

Яндекс

ЯНДЕКС

Android Library Projects, советы по применению

Ильдар Каримов
Android-Разработчик

Зачем обычно нужен library project

- Переиспользование кода/компонентов
- **Код и ресурсы**
- Компоненты/библиотеки->приложение

А что если наоборот –
превратить целое
приложение в разделяемую
библиотеку?

Зачем

Несколько конечных проектов с разными манифестами, в которых можно поменять

- Имя пакета (package name)
- Графику и другие ресурсы
- Конфигурацию (например, адрес сервера)

**Compile-time препроцессинг
– ЭТО ПЛОХО!**

Примеры использования

Для бета-версий

Для платной/бесплатной версии приложения

Для другого дизайна

Бета-версии

- Другое имя пакета
- Другое название
- Другая иконка
- Другой конфиг
- Пример – МЯК

Платная/бесплатная версии

- lite-версии игр и приложений
- 90+% общего java-кода
- Разделяем код защиты/покупки
- Нативные библиотеки копируются автоматически при сборке
- Ресурсы (assets) копируем и разделяем вручную

Другой дизайн/настройки

- Другое имя пакета
- Другое название
- Другая иконка
- Другой конфиг
- Другая графика
- Пример – SPB TV



SPB TV, MTC TV, velcom 3G TV, BSNL nexGTV+

Применение

Манифест

Код

Ресурсы

Манифест

- Имя пакета (package name)
- Своя реализация Application'a
- FQDN для activity, service
- Provider authorities="packagename.foo"

Манифест – SPB Shell 3D

Основные моменты, на которые нужно обратить внимание

```
<manifest      package="com.spb.shell3d">  
  
<application  android:icon="@drawable/ic_shell_launch"  
              android:label="@string/app_name_shell"  
              android:name=  
                "com.spb.shell3d.market.Shell3dApplicationMarket">  
  
<provider     android:authorities=  
              "com.spb.shell3d.weather">
```

Манифест – Яндекс.Shell

Основные моменты, на которые нужно обратить внимание

```
<manifest      package="ru.yandex.shell">  
  
<application  android:icon="@drawable/ic_shell_launch"  
              android:label="@string/app_name_shell"  
              android:name=  
                "ru.yandex.shell.YandexShellApplication">  
  
<provider     android:authorities=  
              "ru.yandex.shell.weather">
```

Код

- Application для расширения
- Provider для правильных authority

Application – library project

Объявление метода (зачастую абстрактного) в library project

```
public abstract class FooLibApplication extends Application{  
    public abstract String getFoo();  
}
```

Application – main project

Переопределение в главном проекте

```
public class Foo1Application extends FooLibApplication {  
    public String getFoo() {  
        return "Foo1";  
    }  
}
```

Application – main project

Декларация в манифесте

```
<application    android:name="com.foo.FoolApplication"  
                android:icon="..."  
                android:label="...">
```

Application – main/library project

Важно отметить, что вне зависимости от места вызова метода, он будет вызван не у самого `FooLibApplication`, а у конкретной реализации, заданной в манифесте (`Foo1Application`)

```
((FooLibApplication) getApplication()).getFoo()
```

Provider – поддержка authority

Non-static uriMatcher.addURI

```
private UriMatcher uriMatcher;

private void initAuthority(String authority) {
    uriMatcher = new UriMatcher(UriMatcher.NO_MATCH);
    uriMatcher.addURI(authority, "people", PEOPLE);
    uriMatcher.addURI(authority, "people/#", PEOPLE_ID);
}

@Override
public void attachInfo(Context context, ProviderInfo info) {
    super.attachInfo(context, info);
    initAuthority(info.authority);
}
```

Provider – проверка authority

Для минимизации шанса на ошибку в манифесте при копировании объявления provider'а, добавим дополнительную проверку на равенство authority из манифеста с ожидаемым authority из по имени нашего текущего пакета

```
private static final String AUTHORITY_SUFFIX = ".contacts";

@Override
public void attachInfo(Context context, ProviderInfo info) {
    super.attachInfo(context, info);
    String manifestAuthority = info.authority;
    String myAuthority = context.getPackageName()
+ AUTHORITY_SUFFIX;
    if (!manifestAuthority.equals(myAuthority)) {
        throw new RuntimeException(
"Unexpected authority in manifest: " + manifestAuthority
+ ", expected: " + myAuthority);
    }
    initAuthority(info.authority);
}
```

Ресурсы

- Графика
- Название приложения (+локализация)
- Конфигурация

В главном проекте ресурсы добавляются и удаляются, но не удаляются

Неудаляемые ресурсы

- Следите за всеми разрешениями
- Не копируйте лишних значений в `values-*`
- Для заменяемых неграфических ресурсов используйте отдельный файл

Ресурсы – DRY

Library: переносим название приложения из values/strings.xml (и всех его переводов) в один файл values/strings-no-loc.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <string name="app_name_shell" translatable="false">
        SPB Shell 3D
    </string>
</resources>
```

Ресурсы – DRY

Яндекс.Shell: задаём дефолтное название приложения в `values` и, при необходимости, переводы для поддерживаемых языков (`ru`, `uk`=Яндекс.Shell).

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name_shell" translatable="false">
    Яндекс.Shell
  </string>
</resources>
```

Конфигурация – доступ

Перенести адрес сервера из константы в ресурсы легко, но как прочитать его без context'a? Простой static метод в Application'e:

```
public abstract class FooLibApplication extends
    Application {

    private static Resources staticResources;

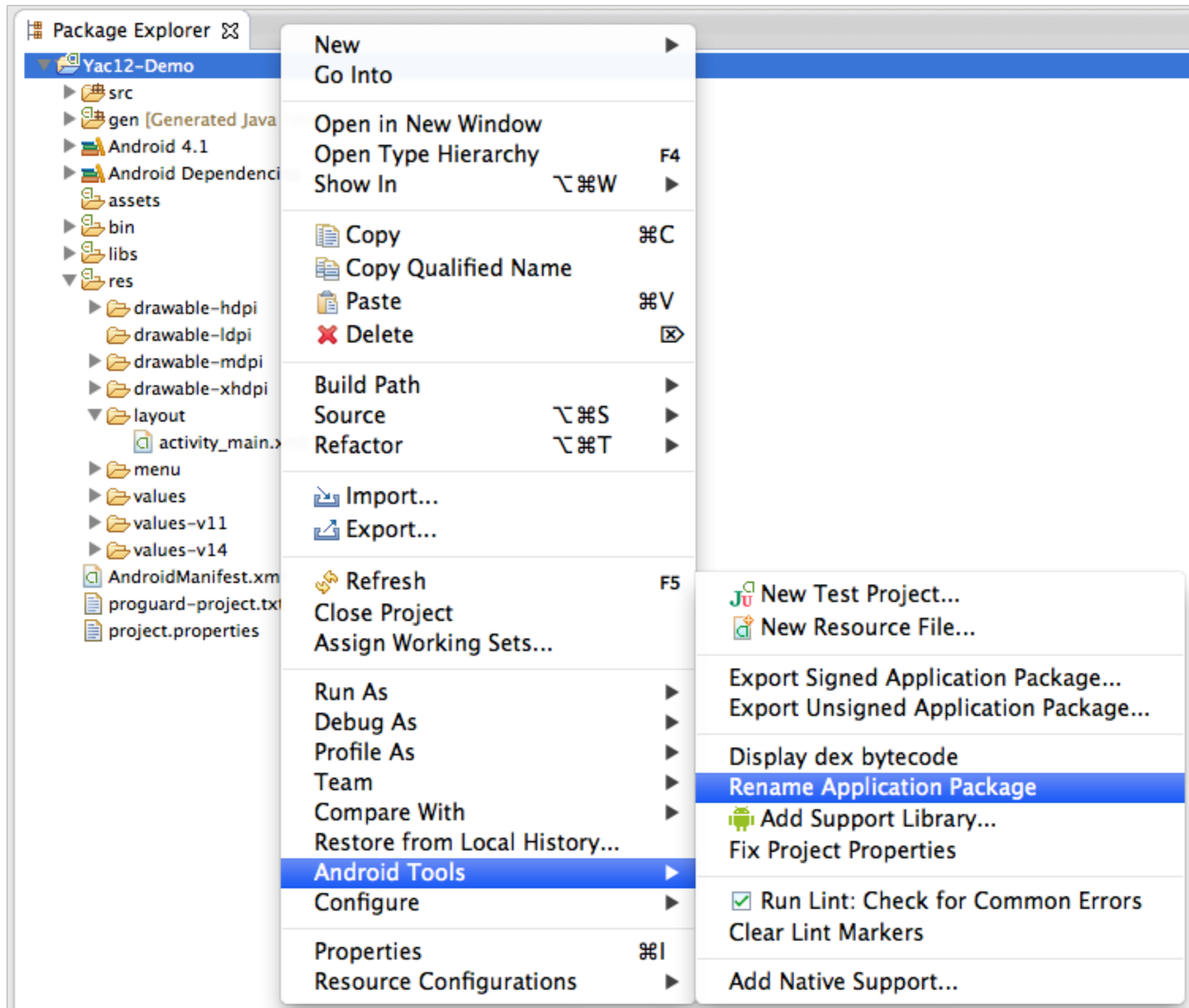
    @Override
    public void onCreate() {
        super.onCreate();
        staticResources = getResources();
    }

    public static String getStartupUrl() {
        return staticResources
            .getString(R.string.startup_url);
    }
}
```

И ещё

Автоматически переименовываем пакеты

Автоматически мёрджим манифесты



Eclipse: Android Tools->Rename Application Package

Автоматически мёрджим манифесты

Доступно с SDK tools r20, для включения добавляем одну строчку в `project.properties`. Итоговый манифест можно проверить в `bin/AndroidManifest.xml`

```
manifestmerger.enabled=true
```

Проблемы

Всё в исходниках рядом:

- Нельзя отдать «закрытый» код
- Полная пересборка каждый раз
- Нет версионирования и хранилища артефактов

Исходники/сборка/версионирование

- **Apklib+maven:** <http://code.google.com/p/maven-android-plugin/wiki/ArkLib>
- **Binary apklib+gradle:** см. следующий доклад

Пробуйте!

Для бета-версий

Для платной/бесплатной версии приложения

Для кастомизированных версий

<https://github.com/thevery/yac12>



Яндекс

Ильдар Каримов



thevery@yandex-team.ru

@the_very

Вопросы?