Sensors Off

When in airplane mode, devices can still access some sensors to enable specific functionality, such as screen rotation and taking pictures. Android 10 provides a developer options setting to shut off all sensors in a device. This feature helps developers test their app's functionality in situations where those sensors become unavailable, and also gives users a way to control the sensors in their device.

When a developer or user enables **Sensors off** in developer options (**Settings** > **System** > **Developer options** > **Quick settings developer tiles**), a new tile appears in the quick settings tray. They can use the tile to prevent apps from accessing the camera, microphone, and all sensors managed by the `SensorManager` class.

Implementation

Android 10 includes a reference implementation that handles the camera, microphone, and `SensorManager` sensors. The system service that manages the **Sensors off** state and notifies clients of state changes is located in `frameworks/base/services/core/java/com/android/server/SensorPrivacyService.java`. The manager that facilitates access to `SensorPrivacyService` within an application's context is located in `frameworks/base/core/java/android/hardware/SensorPrivacyManager.java`.

If your devices use the default implementation of `SensorService`, `CameraService`, and `AudioPolicyService`, then no additional customization is needed to the reference design. If you have other sensors, see Customization (#customization) for more details about supporting this feature.

Common issues

When implementing this feature, sometimes camera apps don't respond properly to the `onError` callbacks, both when first attempting to acquire the camera and when the camera

is no longer available. This typically results in the app crashing when this tile is enabled, but this can be used as a signal to indicate that the feature is behaving as expected.

This behavior indicates that the app isn't properly handling the `onError` callback in `CameraDevice.StateCallback` (https://developer.android.com/reference/android/hardware/camera2/CameraDevice.StateCallback). When **Sensors off** is enabled, the `onError` callback is invoked with

When **Sensors off** is enabled, the **onError** callback is invoked with `CameraDevice.StateCallback.ERROR_CAMERA_DISABLED` (https://developer.android.com/reference/android/hardware/camera2/CameraDevice.StateCallback.html#ERROR_CAMERA_DISABLED) set as the error value. Update any first-party apps to handle the **onError** callback with this value by not making any subsequent calls against `CameraDevice` (https://developer.android.com/reference/android/hardware/camera2/CameraDevice.html) until a subsequent **openCamera** call is successful.

Sensor behavior

When **Sensors off** is enabled, the sensors stop reporting any data to the system or apps. An app can still request a sensor and register a listener when **Sensors off** is enabled, but either silence is returned for the mic or the **onSensorChanged** callback is never invoked for the sensors. As soon as the tile is disabled, those same listeners start to receive the actual output from the mic or the expected callbacks to **onSensorChanged** without needing to do any additional work. The default behavior of silenced sensors is as follows.

Camera

If an app is using the camera when **Sensors off** is enabled, an error is sent to the **onError** callback method and **CameraDevice** is closed.

If an app attempts to access the camera when **Sensors off** is enabled, an error is sent to the **onError** callback method.

Microphone

When **Sensors off** is enabled, access to the microphone is still possible but only silence is returned. If an app is using the microphone when **Sensors off** is enabled, no error is generated, but the recording is silenced and just returns an array of zeros. If **Sensors off** is disabled while the app is still using the microphone, the expected audio data is returned.

If an app attempts to access the microphone when **Sensors off** is enabled, the microphone returns silence.

Sensor

When an app tries to access other sensors when **Sensors off** is enabled, the sensor type

influences the default behavior:

- **Continuous sensors:** Sensors in this reporting mode stop dispatching events. If an app is interacting with a continuous sensor when **Sensors off** is enabled, the sensor sends no additional data to the app until the feature is disabled.
- **Flush events:** A sensor flush can be requested when the tile is enabled and the `onFlushComplete` callback is invoked to indicate that the requested flush completed successfully, but no new events with sensor data are generated and returned to the `onSensorChanged` callback.
- **On-change events:** When **Sensors off** is enabled, no new change events are reported.
- **Trigger events:** When **Sensors off** is enabled, trigger events stop generating. Any existing events complete.

Customization

If your devices use the default implementation of `SensorService`, `CameraService`, and `AudioPolicyService`, then no additional customization is needed to the reference design. However, you can support sensors managed outside of `SensorManager`, remove **Sensors off** from your devices, or change the System UI for the developer quick settings tiles or the icon for the **Sensors off** tile.

Supporting more sensors

If your devices contain sensors managed outside of `SensorManager`, you should add support for them using `SensorPrivacyService` and `SensorPrivacyManager`.

When the **Sensors off** tile is toggled, `SensorPrivacyService` invokes a one-way callback for all registered listeners. When this callback is received, the registered listener can take the necessary steps based on the state of the tile. If it's enabled, all existing connections can be terminated and return empty data, and a flag set to prevent new connections. If it's disabled, the flag can be reset to allow new connections. Using the camera service

(`platform/frameworks/av/services/camera/libcameraservice/`) as an example, follow these steps to add support for a new sensor.

1. Implement the `BnSensorPrivacyListener` interface. For more details, see `SensorPrivacyPolicy` in `CameraService.h` (https://android.googlesource.com/platform/frameworks/av/+/refs/heads/master/services/camera/libcameraservice/CameraService.h#576)

.

2. Register with the `SensorPrivacyManager` and obtain the state of the tile on startup. For more details, see `SensorPrivacyPolicy::registerSelf` in `CameraService.cpp` (https://android.googlesource.com/platform/frameworks/av/+/refs/heads/master/services/camera/libcameraservice/CameraService.cpp#161)
.
3. Handle **Sensors off** state changes in the callback. For more details, see `SensorPrivacyPolicy::onSensorPrivacyChanged` and `CameraService::blockAllClients` in `CameraService.cpp` (https://android.googlesource.com/platform/frameworks/av/+/refs/heads/master/services/camera/libcameraservice/CameraService.cpp)
.
4. Prevent access to the sensor data when the tile is enabled. For more details, see the sensor privacy policy check in `CameraService::validateClientPermissionsLocked` in `CameraService.cpp` (https://android.googlesource.com/platform/frameworks/av/+/refs/heads/master/services/camera/libcameraservice/CameraService.cpp)
.

Removing Sensors off

As a developer tool for testing, **Sensors off** is hidden because a user must first enable developer mode, then choose to make the tile available in settings.

If you don't want to support **Sensors off** on your devices, remove the service tag from `packages/apps/Settings/AndroidManifest.xml`. If you remove the service tag, the **Sensors off** tile won't be available to enable from the developer quick settings tiles page.

Changing the Sensors off UI

There are two elements that can be customized for the **Sensors off** UI: the icon displayed for the developer quick settings tile and the icon displayed in the status bar when the tile is enabled. To customize the look of these icons, replace these files:

- Quick settings tile icon: `packages/apps/Settings/res/drawable/tile_icon_sensors_off.xml`
- Status bar icon: `frameworks/base/packages/SystemUI/res/drawable/stat_sys_sensors_off.xml`

Validation

As an optional developer tool, there are no CTS tests for this feature.

You can test manually by installing an app from Google Play that reads and displays all of the device's sensors. When you enable the **Sensors off** tile, ensure that none of the values for the sensors change, the microphone audio is silent, and the camera isn't accessible.