Running and Debugging Android Applications

Basically, the Android platform is intended for running applications on *Android* mobile phones and other devices. However, for a number of reasons you may find it helpful to at least debug applications using a virtual device emulation, which are supported by the *Android SDK* and are available via IntelliJ IDEA.

On this page:

Running or debugging an entire Android application

1. Start creating an Android run/debug configuration, see Creating an Android Run/Debug Configuration.
2. On the Run/Debug Configuration: Android page that opens, specify the configuration name and choose the module to apply the current configuration to.
3. Choose **Deploy default APK** in the **Package** area and **Launch default activity** in the **Activity** area. IntelliJ IDEA will upload the `.apk` built from the module specified in the **Module** drop-down list above and launch the activity marked as start-up for the chosen module. The `.apk` is built automatically, no preliminary artifact configuration is required from your side.
4. Appoint the device to run:
   - To use an emulation, choose the desired virtual device or create a new one.
   - To use an Android mobile phone or another physical device, attach it according to the model-specific instructions.
5. Start running or debugging the desired activity.
   - If a virtual device has been specified in the current configuration, continue running or debugging.
   - If no emulation has been specified in the run/debug configuration, choose the target device manually.
6. View and analyze Android system messages in the **Logcat** tab of the Android tool window.

Running or debugging a custom .apk that will be later embedded in an application

1. Configure an artifact to generate the .apk from:
   1. Open the **Project Structure** dialog box by choosing **File | Project Structure** . See Accessing Project Structure and Accessing Module Settings for details.
   2. In the left-hand pane, click **Artifacts**.
   3. In the central pane, click the **Add** button + on the toolbar. From the list of available artifact types, choose **Android Application**, and then choose **Empty** on the context menu.
   4. In the right-hand pane, add the artifact components. The artifact should contain all the resources and code that you need packaged in the .apk. For details, see Generating a Signed Release APK Through an Artifact and Configuring Artifacts.
2. Start creating an Android run/debug configuration, see Creating an Android Run/Debug Configuration.
3. On the Run/Debug Configuration: Android page that opens, specify the configuration name and choose the module to apply the current configuration to.
4. To run or debug a custom .apk that will be later embedded in an application, choose **Deploy custom artifact** in the **Package** area and choose the artifact to build the .apk from. In this case, you have to define the relevant artifact manually before creating a run/debug configuration, see Generating a Signed Release APK Through an Artifact and Configuring Artifacts. Then choose the **Launch** option in the **Activity** area and specify the start-up activity from the chosen artifact (.apk). Type the activity name manually or click the **Browse** button ... and select the desired activity in the **Select Activity Class** dialog box, that opens.

   The list of available activities is determined by the choice of the module.
5. Appoint the device to run:
   - To use an emulation, choose the desired virtual device or create a new one.
   - To use an Android mobile phone or another physical device, attach it according to the model-specific instructions.
6. Start running or debugging the desired activity.
   - If a virtual device has been specified in the current configuration, continue running or debugging.
   - If no emulation has been specified in the run/debug configuration, choose the target device manually.
7. View and analyze Android system messages in the **Logcat** tab of the Android tool window.

Debugging an already running application

Besides debugging an Android application by initiating a debugging session, you can apply the debugger to an already running application. For details, see Debugging a Running Application.

In this section

- Creating an Android Run/Debug Configuration
- Choosing the Target Device Manually
- Configuring Device Layout
- Debugging with Logcat
- Debugging a Running Application
- Managing Virtual Devices

See Also

Concepts:

- Run/Debug Configuration

Procedures:

- Creating an Android Run/Debug Configuration
- Managing Virtual Devices
- Choosing the Target Device Manually
- Android

Reference:

- Run/Debug Configuration: Android Application
- Choose Device Dialog
- Select Android Virtual Device Dialog
- Create Android Virtual Device Dialog
- Android Reference

Web Resources:

- Developer Community