# 在Androidmenifest.xml中配置版本信息

## android:versionCode

整型,表示应用程序代码的版本.

其他应用程序可以在程序编码中判断它, 例如检查升级或降级关系.

可以对其设置任何值, 然而应当确保每一个后续发布的应用程序使用更大的值.

该值没有必要与用户可见的应用程序发布版本号有特别的相似之处.

应用程序发布平台不应将此版本值呈现给用户

## android:versionName

字符串类型，向用户显示的应用程序代码的发布版本号.

可以用<major>.<minor>.<point>的形式描述应用程序版本, 或者以任何其他类型的绝对或相对版本号标识的形式.

相对于 android:versionCode, 除了让应用程序向用户显示它以外, 系统不打算将这个值用于任何内部使用目的（即是：该值只是用来展示，并不应参与到版本升降级逻辑中）.

## 一个例子

<?xml version="1.0" encoding="utf-8"?>

<manifest xmlns:android="http://schemas.android.com/apk/res/android" package="com.example.package.name"

android:versionCode="2"

android:versionName="1.1">

<application android:icon="@drawable/icon" android:label="@string/app\_name">

...

</application>

</manifest>

# 使用同一签名

Android系统要求每一个Android应用程序必须要经过数字签名才能够安装到系统中

Android通过数字签名来标识 应用程序的作者 和 在应用程序 的关系，

不是用来决定最终用户可以安装哪些应用程序

这个数字签名由应用程序的作者完成，并不需要权威的数字证书签名机构认证

## 同一签名

一个应用的所有版本，均应使用同一签名

不使用同一签名的后果：

手机上已经装了一个包名为a的应用，当用户下载了该应用的升级版本（包名仍然是a），要安装时，提示：已经安装过的应用的签名 与 即将要安装的应用的签名 不相同，如果要继续安装，必须要卸载掉之前已安装的应用

## 打包

网上有很多，使用eclipse打签名包的例子，在此，提供一个：

<http://www.yoyong.com/archives/62>

值得注意的是：

在后续版本的打包过程中，需要使用现有的keystore，如下图：

