

# Generate Signed APK Wizard

---

## Build | Generate Signed APK

---

To deploy and run an Android Application package ([.apk file](#)) on physical devices, you need to [sign](#) it with your personal signature (certificate). Based on this signature, the Android system identifies the author of every deployed application. You do not need to apply for a personal signature to any authority, a signature generated by IntelliJ IDEA is quite sufficient.

Use this *Extract a Signed Android Application Package Wizard* to have IntelliJ IDEA [digitally sign](#) Android Packages ([.apk files](#)) during package extraction. You can use previously created signature keys, create new ones in existing keystores, as well as create new keystores.

Alternatively, configure the `.apk` file as an [artifact](#) by creating an [artifact definition](#) of the type *Android application* with the **Release signed** package mode turned on. When IntelliJ IDEA builds the package in accordance with this definition, it signs the package automatically.

An *unsigned package* is used when you want to test your application on an emulator. Unsigned packages can be extracted only through artifact definitions with the **Release unsigned** package mode turned on.

You can also extract and sign *debug* packages. This is sufficient for testing and debugging applications but does not allow publishing them. Signing packages in the debug mode is available only through an artifact with the **Debug** package mode turned on.

In this section:

- [Generate Signed APK Wizard. Specify Key and Keystore](#)
- [New Key Store Dialog](#)
- [Generate Signed APK Wizard. Specify APK Location](#)

### See Also

Procedures:

- [Generating Signed and Unsigned Android Application Packages](#)
- [Android](#)
- [Configuring Artifacts](#)

Reference:

- [Android Reference](#)
- [Artifacts](#)

External Links:

- <http://developer.android.com/tools/publishing/app-signing.html#signing>

Web Resources:

- [Developer Community](#)