Generating an APK in the Debug Mode

Besides generating signed packages for deploying on physical devices or unsigned packages for running on emulators, you can also have an application package extracted and signed in the debug mode. This signature is sufficient for testing and debugging applications but does not allow publishing them. Signing an application package in the *debug mode* is available only through an artifact.

In the *debug mode*, you can have an APK signed either with the *default* certificate or with a *custom* one.

If you decide to use the *default certificate*, IntelliJ IDEA signs the extracted package in the *debug mode* using the debug keystore or key that is generated by the Android SDK tools and has the predefined names and passwords:

- **Keystore name:** `debug.keystore`
- **Keystore password:** `android`
- **Key alias:** `androiddebugkey`
- **Key password:** `android`
- **CN** (*common name*): `CN=Android Debug,O=Android,C=US`

This setting is default in IntelliJ IDEA. This means that if you do not configure any artifact manually and choose **Deploy default APK** in the Run/Debug Configuration: Android Application dialog box, IntelliJ IDEA will use the predefined values in the certificate for the generated `.apk`.

If you use a *custom certificate*, IntelliJ IDEA signs the extracted package in the *debug mode* using the debug keystore or key that you specify yourself . You can have a new certificate generated or reuse an existing one. The latter approach is helpful, for example, if you have several applications and you want them all signed with he same certificate so they can be stored in the same folder on the device.

Signing a package in the debug mode

1. Open the Project Structure settings.
2. Click **Artifacts** to open the Artifacts page.
3. Click the **New** button + and choose the artifact type **Android Application** in the **New** drop-down list. Then specify the original contents of the artifact definition using the context menu:
   - To have an empty layout definition created. choose **Empty**.
   - To have the data of a module included in the artifact by default, choose **From module <module name>**.

   The basic settings of the new artifact are displayed in the **Artifact Layout** pane that opens.
4. Specify the general settings of the artifact. In the **Output directory** text box, specify the location of the package `.apk` file to be created.
5. Complete the artifact definition by following these general steps:
   - Configure the artifact structure.
   - Add resources to the artifact.
   - Arrange the elements included in the artifact.
   - If necessary, specify additional activities to be performed before and after building the artifact in the Pre-processing and Post-Processing tabs.

6. In the Android tab, specify the certificate to use:
   - Choose **Debug signed with default certificate** to have IntelliJ IDEA use the debug keystore or key that is generated by the Android SDK tools and has the predefined names and passwords.
   - Choose **Debug signed with custom certificate** to have the package signed with a certificate of your choice. Specify the key to use and the keystore file that contains it by doing one of the following:
     - To have the package signed with a key from a previously generated keystore file:
       1. Specify the file location in the **Keystore path** text box. Type the path manually or click the **Choose existing** button and choose the file in the dialog that opens. In the **Password** text box, type your password to access the selected keystore.
       2. Specify the key alias and the password to access the key.
     - To have a new key generated in an existing keystore:
       1. Specify the keystore location and password in the **Keystore path** and the **Password** text boxes respectively.
       2. Click the **Create new** button and configure the release key to be generated by filling in the data in the New Key Store dialog box that opens.
     - To have a new keystore file with a new key generated:
       1. Click the **Create new** button. In the New Key Store dialog box that opens, specify the location of the file to be generated in the **Key store path** text box. Type the path manually or click the **Browse** button …, then select the parent folder and specify the name of the file.
       2. Specify and confirm the password to access the keystore.
       3. Configure the release key to be generated by filling in the data in the **Key** area. The procedure is identical no matter whether you are using a wizard or an artifact definition, see Generating a Signed Release APK Using a Wizard: Generating a new release key for details.
       4. Click **OK** when ready. IntelliJ IDEA brings you back to the **Android** tab, where the keystore location, the key alias and passwords to access them are filled in in the corresponding fields.
7. To have IntelliJ IDEA obfuscate the application during packaging, select the **Run ProGuard** check box and specify the location of the proguard.txt configuration file. The file is generated during project creation and is stored in the project root. IntelliJ IDEA suggests this default location in the **Config file path** text box. Accept the suggestion or specify a custom configuration file by clicking the **Browse** button … and selecting the required file in the dialog that opens.

See Also

Procedures:

- Generating Signed and Unsigned Android Application Packages
- Generating an Unsigned Release APK
- Generating an APK in the Debug Mode
- Android

Reference:

- Generate Signed APK Wizard
- Android Reference

External Links:

- http://developer.android.com/guide/publishing/app-signing.html

Web Resources:

- Developer Community