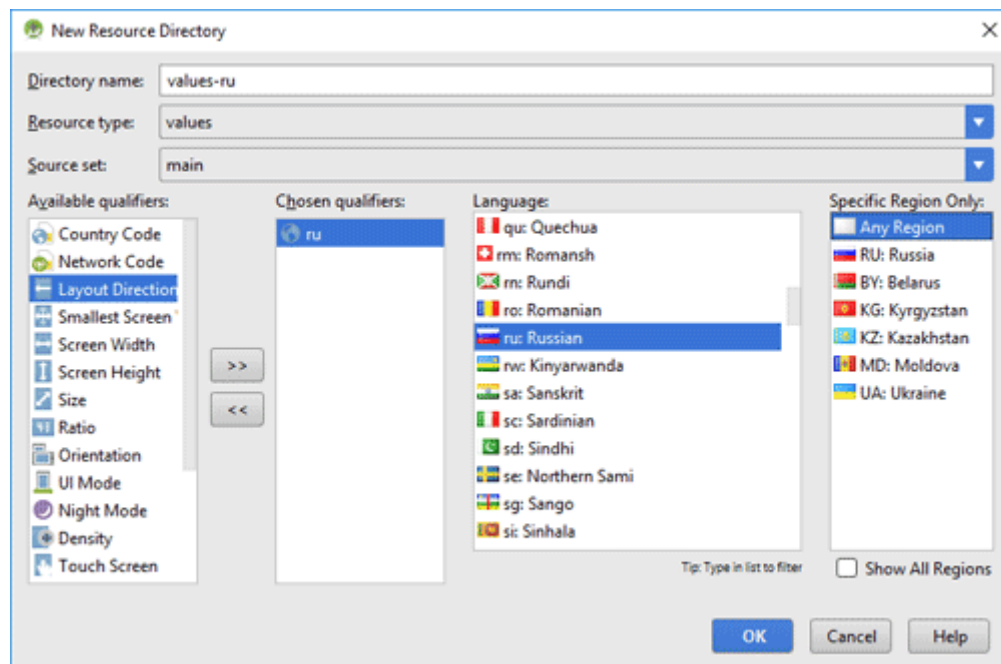
Что это значит? Мы не будем менять строчку **Hello, World**, которая будет считаться строкой по умолчанию, а создадим новый локализованный ресурс. Это даст нам дополнительное преимущество, если пользователь запустит приложение на телефоне с английской локалью, то он увидит текст на знакомом ему языке. Если приложение запустит русский пользователь, то он увидит текст на русском языке. Согласитесь - очень удобно.

Напишем приложение, которое будет поддерживать английскую и русскую локализации. По умолчанию будет использоваться английская локализация. В этом случае, если на телефоне не будет найден нужный локализованный ресурс, то будет применяться английский вариант.

Чтобы не усложнять материал, воспользуемся стандартным приложением, который мы изучали в самом начале (<http://developer.alexanderklimov.ru/android/android2.php>). Создадим новый проект "LocaleApp", но на этот раз мы не будем трогать файл **strings.xml** в каталоге **res/values**.

Для русской локализации необходимо создать новый подкаталог **values-ru** в том же каталоге **res**. Если вы планируете создавать локализацию только для русского языка, то название каталога не трудно запомнить и вы можете создать папку вручную через **res | New | Directory**. Но вряд ли вы запомните названия других папок. Поэтому покажу способ создания с помощью мастера студии.

Щёлкаем правой кнопкой мыши на папке **res** и выбираем **New | Android resource directory**. В диалоговом окне в левой части **Available qualifiers:** выбираем пункт **Locale** и переносим его в правую часть **Chosen qualifiers:** с помощью кнопки с двумя стрелками вправо. В появившейся третьей колонке выбираем нужные языки, например, русский. Вы увидите, что в поле **Directory name** автоматически появится нужное название папки. Дополнительно вы можете указать и регион в колонке **Specific Region Only**



В режиме **Android** вы можете не увидеть созданную папку, поэтому временно переключитесь в режим **Project**. Скопируйте файл **res/values/strings.xml** и вставьте его в новую папку. Можете вернуться в прежний режим. Если папка не пуста, то она уже видна.

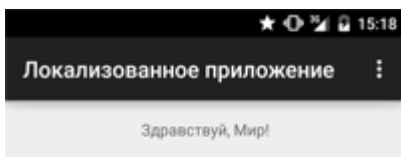
Изменим его содержание.

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name">Локализованное приложение</string>
    <string name="hello_world">Здравствуй, Мир!</string>
</resources>
```

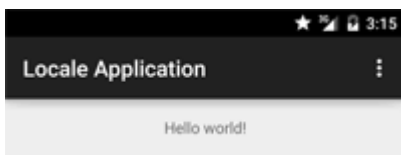
В английской версии им соответствуют ресурсы

```
<string name="app_name">Locale Application</string>
<string name="hello_world">Hello world!</string>
```

Запустим приложение и увидим, что текст выводится на русском, если у вас на эмуляторе или реальном устройстве выбраны русские настройки.



Как перейти на американскую локаль? Идём в **Настройки**→**Язык и клавиатура**→**Выбрать язык**. Выбираем английский язык **English (United States)**. Снова запускаем приложение в эмуляторе или на телефоне. Теперь наше приложение будет выглядеть так:



Ресурсы из каталога **res/values** считаются ресурсами по умолчанию. Они отображаются в том случае, когда система не найдёт ресурсы для текущей локали устройства. В нашем случае, при выборе любого языка, кроме русского будет использоваться **res/values**. Если мы захотим разделить англичан и американцев, тогда придется создавать два каталога: **res/values-en-rUK** и **res/values-en-rUS**. Обратите внимание, что перед кодом второй части ставится буква 'r'. Иными словами, мы определяем не только язык, но и регион.

Проведем несколько экспериментов. Удалим из файла **res/values-ru/strings.xml** строковый ресурс **hello_world** с текстом **Здравствуй, Мир!**. Сохраним изменения и запустим проект на устройстве с русским языком. Платформа заменила отсутствующий в локализации ресурс дефолтным английским.

Продолжим опыты и поступим наоборот - удалим теперь ресурс из файла **res/values/strings.xml**. На устройстве выберем какой-нибудь другой язык, например, итальянский. В этом случае вы увидите что-то непонятное, скорее всего вы получите сообщение об ошибке.

Отсюда можно сделать вывод, что вам необходимо следить за ресурсами по умолчанию. Старайтесь проконтролировать, чтобы ваше приложение всегда содержало все необходимые ресурсы по умолчанию, а уже потом можете начинать локализацию приложения. Для этого вам достаточно будет скопировать файл в новый подкаталог и отредактировать его.

На самом деле локализовать можно не только строковые ресурсы. В разных странах принято рисовать какой-нибудь объект в соответствии со своими традициями, например, изображение почтового ящика. В этом случае вы можете создать каталог **/res/drawable-de**, в котором будут находиться изображения, предназначенные для немцев.

Кстати, выбор нужных языков может повлиять на ваши продажи платных программ. Google проводит различия между странами, в которых можно распространять платные и бесплатные программы. Имейте это в виду.

В коде вы обращаетесь к нужным ресурсам следующим образом:

```
String hello = getResources().getString(R.string.hello_world);

// или так
TextView textView = new TextView(this);
textView.setText(R.string.hello_world);
```

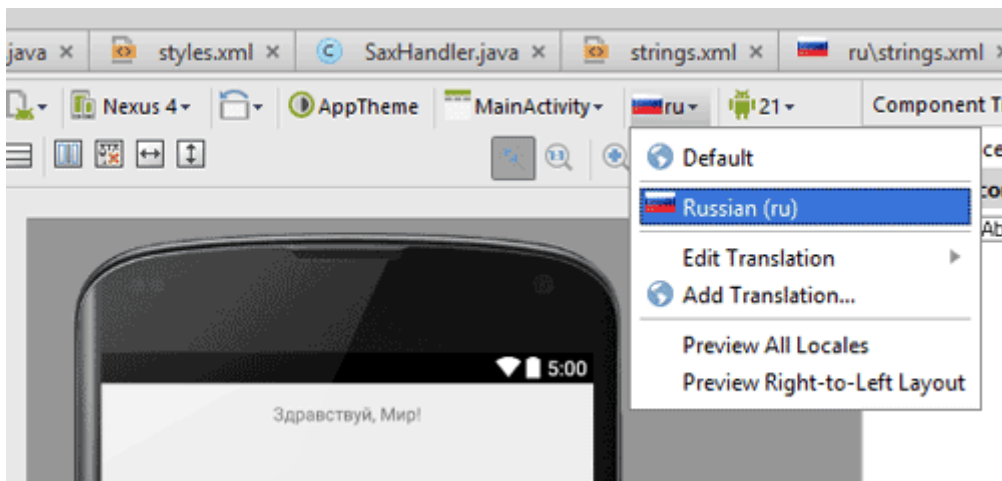
Система сама подставит нужную строку. Явно указать, что хочется использовать ресурс на конкретном языке, у вас не получится.

Спустя продолжительное время один из читателей подсказал, как всё-таки задействовать локальный ресурс. Объяснять код не буду, сами поймёте, когда наберётесь опыта.

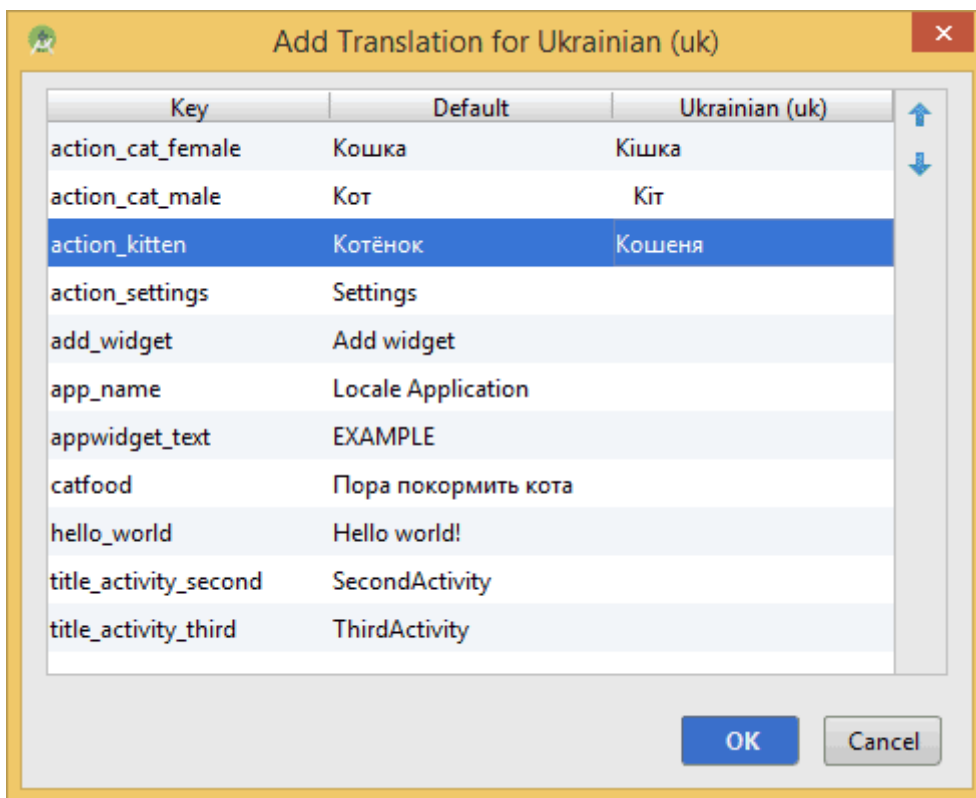
```
Locale locale = new Locale("en");
Locale.setDefault(locale);
Configuration configuration = new Configuration();
configuration.locale = locale;
getBaseContext().getResources().updateConfiguration(configuration, null);
// выводим английский текст на русской локали устройства
setTitle(R.string.app_name);
```

Просмотр и создание локализованных строк в режиме дизайна

Чтобы увидеть, как меняется текст в различных локализованных версиях вашего приложения, необязательно запускать приложение в эмуляторе. Можно в студии на панели инструментов щёлкнуть на значке глобуса и выбрать из выпадающего списка нужный вам пункт (он появится после того, как вы создадите необходимые локализованные ресурсы), и все локализованные строчки сразу поменяются на экране активности.



Можно сразу создать локализованный файл через удобный мастер. Там же в меню есть пункт **Add Translation...**, который выводит диалоговое окно со списком уже имеющихся строковых ресурсов из вашей программы. Вам нужно выбрать язык для перевода, а затем заполнить в таблице нужную колонку.



После того, как вы завершите перевод всех ресурсов, нажимаете на кнопку **OK** и у вас появится готовая папка с новым файлом **strings.xml** для выбранной локализации.

Ничего сложного.

Непереводимая игра слов с использованием местных идиоматических выражений

Если вы решили использовать локализацию на разных языках, то можете встретиться с одной проблемой. Предположим, вы перевели только несколько строковых ресурсов и в один прекрасный день решили выложить программу в магазин приложений. Android Studio при сборке подписанного приложения может заругаться, что вы перевели не все строки. Есть два способа обхода проблемы. Вы можете использовать атрибут **translatable="false"**, чтобы указать среде разработки, что данная строка не требует перевода:

```
<string name="title_activity_start" translatable="false">StartActivity</string>
```

Второй вариант - явно указать, что все строки в файле не нужно переводить с помощью **tools:ignore="MissingTranslation"**.

```
<?xml version="1.0" encoding="utf-8"?>
<resources xmlns:tools="http://schemas.android.com/tools" tools:ignore="MissingTranslation">
...
```

Есть ещё один вариант, когда непере译имые строки размещаются не в ресурсе **strings.xml**, а в другом файле под именем **donottranslate.xml**.

Когда вы исправите строковые ресурсы, то избавьтесь от ошибок через **Project | Clean...** и попытайтесь собрать приложение заново.

Поддержка письма справа налево

Арабы и некоторые другие народы используют непривычную для европейцев схему написания текстов справа налево. В Android 4.2 появилась поддержка данного режима. В блоге (англ.) [Native RTL support in Android 4.2](http://android-developers.blogspot.ru/2013/03/native-rtl-support-in-android-4.2.html?linkId=15420153) (<http://android-developers.blogspot.ru/2013/03/native-rtl-support-in-android-4.2.html?linkId=15420153>) рассказано о некоторых вещах по этой теме.

Теперь вы можете писать приложение для израильских военкоматов. Русские призывники в интервью офицерам душевного здоровья часто используют сочетание хатуль мадан (רוחל מדין). Теперь вы знаете, как локализовать строки пушкинского стихотворения про кота учёного.

Несколько локалей

В Android 7.0 (API 24) пользователь может установить несколько локалей. В этом случае, если не будут найдены ресурсы для первой локали, то будут искаться ресурсы для второй локали. Таким образом, если француз знает два языка, например, французский и испанский, а в приложение не поддерживает

французского, но поддерживает испанский, то пользователь увидит текст на испанском, а не на английском.

Но тут есть одна особенность, если вы в своём приложении используете библиотеку совместимости **appcompat-v7**. Она может показать текст на английском, если не найдет ресурсов для первой локали, игнорируя ресурсы второй локали. Чтобы избежать проблемы, добавьте в **build.gradle** модуля дополнительную настройку.

```
defaultConfig {  
    ...  
    resConfigs "auto"  
}
```

Дополнительное чтение

Обсуждение статьи (<http://forum.alexanderklimov.ru/viewtopic.php?id=23>) на форуме.

Реклама

в типографии <https://vivario.com.ua/blanki> (<https://vivario.com.ua/blanki>) баннер киев
печать;промышленные обогреватели инфракрасные [вот тут](http://ekostar.com.ua/) (<http://ekostar.com.ua/>) инфракрасные
обогреватели одесса ekostar

Реклама