

# Privacy indicators

Archived: 2026-04-05 17:40:17 UTC

Runtime permissions in Android 6 and higher give users control over when they allow audio from a device's microphone or video from a device's camera to be recorded. Before an app can record, a user must either grant or deny it permission through a dialog the system presents.

Android 12 provides users with transparency by displaying indicators when an app uses a *private* data source through the cameras and microphone [app-op permissions](#). App-ops record accesses to runtime permission-protected APIs.

App-op tracks both the activity status and the number of individual API calls, and interacts with the microphone and camera indicators in Android 12 to show users when apps have accessed audio and camera data on their devices. When users click the microphone or camera indicators, they see which apps have accessed their data. This feature is mandatory for all OEMs.

## Indicator visibility requirements

These are the requirements for microphone and camera indicators on devices running on Android 12 or higher:

- The indicators must appear in the status bar, and retain the highest visual priority (for example, located at the rightmost position in the top-right corner).
- The indicators must be consistently located in the same position, and must not be blocked by an app when it launches.
- Both indicators *must* be green (or a variation of green) in color.
- Clicking one or both indicators *must* render an app attribution affordance notification that does the following:
  - Displays the name of the app that's using the microphone and (or) camera
  - Displays the name of the app which used the microphone and (or) camera within the last 15 seconds
  - Takes users to the app permission page in **Settings**

## Usage and features

In Android 12, the UI distinguishes between running usages and recent usages. Usages are considered active if they're either marked by the system as running, or are less than 5 seconds old.

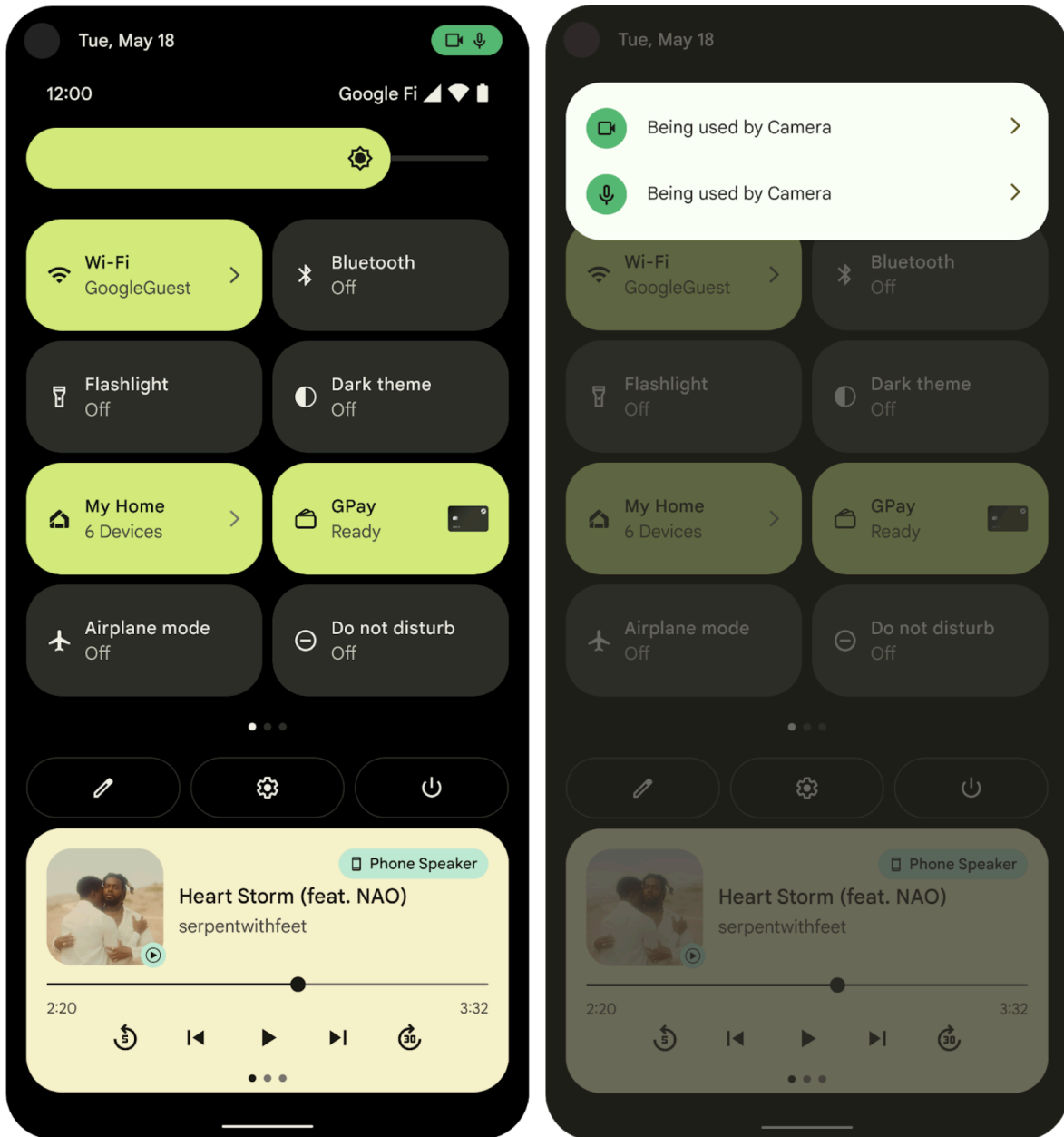
- Status bar icons show each time an app has ongoing user-sensitive microphone or camera access.
- Users can click these icons and see which apps are accessing the microphone, the camera, or both.



**Figure 1.** Microphone and camera in-use indicators showing active access (top-right corner)

Access is considered active for as long as the indicators show. An icon shows first, then transitions to a dot that persists until the app is either dismissed or closed.

When a user opens Quick Settings and then taps the indicators, a dialog appears that shows whether the app is using the camera, the microphone, or both.



**Figure 2.** Active and recent access indicators

The image in **Figure 2** shows active access indicators when data has been accessed in the past 5 seconds by a running app.

Recent access indicators show that data was accessed by an app during the prior 15 seconds, but that the app isn't active. All active apps show in the dialogue, but only one app shows as the source of recent access, even when more than 1 app accessed data within a prior 15 second time-frame. The access view remains frozen until the user dismisses the notification dialog.

### Deliver and enable

The `PermissionManager` class provides a method to populate the dialog, which resides in System UI.

- System UI reacts to a device configuration switch: `privacy/mic_camera_indicators_enabled` .

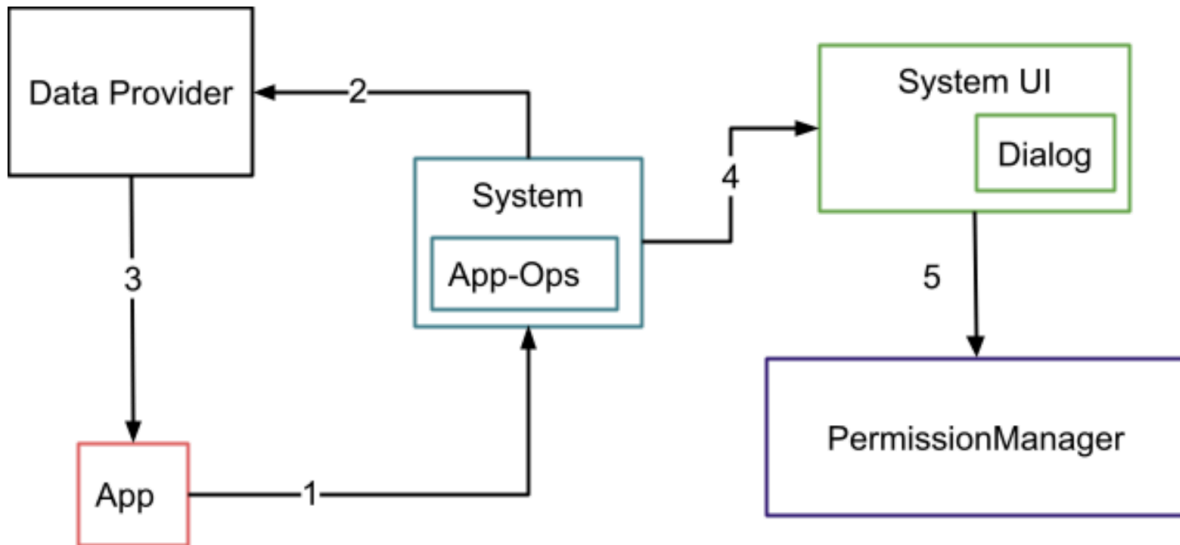
- The switch is necessary because there are two separate delivery vehicles, in this order:
  1. Deliver.
  2. Enable.
- The System UI must not crash if the method in `PermissionManager` can't provide the needed data.

## Process flow

The Permission Indicator functionality has three main parts:

- The apps
- The indicators (handled by SystemUI)
- A way to determine which apps are using the data

The `PermissionController` provides the mechanism to determine which apps are using data. SystemUI listens for apps using private data. The SystemUI shows an icon in the top navigation bar that corresponds to the permissions used. The `PermissionController` displays data about usages when a user clicks an icon.



**Figure 3.** System components and (UI) transitions flow

The numbered transitions in **Figure 3** are described below:

1- An app requests private data from the system.

2- The system checks permissions. If the permissions are allowed, the system notifies the data provider, and notes the usage in app-ops

3- The data provider gives data to the app.

4-5 The user clicks the icons. System UI requests data from the `PermissionManager` and presents the dialog to the user.

## Process details

1. Apps use the microphone and camera, calling `AppOpsManager.startOp` , `stopOp` , and (or) `noteOp` . This creates app-op records in the system server.
2. The System UI listens for new app-ops using the `AppOpsManager.OnOpActiveChangedInternalListener` and `OnOpNotedListener` listeners. When a new usage comes in (through a call to either `startOp` or `noteOp` ), the System UI verifies that the usage is by a system app.
3. If System UI verifies system app usage, and if the usage is for the microphone, System UI checks if the microphone is muted.
4. If System UI verifies non-system app usage, (and, for microphone usage, that the microphone is unmuted; for camera usage, that the camera is toggled on), it shows an icon reflecting such usage.

If System UI receives a `noteOp` , which has no duration, it shows the icon for at least 5 seconds. Otherwise, the icon shows either until it receives a `stopOp` , or for 5 seconds, whichever is longer. The user clicking an icon initiates an intent that goes to the `PermissionController` to start the Dialog.

The `PermissionController` loads all recent usage for microphone and camera. It checks if any of those are running, or were running within the timeframe set by System UI. If it finds a match, it displays the name of the app that used the permission, and which permissions the app used.

Because of this change in Android 12 and higher, some apps must either change their behavior or implement special behavior.

Telephony must implement permission usage (to account for the separate microphone stack used in phone calls), while working with Android Google Search app (AGSA) and Google Mobile Services (GMS).

---

Source: <https://source.android.com/devices/tech/config/privacy-indicators>