

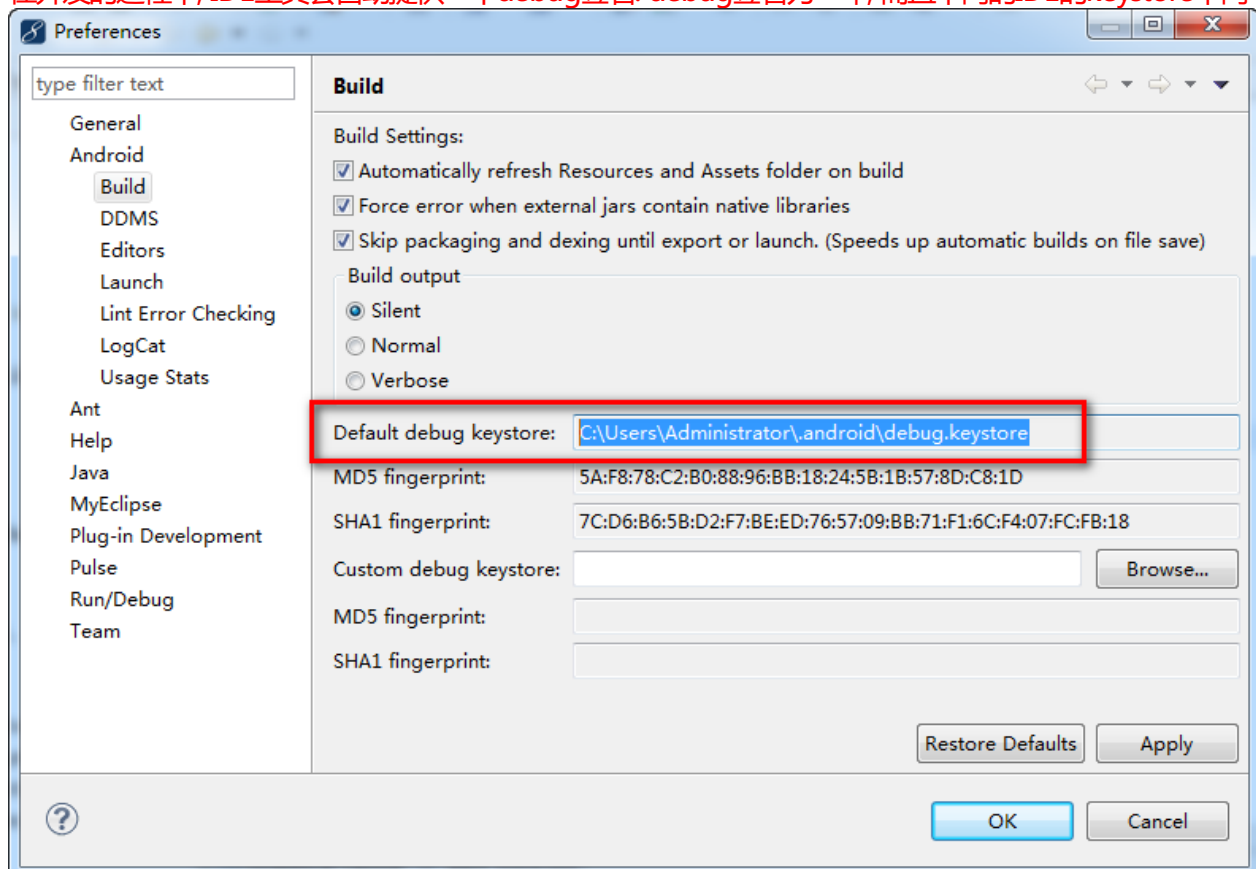
Android 签名详细介绍

在Android 系统中,所有安装到系统的应用程序都必有一个数字证书,此数字证书用于标识应用程序的作者和在应用程序之间建立信任关系. Android使用Java的数字证书相关的机制来给apk加盖数字证书,要理解android的数字证书,需要先了解以下数字证书的概念和java的数字证书机制. Android系统要求每一个安装进系统的应用程序都是经过数字证书签名的,数字证书的私钥则保存在程序开发者的手中. Android将数字证书用来标识应用程序的作者和在应用程序之间建立信任关系,不是用来决定最终用户可以安装哪些应用程序. 这个数字证书并不需要权威的数字证书签名机构认证,它只是用来让应用程序包自我认证的

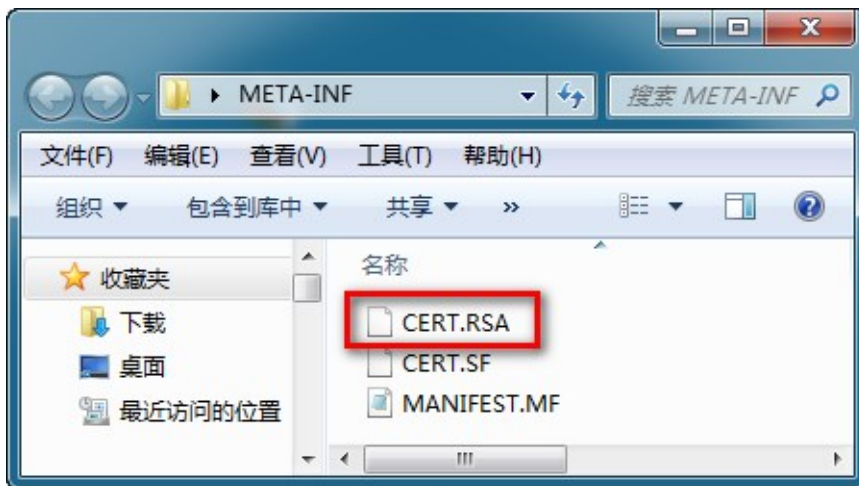
Android数字证书包含以下几个要点

1. 所有的应用程序都必须有数字证书,Android系统不会安装一个没有数字证书的应用程序
2. Android程序包使用的数字证书可以是自签名的,不需要一个权威的数字证书机构签名认证
3. 数字证书都是有有效期的,Android只是在应用程序安装的时候才会检查证书的有效期. 如果程序已经安装在系统中,即使证书过期也不会影响程序的正常功能
4. 签名保存在: META-INF文件夹中, meta-inf是项目发布时用来存储一些辅助的信息文件的

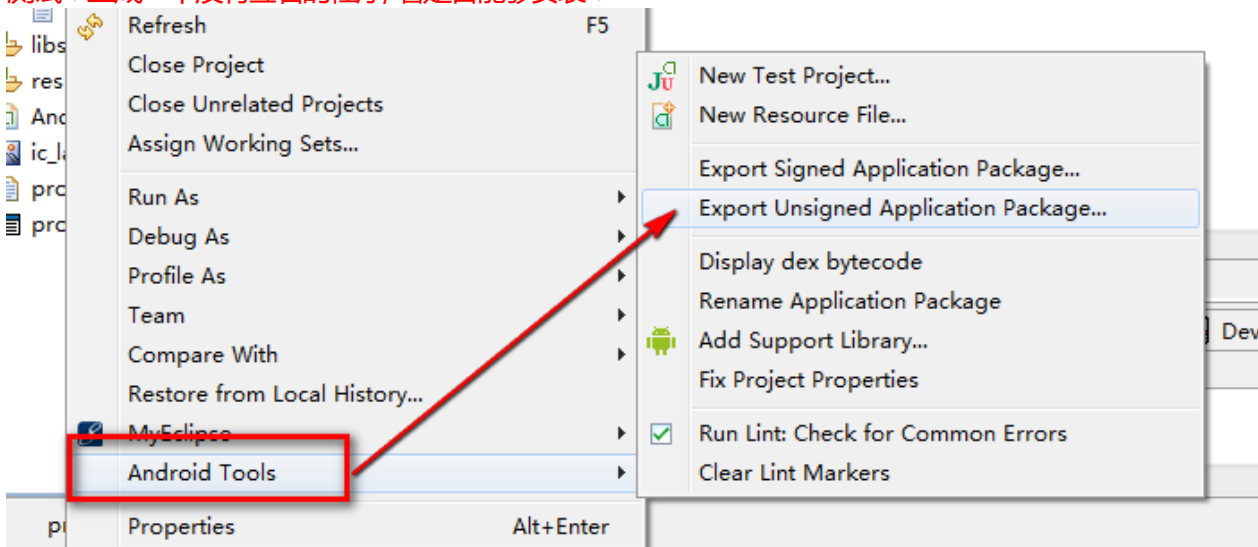
在开发的过程中, IDE工具会自动提供一个debug签名. debug签名为一年,而且不同的IDE的keystore不同



签名文件存储和查看的位置:



测试：生成一个没有签名的程序，看是否能够安装：

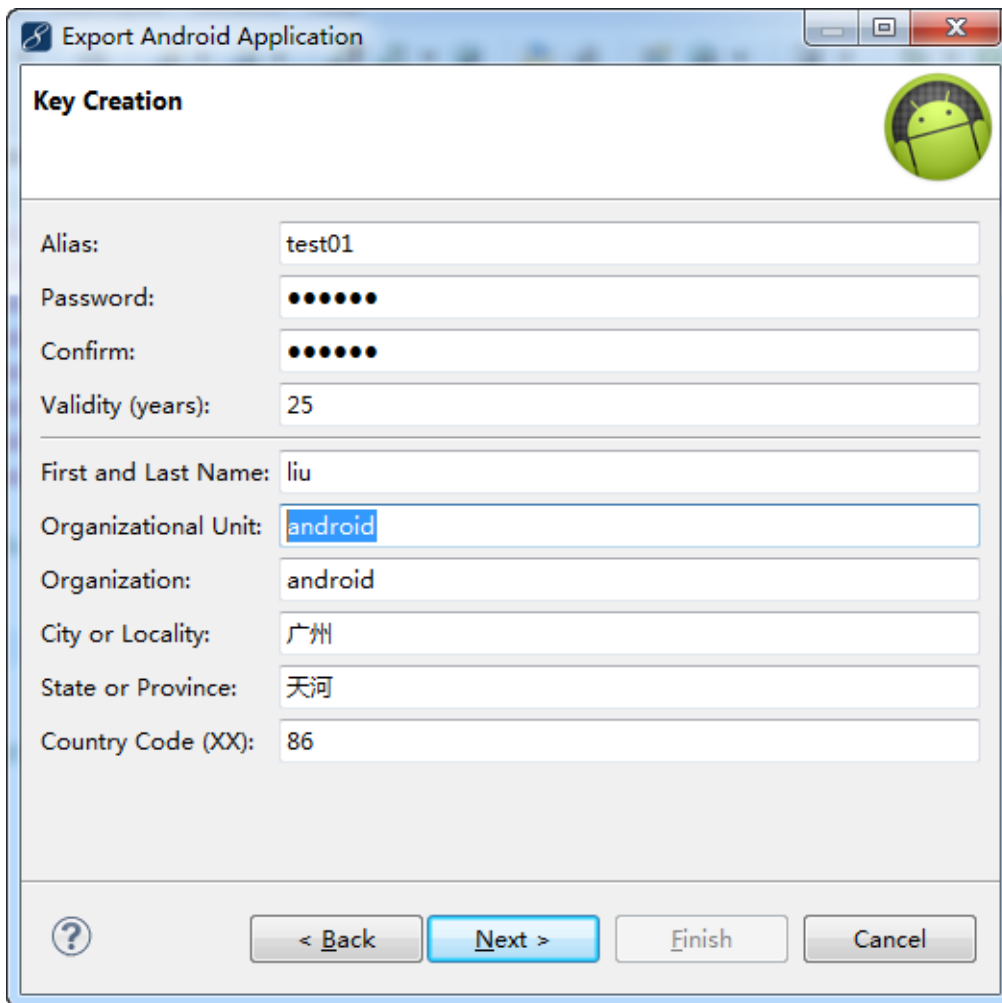


测试结果：在安装的时候提示没有缺少签名

```
C:\Users\Administrator>adb install C:\Users\Administrator\Desktop\project.apk
99 KB/s (253717 bytes in 2.502s)
  pkg: /data/local/tmp/project.apk
Failure [INSTALL_PARSE_FAILED_NO_CERTIFICATES]

C:\Users\Administrator>
```

自定义签名，取代默认的debug签名：设置完毕后就会导出签名文件和相应的apk目录



1. 在相同包名+签名的情况下. 采用 `adb install -r "apk路径"` 可以覆盖原来的程序, 或者可以先卸载原来的程序然后在重装
2. 在包名不同 + 签名相同的情况下. 可以正常安装. 理解为同一个公司开发的两个不同的APP应用程序
3. 包名相同 + 签名不同 (创建两个不同项目测试) 提示: Failure [INSTALL_FAILED_ALREADY_EXISTS]
4. 包名不同 + 签名不同的情况: 不测试了, 你懂的!

那开发android程序的人那么多, 那怎能避免包名重复呢, 比如公司为iteye, 项目为test, 那包名为com.iteye.test, 这样一般不会重复了。因此良好的包命名规则是程序开发的必要保证

Activity的横竖布局设置

回顾Activity的什么周期(通过"返回", "主页", `ctrl+F11`)来测试Activity的生命周期

可以在layout同目录下配置: `layout-land` 则可以自动加载横竖屏的布局操作

```
1.  @Override
2.  protected void onCreate(Bundle savedInstanceState) {
3.      super.onCreate(savedInstanceState);
4.      setContentView(R.layout.activity_main);
5.      Log.i("test", "正在初始化, 用于加载数据, 此时Activity并不能获取焦点!");
6.  }
7.
```

```
8.  @Override
9.  protected void onResume() {
10.     super.onResume();
11.     Log.i("test", "Activity已经初始化成功,并成为当前Activity");
12. }
```

我们有时候需要切换横竖屏的时候 都是一个布局, 也就是说不需要重新创建Activity 此时需要在activity中配置: `configChanges`属性

```
1.  <activity
2.      android:name="com.example.demo01.MainActivity"
3.      android:label="@string/app_name"
4.      android:configChanges="screenSize|orientation|keyboardHidden">
5.      <intent-filter>
6.          <action android:name="android.intent.action.MAIN" />
7.          <category android:name="android.intent.category.LAUNCHER" />
8.      </intent-filter>
9.  </activity>
```

有时候我们需要固定横、竖屏此时需要设置 `android:screenOrientation="portrait|landscape"`

```
1.  <activity
2.      android:name="com.example.demo01.MainActivity"
3.      android:label="@string/app_name"
4.      android:screenOrientation="portrait" >
5.  </activity>
```