

24. A Guide to using Instant Run in Android Studio

Now that some of the basic concepts of Android development using Android Studio have been covered, now is a good time to introduce the Android Studio Instant Run feature. As all experienced developers know, every second spent waiting for an app to compile and run is time better spent writing and refining code.

24.1 Introducing Instant Run

Prior to the introduction of Instant Run, each time a change to a project needed to be tested Android Studio would recompile the code, convert it to Dex format, generate the APK package file and install it on the device or emulator. Having performed these steps the app would finally be launched ready for testing. Even on a fast development system this is a process that takes a considerable amount of time to complete. It is not uncommon for it to take a minute or more for this process to complete for a large application.

Instant Run, in contrast, allows many code and resource changes within a project to be reflected nearly instantaneously within the app while it is already running on a device or emulator session.

Consider, for the purposes of an example, an app being developed in Android Studio which has already been launched on a device or emulator. If changes are made to resource settings or the code within a method, Instant Run will push the updated code and resources to the running app and dynamically “swap” the changes. The changes are then reflected in the running app without the need to build, deploy and relaunch the entire app. In many cases, this allows changes to be tested in a fraction of the time it would take without Instant Run.

24.2 Understanding Instant Run Swapping Levels

Not all project changes are fully supported by Instant Run and different changes result in a different level of “swap” being performed. There are three levels of Instant Run support, referred to as hot, warm and cold swapping:

- **Hot Swapping** – Hot swapping occurs when the code within an

existing method implementation is changed. The new method implementation is used next time it is called by the app. A hot swap occurs instantaneously and, if configured, is accompanied by a toast message on the device screen that reads “Applied code changes without activity restart”.

- **Warm Swapping** – When a change is made to a resource file of the project (for example a layout change or the modification of a string or color resource setting) an Instant Run warm swap is performed. A warm swap involves the restarting of the currently running activity. Typically the screen will flicker as the activity restarts. A warm swap is reported on the device screen by a toast message that reads “Applied changes, restarted activity”.
- **Cold Swapping** – Structural code changes such as the addition of a new method, a change to the signature of an existing method or a change to the class hierarchy of the project triggers a cold swap in which the entire app is restarted. In some conditions, such as the addition of new image resources to the project, the application package file (APK) will also be reinstalled during the swap.

24.3 Enabling and Disabling Instant Run

Instant Run is enabled and disabled via the Android Studio Settings screen. To view the current settings begin by selecting the *File -> Settings...* menu option (*Android Studio -> Preferences...* on macOS). Within the Settings dialog select the *Build, Execution, Deployment* entry in the left-hand panel followed by *Instant Run* as shown in [Figure 24-1](#):

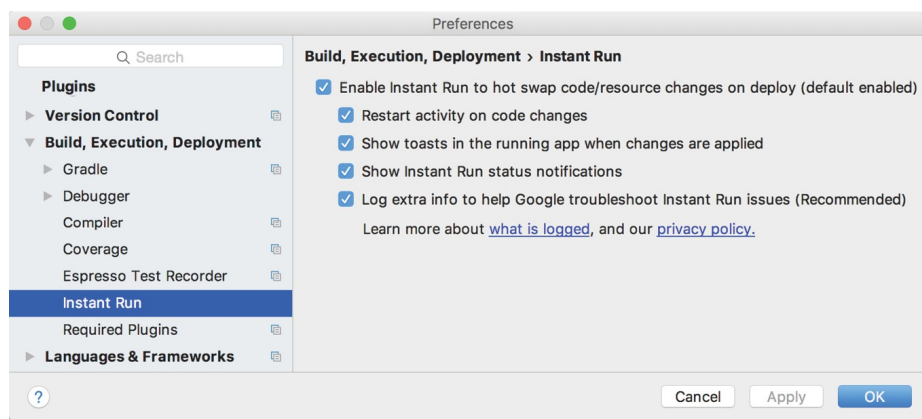


Figure 24-1

The options provided in the panel apply only to the current project. Each new project will start with the default settings. The first option controls whether or not Instant Run is enabled by default each time the project is opened in Android Studio. The *Restart activity on code changes* option forces Instant Run to restart the current activity every time a change is made, regardless of whether a hot swap could have been performed. The next option controls whether or not messages are displayed within Android Studio and the app indicating the type of Instant Run level performed. Finally, an option is provided to allow additional log information to be provided to Google to help in improving the reliability of the Instant Run feature.

24.4 Using Instant Run

When a project has been loaded into Android Studio, but is not yet running on a device or emulator, it can be launched as usual using either the run (marked A in [Figure 24-2](#)) or debug (B) button located in the toolbar:

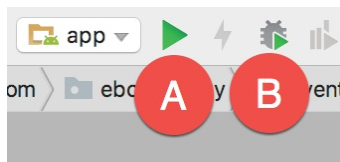


Figure 24-2

After the app has launched and is running, Android Studio will indicate the availability of Instant Run by enabling the *Apply Changes* button located immediately to the right of the run button as highlighted in [Figure 24-3](#):

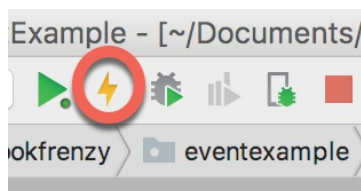


Figure 24-3

When it is enabled, clicking on the Apply Changes button will use Instant Run to update the running app.

24.5 An Instant Run Tutorial

Begin by launching Android Studio and creating a new project. Within the *New Project* dialog, enter *InstantRunDemo* into the Application name field and *ebookfrenzy.com* as the Company Domain setting before clicking on the *Next* button.

On the form factors screen, enable the *Phone and Tablet* option and set the minimum SDK setting to API 23: Android 6.0 (Marshmallow). Continue to proceed through the screens, requesting the creation of a Basic Activity named *InstantRunDemoActivity* with a corresponding layout named *activity_instant_run_demo*.

Click on the *Finish* button to initiate the project creation process.

24.6 Triggering an Instant Run Hot Swap

Begin by clicking on the run button and selecting a suitable emulator or physical device as the run target. After clicking the run button, track the amount of time before the example app appears on the device or emulator.

Once running, click on the action button (the button displaying an envelope icon located in the lower right-hand corner of the screen). Note that a Snackbar instance appears displaying text which reads “Replace with your own action” as shown in [Figure 24-4](#):

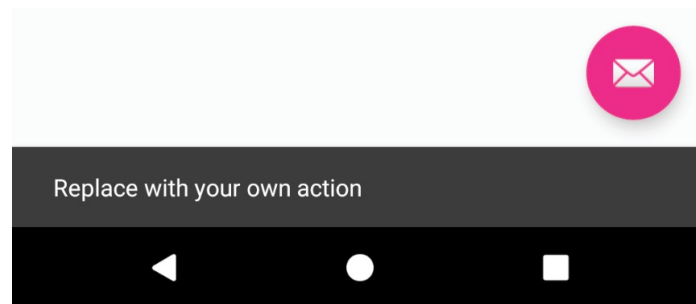


Figure 24-4

Once the app is running, the Apply Changes button should have been enabled indicating the availability of Instant Run. To see this in action, edit the *InstantRunDemoActivity.java* file, locate the *onCreate* method and modify the action code so that a different message is displayed when the action button is selected:

```
FloatingActionButton fab = (FloatingActionButton)
findViewById(R.id.fab);
fab.setOnClickListener(new View.OnClickListener() {
    @Override
    public void onClick(View view) {
        Snackbar.make(view, "Instant Run is Amazing!",
            Snackbar.LENGTH_LONG)
            .setAction("Action", null).show();
    }
})
```

```
});
```

With the code change implemented, click on the Apply Changes button and note that the toast message appears within a few seconds indicating the app has been updated. Tap the action button and note that the new message is now displayed in the Snackbar. Instant Run has successfully performed a hot swap.

24.7 Triggering an Instant Run Warm Swap

Any resource change should result in Instant Run performing a warm swap. Within Android Studio select the *app* -> *res* -> *layout* -> *content_instant_run_demo.xml* layout file. With the Layout Editor tool in Design mode, select the ConstraintLayout view within the Component Tree panel, switch the Attributes tool window to expert mode and locate the *background* property. Click on the button displaying three dots next to the background property text field, select a color from the Resources dialog and click on OK. With the background color of the activity content modified, click on the Apply Changes button once again. This time a warm swap will be performed and the currently running activity should quickly restart to adopt the new background color setting.

24.8 Triggering an Instant Run Cold Swap

As previously described, a cold swap triggers a complete restart of the running app. To experience an Instant Run cold swap, edit the *InstantRunDemoActivity.java* file and add a new method after the *onCreate* method as follows:

```
public void demoMethod() {  
  
}
```

Click on the Apply Changes button and note that the app now has to terminate and restart to accommodate the addition of the new method. Within Android Studio a message will appear indicating that the app was restarted due to a method being added:

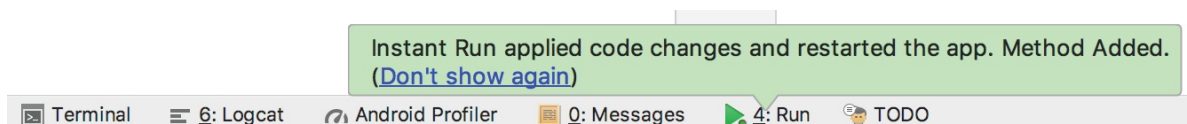


Figure 24-5

24.9 The Run Button

When no apps are running, the run button appears as shown in [Figure 24-2](#). When an app is running, however, an additional green dot appears in the bottom right-hand corner of the button as shown in [Figure 24-6](#) below:



Figure 24-6

Although the Instant Run feature has improved significantly since being introduced it can still occasionally produce unexpected results when performing hot or warm swaps. It is worth being aware, therefore, that clicking the run button when an app is currently running will force a cold swap to be performed regardless of the changes made to the project.

24.10 Summary

Instant Run is a feature of Android Studio designed to significantly accelerate the code, build and run cycle. Using a swapping mechanism, Instant Run is able to push updates to the running application, in many cases without the need to re-install or even restart the app. Instant Run provides a number of different levels of support depending on the nature of the modification being applied to the project. These levels are referred to as hot, warm and cold swapping. This chapter has introduced the concepts of Instant Run and worked through some demonstrations of the different levels of swapping.