



打包完成后显示为如下：

```

C:\MSR: C:\MSR\android\system\resources\
er, this can lead to crashes when the drawable is queried in a configuration that does not match this qualifier [MissingDefaultResource].

Explanation for issues of type "MissingDefaultResource".
If a resource is only defined in folders with qualifiers like "-land" or "-en,
and there is no default declaration in the base folder (layout or values
etc), then the app will crash if that resource is accessed on a device
where the device is in a configuration missing the given qualifier.

As a special case, drawables do not have to be specified in the base
folder if there is a match in a density folder (such as drawable-xhdpi)
that image will be used and scaled. Note however that if you only specify
a match in a folder like drawable-en-hdpi, the app will crash in
non-English locales.

There may be scenarios where you have a resource, such as a -fr drawable,
which is only referenced from some other resource with the same qualifiers
(such as a <string>), which itself has safe fallbacks. However, this still
makes it possible for somebody to accidentally reference the drawable and
crash, so it is safer to create a default dummy fallback in the base
folder. Alternatively, you can suppress the issue by adding
tools:ignore="MissingDefaultResource" on the element.

(This scenario frequently happens with string translations, where you might
delete code and the corresponding resources, but forget to delete a
translation. There is a dedicated issue id for that scenario, with the id
ExtraTranslation.)

12 errors, 0 warnings

Deprecated Gradle features were used in this build, making it incompatible with Gradle 7.0.
See https://docs.gradle.org/6.5/userguide/command_line_interface.html#sec:command_line_warnings
BUILD SUCCESSFUL in 10s
43 actionable tasks: 4 executed, 39 up-to-date
Build the following apk(s):
C:\GreaterRES\templates\src-cordova\platforms\android\app\build\outputs\apk\release\app-release-unsigned.apk

Build succeeded

Build mode..... cordova
Pre qasar..... v1.15.21
Pre qasar/apk..... v2.2.10
Debugging..... no
Publishing..... no
Packaging mode..... cordova
Transpiled JS..... yes (babel)
*****
Output folder..... C:\GreaterRES\templates\dist\cordova

Tip: "src-cordova" is a Cordova project folder, so everything you know
about Cordova applies to it. Qasar CLI only generates UI the content
for "src-cordova\www" folder and then Cordova takes over and builds
the final packaged file.

Tip: Feel free to use Cordova CLI ("cordova <params>") or change any files
in "src-cordova", except for "www" folder which must be built by Qasar CLI.

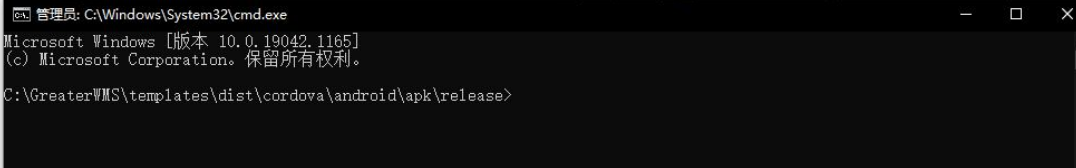
C:\GreaterRES\templates>

```

打包成功完成后打包好的文件会存放在项目的
GreaterWMS\templates\dist\cordova\android\apk\release 下的
app-release-unsigned.apk

2.生成签名文件

然后在刚才打包好的文件目录下启动 cmd



管理员: C:\Windows\System32\cmd.exe

Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation. 保留所有权利。

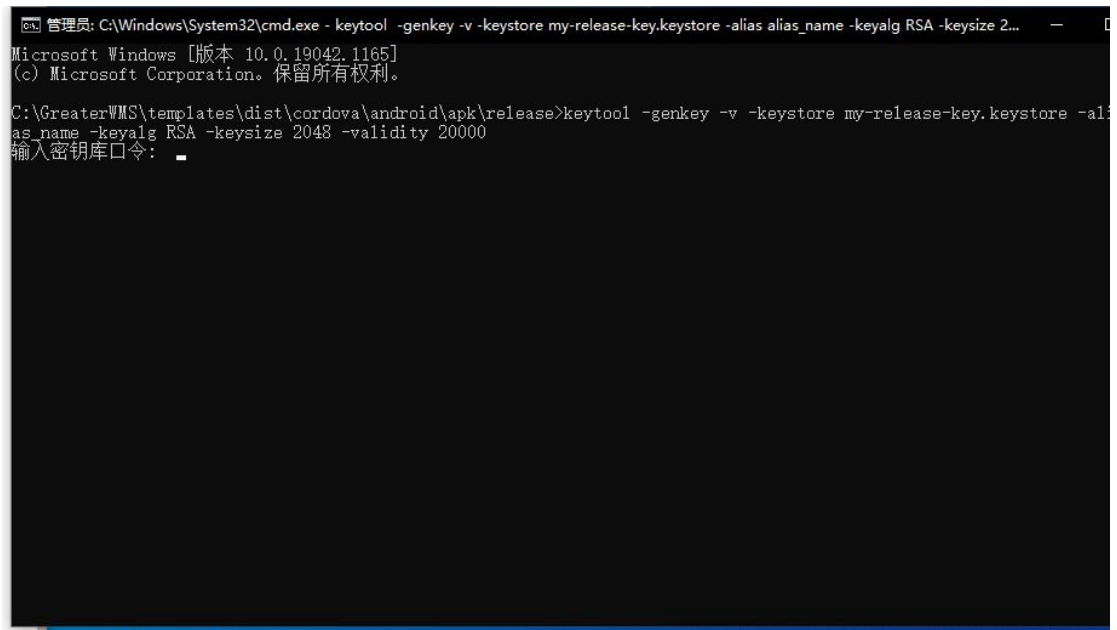
C:\GreaterWMS\templates\dist\cordova\android\apk\release>

输入：keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 20000

注释：其中 my-release-key 为稍后生成 keystore 的文件名（可自定义），alias_name 为生成 apk 别名（可自定义），

回车后会出现让你输入密钥库口令，然后自己设置口令，然后牢记

（在输入口令是光标不会变化 不过你输入的口令已经输进去了的所以记住输入的是啥就好）



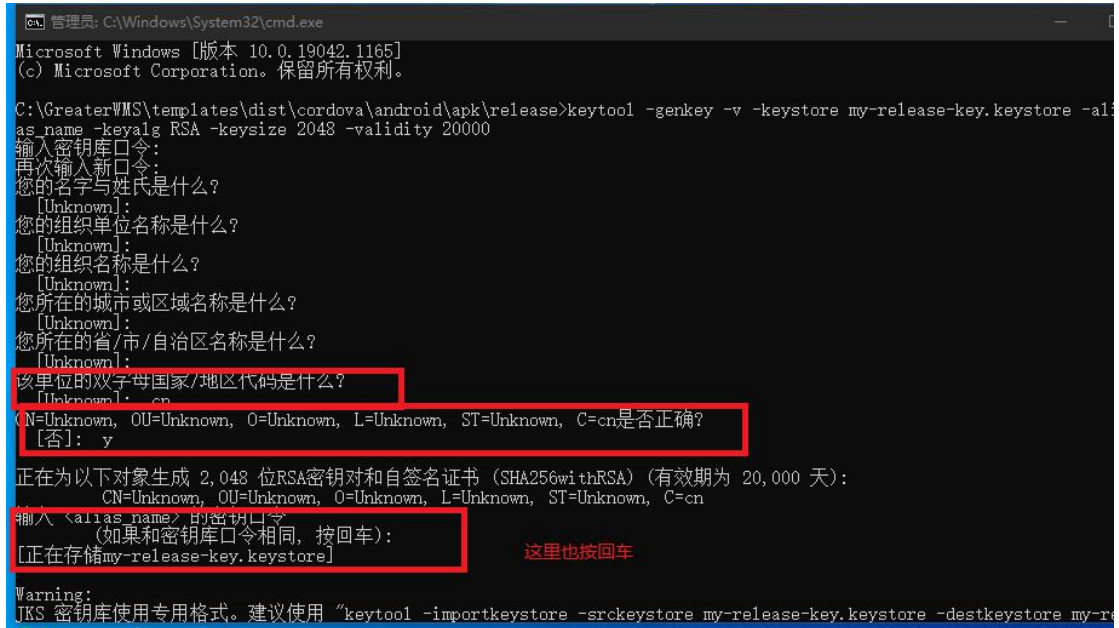
```
管理员: C:\Windows\System32\cmd.exe - keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 20000
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

C:\GreaterVMS\templates\dist\cordova\android\apk\release>keytool -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -keysize 2048 -validity 20000
输入密钥库口令: 
```

然后回车后，会让你确认一遍 按照提示 再次输入一遍口令

随后回车后回依次出现图中的内容，不用输入内容依次回车就好了

注意：在国家的地方需要输入 **cn**（中国地区的意思），是否正确输入 y



```
C:\Windows\System32\cmd.exe
Microsoft Windows [版本 10.0.19042.1165]
(c) Microsoft Corporation。保留所有权利。

C:\GreaterWMS\templates\dist\cordova\android\apk\release>keytool -genkey -v -keystore my-release-key.keystore -alias_name -keyalg RSA -keysize 2048 -validity 20000
输入密钥库口令:
再次输入新口令:
您的名字与姓氏是什么?
[Unknown]:
您的组织单位名称是什么?
[Unknown]:
您的组织名称是什么?
[Unknown]:
您所在的城市或区域名称是什么?
[Unknown]:
您所在的省/市/自治区名称是什么?
[Unknown]:
该单位的双字母国家/地区代码是什么?
[Unknown]: cn
(N=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=cn)是否正确?
[否]: y

正在为以下对象生成 2,048 位RSA密钥对和自签名证书 (SHA256withRSA) (有效期为 20,000 天):
CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=cn
输入 <alias_name> 的密钥口令:
(如果和密钥库口令相同, 按回车):
[正在存储my-release-key.keystore] 这里也按回车

Warning:
JKS 密钥库使用专用格式。建议使用 "keytool -importkeystore -srckeystore my-release-key.keystore -destkeystore my-release-key.keystore" 命令来转换。
```

结束后会在刚刚的目录下会新生成一个 **keystore** 文件

然后继续在 **cmd** 命令行输入：

jarsigner -verbose -keystore （签名文件后的 keystore 文件名）**.keystore**
-signedjar （生成签名的 apk 名，也就是自定义）**.apk app-release-unsigned**（未签名之前的 apk 文件名）**.apk alias_name**(别名)

其中带括号的都是需要手动输入的文件名

解释：**jarsigner -verbose -keystore** [您的私钥存放路径] **-signedjar** [签名后文件存放路径] [未签名的文件路径] [您的证书的别名]

演示如下：

```
C:\GreaterWMS\templates\dist\cordova\android\apk\release>jarsigner -verbose -keystore my-release-key.keystore -signedjar
wms.apk app-release-unsigned.apk alias_name
输入密钥库的密码短语:
正在更新: META-INF/MANIFEST.MF
正在添加: META-INF/ALIAS.NA.SF
正在添加: META-INF/ALIAS.NA.RSA
正在签名: AndroidManifest.xml
正在签名: assets/www/cordova-js-src/android/nativeapiprovider.js
正在签名: assets/www/cordova-js-src/android/promptbasednativeapi.js
正在签名: assets/www/cordova-js-src/exec.js
正在签名: assets/www/cordova-js-src/platform.js
正在签名: assets/www/cordova-js-src/plugin/android/app.js
正在签名: assets/www/cordova.js
正在签名: assets/www/cordova_plugins.js
正在签名: assets/www/css/3.f8666498.css
正在签名: assets/www/css/app.bb7b8dbb.css
正在签名: assets/www/css/vendor.ae9dbbef.css
正在签名: assets/www/fonts/f1UhRq6tzZc1QeJ-Vdg-IuiaDsNa.woff
正在签名: assets/www/fonts/f1UhRq6tzZc1QeJ-Vdg-IuiaDsNcIhQ8tQ.woff2
正在签名: assets/www/index.html
正在签名: assets/www/js/1.js
正在签名: assets/www/js/10.js
正在签名: assets/www/js/11.js
正在签名: assets/www/js/12.js
正在签名: assets/www/js/13.js
正在签名: assets/www/js/14.js
正在签名: assets/www/js/15.js
正在签名: assets/www/js/16.js
正在签名: assets/www/js/17.js
```

运行完成后如图表示打包签名成功

```
C:\GreaterWMS\templates\dist\cordova\android\apk\release>jarsigner -verbose -keystore my-release-key.keystore -signedjar
wms.apk app-release-unsigned.apk alias_name
输入密钥库的密码短语:
正在签名: res/mipmap-ldpi-v26/ic_launcher_foreground.png
正在签名: res/mipmap-ldpi-v4/ic_launcher.png
正在签名: res/mipmap-mdpi-v26/ic_launcher.xml
正在签名: res/mipmap-mdpi-v26/ic_launcher_background.png
正在签名: res/mipmap-mdpi-v26/ic_launcher_foreground.png
正在签名: res/mipmap-mdpi-v4/ic_launcher.png
正在签名: res/mipmap-xhdpi-v26/ic_launcher.xml
正在签名: res/mipmap-xhdpi-v26/ic_launcher_background.png
正在签名: res/mipmap-xhdpi-v26/ic_launcher_foreground.png
正在签名: res/mipmap-xhdpi-v4/ic_launcher.png
正在签名: res/mipmap-xxhdpi-v26/ic_launcher.xml
正在签名: res/mipmap-xxhdpi-v26/ic_launcher_background.png
正在签名: res/mipmap-xxhdpi-v26/ic_launcher_foreground.png
正在签名: res/mipmap-xxhdpi-v4/ic_launcher.png
正在签名: res/mipmap-xxxhdpi-v26/ic_launcher.xml
正在签名: res/mipmap-xxxhdpi-v26/ic_launcher_background.png
正在签名: res/mipmap-xxxhdpi-v26/ic_launcher_foreground.png
正在签名: res/mipmap-xxxhdpi-v4/ic_launcher.png
正在签名: res/xml/config.xml
正在签名: resources.arsc
>>> 签名者
X.509, CN=Unknown, OU=Unknown, O=Unknown, L=Unknown, ST=Unknown, C=cn
[可信证书]
jar 已签名。
警告:
签名者证书为自签名证书。
C:\GreaterWMS\templates\dist\cordova\android\apk\release>
```

此时在我们的目录下会出现一个刚刚我们输入自定义文件名的 **apk** 文件，这就是我们最终签名好的 **apk**，这个就是可以供我们安装在安卓终端的安装包！