

HAL Types

As part of the Android O rearchitecture of the lower layers of the Android OS to support better modularity, devices running Android O must support binderized or passthrough HALs:

- **Binderized HALs.** HALs expressed in HAL interface definition language (HIDL). These HALs replace both conventional and legacy HALs used in earlier versions of Android and can serve the HAL in binderized mode. All devices launching with Android O or later must support binderized HALs only.
- **Passthrough HALs.** A HIDL-wrapped conventional or legacy HAL. These HALs wrap existing HALs and can serve the HAL in binderized and same-process (passthrough) modes. Devices upgrading to Android O can use passthrough HALs.

HAL mode requirements

Device	Passthrough	Binderized
Launch with Android O	HALs listed in Passthrough HALs (#passthrough) must be passthrough.	All other HALs are binderized (including HALs that are vendor extensions).
Upgrade to Android O	HALs listed in Passthrough HALs (#passthrough) must be passthrough. All other HALs provided by the vendor image can be in passthrough OR binderized mode.	HALs listed in Binderized HALs (#binderized) must be binderized.

Binderized HALs

Android requires the following HALs to to be binderized on all Android devices regardless of whether they are launch devices or upgrade devices:

- `android.hardware.biometrics.fingerprint@2.1`. Replaces `fingerprintd` which is no longer in Android O.
- `android.hardware.configstore@1.0`. New in Android O.
- `android.hardware.dumpstate@1.0`. The original interface provided by this HAL could not be shimmed and was changed. Because of this, `dumpstate_board` must be re-implemented on a given device (this is an optional HAL).
- `android.hardware.graphics allocator@2.0`. Required to be binderized in Android O so file descriptors don't have to be shared between trusted and untrusted processes.
- `android.hardware.radio@1.0`. Replaces the interface provided by `ril`d which lives in its own process.
- `android.hardware.usb@1.0`. New in Android O.
- `android.hardware.wifi@1.0`. New in Android O, replaces the legacy Wi-Fi HAL library that was loaded into `system_server`.
- `android.hardware.wifi.suplicant@1.0`. A HIDL interface over the existing `wpa_suplicant` process.

NOTE: Android provides the following HIDL interfaces which will always be in binderized mode: `android.frameworks.*`, `android.system.*`, and `android.hidl.*` (except for `android.hidl.memory@1.0` as described below).

Passthrough HALs

Android requires the following HALs to be in passthrough mode on all Android devices regardless of whether they are launch devices or upgrade devices:

- `android.hardware.graphics.mapper@1.0`. Maps memory into the process it lives in.
- `android.hardware.renderscript@1.0`. Passes items in the same process (equivalent to `openGL`).

All HALs not listed above must be binderized for launch devices.

Same-Process HALs

Same-Process HALs (SP-HALs) always open in the same process in which they are used. They include all HALs not expressed in HIDL as well as some that are **not** binderized. Membership in the SP-HAL set is controlled only by Google, with no exceptions.

SP-HALs include the following:

- `OpenGL`
- `Vulkan`
- `android.hidl.memory@1.0` (provided by the Android system, always passthrough)
- `android.hardware.graphics.mapper@1.0`.
- `android.hardware.renderscript@1.0`

Conventional & legacy HALs

Conventional HALs (deprecated in Android O) are interfaces that conform to a specific named and versioned application binary interface (ABI). The bulk of Android system interfaces ([camera](https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/camera3.h) (<https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/camera3.h>), [audio](https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/audio.h) (<https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/audio.h>), [sensors](https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/sensors.h) (<https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware/sensors.h>), etc.) are in the form of conventional HALs, which are defined under [hardware/libhardware/include/hardware](https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware) (<https://android.googlesource.com/platform/hardware/libhardware/+/master/include/hardware>).

Legacy HALs (also deprecated in Android O) are interfaces that predate conventional HALs. A few important subsystems (Wi-Fi, Radio Interface Layer, and Bluetooth) are legacy HALs. While there's no uniform or standardized way to describe a legacy HAL, anything predating Android O that is not a conventional HAL is a legacy HAL. Parts of some legacy HALs are contained in [libhardware_legacy](https://android.googlesource.com/platform/hardware/libhardware_legacy/+/master) (https://android.googlesource.com/platform/hardware/libhardware_legacy/+/master), while other parts are interspersed throughout the codebase.

Except as otherwise noted, the content of this page is licensed under the [Creative Commons Attribution 3.0 License](http://creativecommons.org/licenses/by/3.0/) (<http://creativecommons.org/licenses/by/3.0/>), and code samples are licensed under the [Apache 2.0 License](http://www.apache.org/licenses/LICENSE-2.0) (<http://www.apache.org/licenses/LICENSE-2.0>). For details, see our [Site Policies](https://developers.google.com/terms/site-policies) (<https://developers.google.com/terms/site-policies>). Java is a registered trademark of Oracle and/or its affiliates.

Last updated August 21, 2017.