

# Generating a Signed Release APK Through an Artifact

---

To deploy and run an Android application on a physical device, you need to [sign the application digitally](#). With IntelliJ IDEA, you can have your Android Application Package (.apk file) signed with an existing release key during the package extraction. IntelliJ IDEA also incorporates a release key generation tool that can be invoked during packaging. Generated keys are saved in a keystore binary file.

You can have as many keystore files and keys as you need and use either previously created keys, or create new ones in existing keystores, or even create new keystores.





Besides using the [Generate Signed APK Wizard](#), you can configure the .apk file as an artifact by creating an [Android application artifact](#) definition. When IntelliJ IDEA builds the package in accordance with this definition, the package is signed automatically.

## Extracting and signing a release Android application package using an artifact definition

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, choose **Release signed**.

7. Specify the release key to use and the keystore file that contains it. Do 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.
8. 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 a Signed Release APK Using a Wizard](#)
- [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](#) 

