HADK - Hot

Unofficial hadk-hot
Spend few minutes reading it all even briefly or if you are too lazy then waste much more time later fighting with known issues
These might be contenders to merge to: https://github.com/mer-hybris/hadk-faq
Improve and make PR to hadk-faq if you think that particular instruction is and will be useful for longer time.
The intention is to begin making this a wiki.
---------- ---------- -----

2026.03.27

*mal PSA: anyone with devices using android base 8 or newer make sure you update the libhybris used in hybris-hal build because latest qt5-qpa-hwcomposer-plugin requires updating that

* mal PSA: anyone building 5.0 adaptations in testing projects on OBS please add sailfishos_5.0 target to your adaptation project, I switched the repos to use major.minor versioning to remove the need to have separate target for each small release.

An example https://build.sailfishos.org/project/show/nemo:testing:hw:fairphone:fp5. So next time someone updates their devices from testing project the adaptation and common repo will change to use that 5.0 target

* "/bin/sh: ccache: command not found" when building pulseaudio-modules-droid from middleware.
Do:

```
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -R -m sdk-install zypper in ccache
```

* File /etc/bluez5/bluetooth/main.conf from droid-config-*-bluez5 conflicts with file from package bluez5-configs-mer.
Do:

```
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -m sdk-install -R zypper rm bluez5-configs-mer
```

* For some older sailfishos versions < 5.0 (or maybe < 4.6) 'zypper ref' doesn't really check whether url is correct or even reachable. It doesn't even mention failed refresh metadata for unavailable repositories. It seems to be related to ssu repository plugin.

2026.09.05: Android base specific fixes

Common

* since hybris-18:
- Do not use fetch-submodules when repo cloning repostories

* Since hybris-17:

- Add appropriate android_version_major: https://github.com/mer-hybris/hadk-faq/#hybris-17-1
- 'library "libandroidicu.so" not found': If you device do not use /odm then create sparse/odm/lib64/ (or sparse/odm/lib for 32bit minimediaservice) and put libandroidicu.so symlink pointing to /apex/com.android.runtime/lib64/libandroidicu.so or some similar path with that library. Usually adding only one simlink is enough, sometimes you may need also libicui18n.so and libicuuc.so. If your device use /odm and it contains valid data only in /odm/etc/ directory then mount odm partition to /.odm_root, create sparse/odm/lib* directory with needed symlinks in droid-config and bind mount /.odm_root/etc/ to /odm/etc/.
* Since hybris-16:

Before building hybris-hal run the following commands:

```
cd $ANDROID_ROOT/external
git clone --recurse-submodules https://github.com/mer-hybris/libhybris.git
cd $ANDROID_ROOT
hybris-patches/apply-patches.sh --mb
```

* Since android 8: some partitions can be mounted very early based on entries from devicetree instead android's fstab. Make sure you have all these partitions mounted in sailfish. For example samsung exynos7420.dtsi devicetree have defined efs partition which may be required to get wifi working correctly:

https://github.com/Exynos7420/android_kernel_samsung_exynos7420/blob/accf474903c0753fc1fcc4d25444481b701e7d94/arch/arm64/boot/dts/exynos7420.dtsi#L100

It can be checked also on device in:
    /proc/device-tree/firmware/android/fstab/

* For android 8-10 devices which use system-as-root without separate recovery you need to disable skip_initramfs handling so kernel can use hybris-boot initramfs. In kernel init/initramfs.c history find 2 following commits and revert them: "call free_initrd() when skipping init" and "Add skip_initramfs command line option". Alternatively, you can force in source code do_skip_initramfs = 0.

* If you do not package '/system' partition like in Sailfish X AOSP build instructions then you must have actual files instead symlinks in sparse/etc/selinux. Possible error message: Servicemanager: add_service Permission denied.

hybris-20 WIP

* Make sure you use latest hybris-patches

* "logd: libprocessgroup: Loading /etc/task_profiles.json for [2209] failed", then copy task_profiles.json from /system/etc to /etc and also add symlink /etc/task_profiles which points to /system/etc/task_profiles

* build *libui_compat_layer in HABUILD*

hybris-18

* dmesg errors:

```
kernel: loop_set_status: loop0 () has still dirty pages (nrpages=
kernel: print_req_error: I/O error, dev loop1, sector 0
kernel: Buffer I/O error on dev loop1, logical block 0, async page read
apexd: Failed to LOOP_SET_BLOCK_SIZE: Try again
```

Leads to apexd-bootstrap failure, ueventd start and reboot to fastboot mode. Apply this:

https://github.com/mer-hybris/android_kernel_sony_msm/pull/95
* 'Buffer I/O error on dev loop2, logical block 0, async page read'
Try adding loop.max_part=7 to kernel cmdline to makes loop images partitionable

* Starting ueventd can be caused by failure of some important services with reboot_on_failure option like apexd. This should be fixed in hybris-patches to ease early debugging.

* For devices which use dynamic partitions without real block devices for system, vendor or product either follow aosp hadk way and package all dynamic partitions or use

https://github.com/sailfish-on-nabu/parse-android-dynparts to map them and then mount by some script or custom systemd mount units. List of dynamic partitions can be found in BoardConfig*.mk and android's fstab.
Example solution for dynamic partitions handled via systemd:

- script which makes dynamic partitions available for mounting: https://github.com/mlehtima/droid-config-fp4/blob/devel/sparse/usr/lib/systemd/system/dmsetup.service

- example mount unit: https://github.com/mlehtima/droid-config-fp4/blob/devel/sparse/usr/lib/systemd/system/vendor.mount
- Do not forget to enable dmsetup and these mount units
- To prevent droid-hal creating incorrect systemd mount units add all dynamic partitions to makefstab_skip_entires similar to: https://github.com/mer-hybris/droid-hal-sony-lena/blob/master/rpm/droid-hal-common.inc#L20
- Be aware that some devices may have for example different vendor from stock/lineage super partition which can results in something doesn't work like wlan driver on spacewar

https://github.com/SailfishOSSpacewar/droid-system-Spacewar
- If your kernel have enabled selinux in permissive instead enforcing mode, files (not symlinks) in /etc/selinux are correct and logcat shows:

```
Unknown class service_manager
Could not self register servicemanager
```

then you may need selinux stubs for hw,vnd and servicemanager: https://github.com/mer-hybris/droid-config-sony-nile/blob/master/sparse/usr/libexec/droid-hybris/system/etc/init/hwservicemanager.rc#L4
99-qts
- If apexd-bootstrap fails with non zero code beacause of not updatable/flattened apexes:

```
This device does not support updatable APEX. Exiting
```

try experimental fix. This issue often leads to boringssl test failure, ueventd start and reboot to fastboot mode

```
Mixing copies of libbinder
```

You may need additional patches. Explanation: https://github.com/sailfish-on-karatep/karatep-port/blob/main/docs/rca/vndservicemanager-libbinder.md

---

2026.10.31: Recommended revisions

Using latest (default) instead following may cause issues
5.2.0
- droid-hal-configs = 6a9f35e
- pulseaudio-modules-droid = 14.2.110
- pulseaudio-modules-droid-jb2q = 14.2.104
- droidmedia = 0.20260522.0
5.1.0.x
- droid-hal-configs = 41cd1d9
- droid-hal-device = be67bbc
- pulseaudio-modules-droid = 14.2.106
- droidmedia = 81db5a7

You can use git reset --hard <tag/commit> to change revision and then build_packages.sh -b <path to source> to build except for droid-hal/configs/version
For some software which where build previously with different version you may get strange errors during compilation. In that case git clean -xdf sources might help

---

2026.09.29: Release specific issues

* 5.1.0.x
- You may need to update platform sdk before target/tooling

- Clean pulseaudio-modules-droid source directory before building new version
- If you loose internet access, 'dig jolla.com' timeouts try adding systemd-resolved to inet group: https://github.com/mer-hybris/droid-config-sony-nile/pull/151 or disable CONFIG_ANDROID_PARANOID_NETWORK in kernel
* 4.5.0.18
- End of bluez4 compatibility: lipstick-jolla-home-qt5 requires libKF5BluezQt.so.6. kf5bluezqt was updated and bluez4 patches were droped
* 4.4.0.64:
- 'tracker-extract-3 Failed to load seccomp rules' on some <= 3.4 kernels. Either add check for ECANCELED in https://github.com/sailfishos/tracker-miners/blob/master/rpm/0003-Prevent-tracker-extract-failing-when-seccomp-loading.patch#L26 or add support for Seccomp BPF to kernel: https://bugzilla.mozilla.org/show_bug.cgi?id=790923. generic_seccomp-bpf_3.0.x patch is outdated and cause random kernel panic.
* 4.3.0.12:
- remove /var/cache/ssu and run 'ssu ur' before OTA upgrade to be sure that adaptation-common repo is enabled during OTA and packages will be upgrade otherwise you will have issues like: hybris.c:53: __resolve_sym: Assertion `ptr != NULL' failed: because too old packages: https://forum.sailfishos.org/t/4-3-0-12-issue-with-droidmedia/8963
* 4.2.0.x:
- mapplauncherd[1497]: Connection: getMountNamespace: stat failed for pid 1497: No such file or directory
Mount Namespace /proc/<pid>/ns/mnt was added in kernel 3.8 and probably backported to some 3.4 kernels. If your kernel do not have it then git am 12 patches (since "vfs: Add setns support for the mount namespace") for 3.4 kernel from: https://github.com/aosp-mirror/kernel_common/commits/556ef7fe2377ad919f20f108c2b5b205a57bbfae. Not needed for mapplauncherd >= 4.2.2
- Everyone using old hal versions of sensorfw and geoclue hybris plugins (used on android base versions <= 7, but some might use even on newer android bases): before building new release images you need to update patterns to manually fix the package names (to geoclue-provider-hybris-hal and hybris-libsensorfw-qt5-hal) if you need hal versions
An example of the fix https://github.com/mlehtima/droid-config-fp2-sibon/commit/1f6720efc02219ab7eabc185c8d07b44ebcfd421
* 4.0.1.x:
- psmisc, bash and procps have been replaced by busybox. busybox pidof cause /etc/profile.d/developer-profile.sh failure on login. busybox bash/ash is limited and also cause some strange issues. busybox killall is not compatible with bluetooth-rfkill-event <= 1.0.
To install non busybox version of bash and psmisc tools add to patterns:
Requires: gnu-bash
Requires: psmisc-tools
- Migrate from patterns to meta packages: https://github.com/mer-hybris/hadk-faq#migrate-patterns-to-meta-packages
After transition make sure you have either patterns-sailfish-device-tools or zypper and busybox-static in patterns-sailfish-device-configuration-$DEVICE.inc:

https://github.com/mer-hybris/droid-hal-configs/blob/692f6aa48c3226631ab83263e13ed4ed9336e19e/patterns/templates/patterns-sailfish-device-configuration-%40DEVICE%40.inc#L10-L25
- Move your systemd services from /lib/systemd/system/ to /usr/lib/systemd/system/
- adaptation0 repo can't be refreshed on fresh platform sdk < 4.1.0.x:
    sudo ssu domain sailfish

2026.09.16: General info and various issues

* Read https://github.com/mer-hybris/hadk-faq#android-base-specific-fixes before you start building

* sailfishos-porters channel logs search: https://piggz.co.uk/sailfishos-porters-archive/index.php

* Cheat Sheet: https://docs.sailfishos.org/Reference/Sailfish_OS_Cheat_Sheet

* Latest sailfishos version can be found in forum. Make sure you build sailfish image for latest minor release not the one from thread title.
* Latest sdk target url usually points to not latest sailfish release so always use versioned target url when preparing sdk. Most likely target/tooling minor release is not important. Example for 4.5.0.18:
    sdk-manage target install $VENDOR-$DEVICE-$PORT_ARCH
https://releases.sailfishos.org/sdk/targets/Sailfish_OS-4.5.0.18-Sailfish_SDK_Target-$PORT_ARCH.tar.7z --tooling SailfishOS-4.5.0 --tooling-url
https://releases.sailfishos.org/sdk/targets/Sailfish_OS-4.5.0.18-Sailfish_SDK_Tooling-i486.tar.7z --no-snapshot
* HADK:
- use 'repo init -u https://github.com/mer-hybris/android.git -b hybris-X' instead 'repo init -u git://github.com/mer-hybris/android.git -b hybris-X'
- gstdroidcamsrc-0.conf created by mk-cam-conf is no longer needed
- local manifest: $DEVICE.xml
First you need to find your device specific repository for example in github.com/lineageos (or in some custom github if your device is not supported by lineageos) like:
https://github.com/LineageOS/android_device_sony_pdx214 and add appropriate entry to local manifest. Then cm/lineage.dependencies file will tell you what other device specific repositories you may need like android_device_sony_sm8350-common according to:
https://github.com/LineageOS/android_device_sony_pdx214/blob/lineage-20/lineage.dependencies and add it also. In common repository you may find another cm/lineage.dependencies file with kernel or some hardware repository: https://github.com/LineageOS/android_device_sony_sm8350-common/blob/lineage-20/lineage.dependencies. Usually you need to add all of device specific repositories to local manifest. Some android_hardware_* repos might be already in https://github.com/mer-hybris/android/tree/hybris-18.1 xmls.
- fixup-mountpoints converts your by-name path defined in android's fstab and .rc files (find them on device or in device repo) for example: /dev/block/platform/msm_sdcc.1/**by-name**/system to

block device for example: /dev/mmcblk0p23 which is needed to create working systemd mount units: /usr/lib/systemd/system/*.mount. These mount units will take care of mounting your partitions before droid-hal-init starts. So to create correct map (entry in fixup-mountpoints) for your device you need entries from your fstab (find it in $ANDROID_ROOT/device/$VENDOR/*/rootdir/ or directly on device) and ls -l */by-name/* (according to HADK) output from device for coressponding block device (e.g. /dev/sda24). Do not assume that your device use "block/bootdevice/by-name/" or "block/platform/msm_sdcc.1/by-name" path for all partitions, check fstab instead! Do not follow _a/_b syntax in fixup-mountpoints used for official devices like pdx213 unless you really have these suffixes in fstab.
Entries like: system /system ext4 ro,bar
indicates dynamic/super partition and then needs to be handled differently
Example for hammerhead:

- fstab entry: /dev/block/platform/msm_sdcc.1/by-name/userdata /data ext4

noatime,nosuid,...
-output of 'ls -l /dev/block/platform/*/by-name/' or 'mount' from device: ../../../../mmcblk0p28
Then entry in fixup-mountpoints should looks like:

-e 's block/platform/msm_sdcc.1/by-name/userdata mmcblk0p28 ' \
* mer-kernel-check: do not enable OPTIONAL filesystems unless you plan to use them.

* Make sure that after making hybris-hal, you have fstab* and ueventd*rc in

out/target/product/$DEVICE/root/: https://github.com/mer-hybris/droid-hal-

device/blob/53e6b3018148e4cc2a73e816ae979af8c5d2bfaa/droid-hal-device.inc#L498
If they are not there and you are using old android base then 'make modules' (on hybris-18 use 'allmod') can help figuring out name of modules you need and build it like: 'make fstab.qcom ueventd.rc'. If there are no such modules or files are still not copied to out/ directory then do it manually. Read $ANDROID_ROOT/droid-hal-$DEVICE.log to confirm that mount units for /system and other partitions are created. If android's fstab do not include '/system' (or '/' in case of system-as-root), '/vendor' or '/boot' entry then you might need to add them manually.

* system-as-root: https://source.android.com/docs/core/architecture/partitions/system-as-root
Generally it is a system partition with init files (part of ramdisk) and with another /system directory and if mounted incorrectly strange looking symlinks like /system/etc -> /system/etc. This should be handled automatically by droid-hal scripts provided your android's fstab in out/ contains proper entry for system partition. In that case system partition is mounted to /system_root and /system_root/system is mounted to /system. Some devices which use system-as-root uses '/system' mountpoint for system partition entry in fstab instead '/' which needs to be changed to get correct systemd mount units generated: system_root.mount and system.mount.
* For some devices ueventd*rc file do not contains all needed rules. Common missing rule is /dev/ion but can be anything else. In that case, copy ueventd*rc somewhere from /vendor partition or /system_root as ueventd.$DEVICE.rc to appropriate place in out/ directory.
* Format /data partition before flashing sailfish to remove full disk encryption (FDE) or file based encryption (FBE) which will cause permission issues later (failed: Required key not available)
* If you kernel crash very early without any logs in console-ramoops after enabling CONFIG_VT, check whether you have breaking commit "tty: move tty_port workqueue to be a kthread" in drivers/tty/tty_buffer.c recent history and revert it. There are usually 2 or 3 more related commits with "tty_port_set_policy" changes in drivers/tty/ which needs to be reverted too . Mostly for some

lineage-15/16 kernel sources but not always.
* TWRP is known to fail when unpacking sailfish rootfs. Check /tmp/recovery.log for the exact reason: tar OOM or lack of bunzip2. There are many possible solution like downgrading TWRP, telnet to hybirs-boot without flashing whole image and unpacking image manually, copying static busybox from build environment and using it to unpack image or switching from bzip2 to gzip for sailfish image.

gzip instead bunzip: https://piggz.co.uk/sailfishos-porters-archive/index.php?log=2023-04-13.txt#line293. Do not forget to update local repo metadata.
* When creating local manifest add device specific repositories from device/*/*/{cm,lineage}.dependencies files
* Updating and building droidmedia:

```
git fetch, pull and checkout to specific revision
make droidmedia (in HABUILD)
build_packages.sh --gg
```

Be careful if you use not recommended version for particular release. These steps will build only droidmedia but not droidmedia-devel which is used during gst-droid build done on jolla's servers
* To access scratchbox target as root: sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -m sdk-install -R
* To access scratchbox target for building manually (cmake, make,..): sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -m sdk-build
* HABUILD 'sudo: account validation failure, is your account locked?'
sudo pwck -q --root "$PLATFORM_SDK_ROOT"/sdks/ubuntu
* Latest ubuntu sdk do not have cpio installed which is required to create initramfs
* build_packages.sh fails with: "usage: mb2 [global-opts] <command> [command-opts]": update platform sdk to >= 4.3.0.15
* journal: 'systemd: Failed to create /user.slice/user-100000.slice/user@100000.service/init.scope control group: Permission denied
systemd: Failed to allocate manager object: Permission denied
systemd: Failed to create cgroup /user.slice/user-100000.slice/user@100000.service/*: Permission denied'

Apply "Remove /dev/cg2_bpf cgroup device" patch: https://github.com/mer-hybris/hybris-patches/commit/54008fe9fc67c13eff6d3d48f78bb268af7883be.
If that doesn't help then add systemd.legacy_systemd_cgroup_controller=yes to kernel command line
* Sailfish browser use now gmp-droid instead gst-droid. If you used "dont-use-droid-convert" quirk or have some playback issues then you probably need:

https://github.com/sailfish-on-fxtecpro1/gmp-droid/commit/81c69798fc7c785e72aea613a8b2cc5415c5b7ee
Now gecko-camera-droid-plugin?
* build_packages.sh fails: 'File /etc/ofono/ril_subscription.conf
  from install of
    droid-config-$DEVICE-1* (local-$DEVICE-hal)
  conflicts with file from package
    ofono-configs-mer-* (@System)'

or similar for 'File /etc/ofono/binder.conf' and ofono-configs-binder conflicts

Add https://github.com/mer-hybris/droid-config-sony-nile/blob/ef51664245038906283c52aac22c7958d9a17107/droid-config-common.inc#L14-L15 to droid-config-$DEVICE.spec, build packages and then:

```
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -m sdk-install -R zypper --plus-repo $ANDROID_ROOT/droid-local-repo/$DEVICE in droid-config-$DEVICE
```

* droid-hal-version-$DEVICE.log: Problem: the to be installed droid-config-$DEVICE-1-202111031520.armv7hl requires '$DEVICE-bluez-configs', but this requirement cannot be provided:
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -m sdk-install -R zypper --plus-repo $ANDROID_ROOT/droid-local-repo/$DEVICE in droid-config-$DEVICE
* logcat: 'E/qdoverlay(2142): getRotDevFd failed to open rotator device' or 'HWRotator::Open: open /dev/mdss_rotator failed err = 13 errstr = Permission denied':

user is no longer in system group: https://github.com/mer-hybris/droid-hal-device/commit/1dac6f1ff3b58215003af61ba52bd91c5b177219
Change *_rotator group to graphics in: device/$VENDOR/*/ueventd.*.rc:/dev/*_rotator 0664 system graphics
* logcat: 'QC-time-services: Daemon:genoff_init_config: RTC initilization failed'. Disable time_daemon or vendor.time_daemon in disabled_services.rc: https://github.com/mer-hybris/droid-config-sony-ganges-pie/blob/master/sparse/usr/libexec/droid-hybris/system/etc/init/disabled_services.rc. On some older hybris base time_daemon is defined in some init*rc file which is parsed before /usr/libexec/droid-hybris (see logs). In that case you must disable service in that init*rc file. Sometime there can be also vendor.time_daemon defined somewhere in /vendor.
* Touchscreen doesn't work: remove :/dev/input/eventX from droid-hal-device.conf to enable auto detection of touchscreen node. If that doesn't help use evdev_trace (from mce-tools package) or 'cat /dev/input/eventX' to figure out correct node. (`for f in /sys/class/input/event*/device/name ; do printf "%s: %s\n" $f $(cat $f); done` can give an idea which device is which)
* Black background in homescreen or black images in gallery:

Set QT_OPENGL_NO_BGRA=1 in 99-qtscenegraph.conf: https://github.com/mer-hybris/droid-config-sony-nile/blob/master/sparse/var/lib/environment/nemo/99-qtscenegraph.conf
* After upgrading target rpm fail: 'error: db4 error(-30971) from dbenv->open: DB_VERSION_MISMATCH: Database environment version mismatch
error: cannot open Packages index using db4 - (-30971)
error: cannot open Packages database in /var/lib/rpm':

```
sb2 -t $VENDOR-$DEVICE-$PORT_ARCH -m sdk-install -R rm -rf /var/lib/rpm/__db.00*
```

* If your kernel do no have NETFILTER_XT_MATCH_QTAGUID then add "!" in mer-kernel-check script to make it optional: https://github.com/mer-hybris/mer-kernel-check/blob/master/mer_verify_kernel_config#L221
* Do not upload tmp/ or documentation.list in droid-config to github. These are temporary files recreated every time droid-config is build and they make your commits more obscure

* ' max77843-fuelgauge: driver failed to report `status' property: -22' this kind of spam in dmesg is usually caused by broken drivers which reports property which they do not support. Suppresing error printing in power_supply_sysfs.c can lead to several issues so real bug must be fixed in particular driver. For example following driver pretends to support status property:

https://github.com/edp17/android_kernel_samsung_exynos5433/blob/a06e248eafb831411afbf3bc57f4a47d50a89fc3/drivers/battery/max77843_fuelgauge.c#L22

but clearly status is not handled neither in max77843_fg_get_property nor in max77843_fg_set_property. Removing POWER_SUPPLY_PROP_STATUS should fix issue for this particular driver but other drivers might require different fixes. Fixing one bug might reveal another:

'max77843-fuelgauge: driver failed to report `charge_full' property': It support setter but not getter so add 'return -ENODATA in appropriate get_property function'

'bq51221-charger: driver failed to report `current_now' property'

Be careful: these are just examples and you are dealing with Li-Ion related drivers.

* USB cable detection is handled by usb-moded. It expects online/present and type events from power_supply devices for proper functioning. For debugging issues use 'udevadm monitor -p' and -D argument in /var/lib/environment/usb-moded/usb-moded-args.conf.

Some devices doesn't send online/present udev events on usb connection/disconnection but still set this parameters: /sys/class/power_supply/*/{uevent,present,online,type}. In that case adding power_supply_changed(<usb power_supply pointer>) to driver might be enough to get everything working:

https://github.com/edp17/android_kernel_samsung_exynos5433/blob/4dc34f7772e8e48bc2ecdc75bbfc4c0e81c9b3cd/drivers/battery/sec_battery.c#L2808

Alternative way is tracking ext_con or android_usb devices: https://github.com/sailfishos/usb-moded/pull/12

* Some devices needs forcing the USB to start in peripheral mode otherwise there will be no sign of usb connection in dmesg:

https://review.lineageos.org/c/LineageOS/android_device_xiaomi_sm6150-common/+/291494/4

https://github.com/sailfishos-on-sake/hybris-boot/blob/fa1ec67d3e8cfa564f61f6c1a0f65caf7c64fbfa/init-script#L197-L199

* Plugins: which one you should use is really device and adaptation/stock android version specific
- ofono plugin:
ofono-ril-plugin <= android 7
ofono-ril-binder-plugin <= android 10
ofono-binder-plugin >= android 10: binder-list -d /dev/hwbinder | grep IRadio (for supported radio interfaces)

radioInterface can be set in /etc/ofono/binder.d/*.conf

You may you need configuration files in sparse/etc/ofono, different for each plugin. See sony droid-config repos https://github.com/mer-hybris?q=droid-config- to figure out correct configs for plugin.
- pulseaudio plugin:
pulseaudio-modules-droid-jb2q <= android 10
pulseaudio-modules-droid >= android 11

For debugging you may add one or more -v to pulseaudio command. Take a look at /usr/lib/systemd/user/pulseaudio.service and /etc/sysconfig/pulseaudio and make sure you run it as user.
audio calls: https://github.com/mer-hybris/audiosystem-passthrough/blob/master/README.md
If you device have only 32bit audio library (ls /vendor/lib*/hw/audio*.so) e.g. /vendor/lib/hw/audio.primary.universal7420.so but without 64bit version e.g. /vendor/lib64/hw/audio.primary.universal7420.so then hidl_compat wrapper may help you:

https://github.com/Halium/android_vendor_halium_hardware/tree/halium-10.0

Clone repository and make sure you use correct branch for your android adaptation then you can build it similar to hybris-hal so: make audio.hidl_compat.default
Next figure out which audio.primary.*.so file is loaded via strace and adapt following mount unit.

https://github.com/mlehtima/droid-config-fp4/blob/devel/sparse/usr/lib/systemd/system/vendor-lib64-hw-audio.primary.default.so.mount

Enable android.hardware.audio* service if it's disabled in disabled_services.rc.
https://docs.ubports.com/gl/latest/porting/configure_test_fix/Sound.html
- sensorfw plugin:
hybris-libsensorfw-qt5-hal <= android 7: test_sensors
hybris-libsensorfw-qt5: >=android 8: binder-list -d /dev/hwbinder | grep ISensors

- gbinder api level: /etc/gbinder.conf

If binder-list -d /dev/binder doesn't show interface then try changing api level
possible values:

https://github.com/mer-hybris/libgbinder/blob/2b2e22aced55d023f2419817e92935e78675c108/src/gbinder_config.c#L189

- bluetooth:
On recent android bases you need to install bluedroid and enable HCIVHCI and some other configs:

https://github.com/mer-hybris/android_kernel_sony_msm/blob/fbfc0728b6a708ed67791ccca23821777d6c89eb/arch/arm64/configs/aosp_lena_pdx213_defconfig#L920

- Vibrator

https://hadk.sailfishos.org/subsystems/#vibration-force-feedback
https://hadk.sailfishos.org/middleware/#non-graphical-feedback-daemon-ngfd
partially outdated informations: https://github.com/mer-hybris/hadk-faq#id54
There few plugins:
- ngfd: libngfd_ffmemless.so: have_ffmemless in droid-hal-version
- ngfd-plugin-native-vibrator: have_vibrator_native in droid-hal-version
- ngfd-plugin-droid-vibrator: have_vibrator in droid-hal-version
- ngfd-plugin-binder-vibrator: have_vibrator_binder in droid-hal-version
- libngf-qt5-qtfeedback: It bridges haptic feedback support from ngfd to qt applications. Deprecates qt5-feedback-haptics-*.
- qt5-feedback-haptics-droid-vibrator and qt5-feedback-haptics-native-vibrator: depracated

- qt5-feedback-haptics-ffmemless: depracated

Add appropriate have_* variable in your droid-hal-version

* suspend
- test with: mcetool --get-suspend-stats or systemdatascope app
- You may need to enable autosleep in kernel: CONFIG_PM_AUTOSLEEP=y
- If you usb mode doesn't change from peripheral/none when you plug in/out usb. Permanent peripheral mode might block suspend. Following service should help:

https://github.com/VerdandiTeam/usb-state-notifier

Make sure it uses correct '/sys/devices/platform/soc/4e00000.ssusb/mode' path
TO CONFIRM:
- Make sure you have this ne patch applied: "Add kernel wakelock to Android suspend":

https://github.com/mer-hybris/hybris-patches/pull/33

Then build patched service: 'make android.system.suspend-service'

and add rc service to start patched version of android suspend: https://github.com/sailfishos-sony-nagara/droid-config-sony-nagara/blob/main/sparse/usr/libexec/droid-hybris/system/etc/init/android.system.suspend-service.rc

* 'android.hardware.keymaster@3.0-impl: Configure failed'
Set correct OS version and patch level for your *boot.img

https://github.com/sailfishos-sony-tama/droid-hal-img-boot-sony-tama-pie/commit/4712b68c2f7387b5c342d4a3fe492d78a568bf50

https://wiki.postmarketos.org/wiki/Category:Bootloaders/Keymaster

* Random kernel <= 3.10 crash at:
'fs/sysfs/dir.c' called by sysfs_remove_dir: BUG_ON(sd->s_flags & SYSFS_FLAG_REMOVED);

Apply "kobject: grab an extra reference on kobject->sd to allow duplicate deletes" patch

* Some 3.4 and 3.10 kernels requires following patches to get systemd and sailjail/firejail working

- Revert "VFS: make vfs_fstat() use f[get|put]_light()

- vfs: make O_PATH file descriptors usable for fstat()

- vfs: allow O_PATH file descriptors for fstatfs() or or this one

- proc: show mnt_id in /proc/pid/fdinfo

or bunch of others

- https://github.com/shr-distribution/linux/commits/hammerhead/3.4/halium-9.0

- https://github.com/shr-distribution/linux/commits/mako/3.4/halium-9.0

- net loopback: Set loopback_dev to NULL when freed

Also make sure you have enabled CONFIG_.*_NS in kernel.

* ohmd failing twice or more with error: 'ohmd.service: Failed at step NAMESPACE spawning /usr/sbin/ohmd: No such file or director' is caused by mising namespace options from mer-kernel-check and CONFIG_USER_NS

* wpa3 support (allowed options: none, mixed, full): https://github.com/mlehtima/droid-config-fp5/blob/00b78f4d2df9e09980fe6cda0b6a1520c7406ff2/sparse/etc/connman/main.conf.d/10-fp5.conf#L5

* Updated promote.py for python3: https://pastebin.com/KC4LKydz. Make sure you use right "apiurl = https://build.sailfishos.org" in ~/.oscrc

* bluez4 was dropped: https://github.com/mer-hybris/droid-hal-configs/pull/245

* Alternative to lipstick-hack (aka surfaceflinger-hack): https://wiki.merproject.org/wiki/Adaptations/faq-hadk#Alternative_to_lipstick-hack_.28aka_surfaceflinger-hack.29
* stracing android services
Because android services might be started by specific user/group then regular stracing may hide or show false culprit. Sometimes it may be good idea to strace service directly in *.rc file which it is defined:

```
-service tad /vendor/bin/tad /dev/block/bootdevice/by-name/TA 0,16
+service tad /usr/bin/strace -ff -o /data/strace/tad.strace /vendor/bin/tad /dev/block/bootdevice/by-name/TA 0,16
```

also create writable directory for generated files:

```
mkdir -m777 /data/strace/
```

2026.07.10: Generating logs:

* Always provide full logs with information of what you did so far, full command you are running with arguments and shell prompt in paste so it's clear what you execute and where: host, device, sdk, target sdk or ubusdk. If you are debugging issues on device: reboot (so it is not filled with too much noise) and make sure logs contains early boot messages. It's better to highlight relevant part than providing partial log. Do not paste long logs directly to channel, use paste services instead: https://pastebin.com - full of ads, https://paste.opensuse.org - raw view links expire quickly or similar. ubuntu one without raw view and lack of horizontal scrollbar is worst choice. Better alternatives are welcome
* Increase your terminal buffer to not loose early logs
* Always upload your latest changes in droid-config/hal/kernel so it's easy to inspect them. Crazy ideas: read logs and search for errors from the top before you upload them
* In TWRP sailfish root path / is under /data/.stowaways/sailfishos/
* Connecting device to pc shortly after powering it on and running dmesg in your host system can reveal whether you are booting hybris initframfs, android init or there is no sign of life on usb from your device
* If ifconfig shows that inet address is not assigned and you can't telnet to device: https://docs.halium.org/en/latest/porting/debug-build/early-init.html. Sometimes init*usb*.rc file can mess up your usb connection after droid-hal-init is started. To avoid it you must prevent that particular *.rc file from being included and parsed. This is probably done in some more general

init*.rc file. Sometimes masking usb-moded can help also untill you get gui working.
* If your device bootloops but you can force boot straight to TWRP then cat /proc/last_kmsg or /sys/fs/pstore/console-ramoops might contains logs from faulty boot but only if kernel crashed. Make sure this log are really from sailfish boot attempt instead TWRP or android.
* If you are on 23 telnet port then device stuck in initramfs, cat init.log and diagnosis.log will tell you why switching to real rootfs failed
* If your device bootloop fast enough or reboot to some different mode so you can't access telnet check in TWRP whether /data/.stowaways/sailfishos/init.log exists if no then your device did not pass or even reach initramfs stage. In that case make sure your hybris-boot.img contains valid initramfs: 'abootimg -i hybris-boot.img'. If init.log is generated you can stop booting device in /init-debug script by creating /data/.stowaways/sailfishos/init_enter_debug2 which might allows you to telnet.
* If your device reboots after 60s without any sign of life in dmesg then check whether you have enabled CONFIG_USB_CONFIGFS_*RNDIS
* dmesg: usually useful to confirm whether selinux is disabled at boot time: 'SELinux: Disabled at boot' for hybris <=16 or initialized: 'SELinux: Initializing' for hybris >= 17. Also if something fails very early before systemd is started or some droid-hal-init logs are partialy in dmesg and partially in journal. Kernel buffer can be small so you need to be fast to grab early boot messages.
Arguments:
    -w (live log, not supported on very old kernels)
* journalctl: enable persistent logging if you have trouble with getting access to device, you kernel is noisy or you are too slow so early boot logs are lost: https://github.com/mer-hybris/hadk-faq#persistent-journalctl. Journal binary log can be also retrieved from /var/log/journal via TWRP.
Arguments:
    --no-pager (shows everything without paging)
    --no-tail -f (shows everything and continuously prints new messages)
    -b0 (only from current boot, useful if you have persisten logging enabled) -b-1 (from previous boot)
* /usr/libexec/droid-hybris/system/bin/logcat: for android part debugging. Can produce different logs as regular user than root user.
* strace <command>
Arguments:
   -f (trace child processes - if you are not sure just always add it)
   -e file (for tracing files)
   -s256 (increase max string size)
* If droid-hal-init do not print: 'Parsing file *.rc' or you have: 'output lines suppressed due to ratelimiting' in dmesg then add 'printk.devkmsg=on' to kernel cmdline to enable unlimited logging from user space applications
* Always add audit=0 to kernel bootparams to suppress audit spam and clean up logs unless you plan to use audit/selinux.
* binder-list from libgbinder-tools package:

```
binder-list
binder-list -d /dev/hwbinder
binder-list -d /dev/vndbinder
```

---------- ----------

No sign of life

If you flashing hybris-boot.img leads to no sign of life from device neither in host dmesg nor in device then make sure boot image is correctly created. Best way to do this is by comparing with lineage's boot.img via unpack_bootimg. If all looks sane then building hybris-boot without any defconfig changes is good idea. As a last resort recreate manually lineage boot.img if that shows same sign of life then try to replace lineage's ramdisk with hybris one. Make sure tarball uses same format.

TODO: vendor_boot boot ing issues

---------- ----------

I have telnet but not gui:

* Check in .config, zcat /proc/config.gz or dmesg whether selinux is set up correctly. When you build kernel, defconfig is used as source from which out/target/product/$DEVICE/obj/KERNEL_OBJ/.config is generated. If you forgot about dependencies in defconfig then it will not match generated .config. Sometimes defconfig is stripped from some configs so it's good to regenerate it: build kernel without any hybris changes in defconfig, copy generated .config back to defconfig and commit that change. Disabling CONFIG_AUDIT which is part of CONFIG_SECURITY_SELINUX dependencies is common mistake.

* Search for "droid mount" in journal and make sure /system and other partitions are mounted. You should have similar partitions mounted like on android. android's fstab can give you a clue but some partitions are mounted differently

* Search for firmware loading failure. If particular driver needs firmware before partitions are mounted and droid-load-firmware.sh is not triggered then building driver as module and creating systemd service (like for wlan) which will modprobe it might help.

* droid-hal-init (modified android init) must not fail and usually can't be started twice unless it fails on early stage or you clean up some files it created before starting it again. If you want to strace it then mask service and reboot: 'systemctl mask droid-hal-init'. logcat can be used only after droid-hal-init started.

* compositor used by sailfish is called lipstick. For debugging gui issues use 'systemctl mask user@100000' and reboot device. This will prevent lipstick starting/failing and ease test_hwcomposer/minimer use.

To test android part, mask user@100000, reboot an start surfaceflinger:

```
/system/bin/surfaceflinger
# On android >= 11 you may also need:
ANDROID_ROOT="/system" /system/bin/bootanimation
```

2025.06.15: OBS (build.sailfishos.org) tips and tricks

Extends: https://github.com/mer-hybris/hadk-faq#obs-build-and-over-the-air-updates-ota

Useful OBS Notes: https://github.com/sailfishos-chum/main/blob/main/GettingStarted.md
* For account conntact Keto at #sailfishos* IRC channel or on sailfish forum with username and email

* OBS API url for osc: https://build.sailfishos.org
* Follow fp2-sibon example repos and adapt for your device:

https://build.sailfishos.org/project/show/nemo:devel:hw:fairphone:fp2-sibon and

https://build.sailfishos.org/project/show/nemo:testing:hw:fairphone:fp2-sibon
* You can add new relases using "Meta" tab instead via "Add repositories"
* '<path project="sailfishos:*" repository="latest_armv7hl"/>' should be latest in projects list in "Meta"
* devel repo is for you and your testers development. It should have only one release sailfish_latest_armv7hl. Testing repository might have releases for every no EA (Early Access) sailfish version like sailfishos_4.2.0.21.
* Copy and adapt "Project Config" from fp2-sibon to your device:

https://build.sailfishos.org/project/prjconf/nemo:testing:hw:fairphone:fp2-sibon
* OBS targets are not added at the same time as new sailfish os version is announced. Search for "sailfishos:4." in https://build.sailfishos.org/project to be sure that you can build for particular release. "sailfishos:latest" for devel repository is updated only after some time when full release is out. Confirm here: https://build.sailfishos.org/repositories/sailfishos:latest
* Always disable old releases before you create new in testing repo. You can do this in "Repositories" tab by setting "Build Flag" to disable or manually in "Meta" tab (see fp2-sibon example). If you do not do this then you will override your old release. Alternatively you can create subproject for each release which will ease maintenance
* Add droid-config/hal/version and other adaptation packages but do not add packages like qt5-feedback-haptics-native-vibrator or community-adaptation-* which are already in repositories enabled on your device: https://build.sailfishos.org/project/show/nemo:testing:hw:common,

https://releases.jolla.com/releases/$RELEASE/jolla-hw/adaptation-common/$PORT_ARCH/ (since 4.3.0) or in https://releases.jolla.com/releases/$RELEASE/jolla/$PORT_ARCH/ unless you need to build some different version.
* Only droid-hal and droidmedia needs to by uploaded because they are build from android sources. For all other packages, create _service file and fill it with:

https://build.sailfishos.org/package/view_file/nemo:testing:hw:fairphone:fp2-sibon/gst-

droid/_service?expand=1. You can set revision to HEAD but sometimes latest version for particular package which is under development can not be used then you can set revision to tag or git commit hash. OBS will fetch sources from git and build automatically. If you want to rebuild package then use "Trigger services" button.

* osc can be also used for uploading new packages: https://wiki.merproject.org/wiki/Osc_Setup
* Package name determine which spec file will be used to build it. Here sensorfw-qt5-hybris.spec will be used https://build.sailfishos.org/package/show/nemo:testing:hw:fairphone:fp2-sibon/sensorfw-qt5-hybris but if you use binder backend then you need sensorfw-qt5-binder.spec thus: https://build.sailfishos.org/package/show/nemo:testing:hw:oneplus:cheeseburger/sensorfw-qt5-binder
* Package blocked for very long time with following message: "downloading 1 dod packages". Ping OBS maintainer
* Your $ANDROID_ROOT/*.ks file which is used by "mic" during image creation by default use local repository on your disk for adaptation packages and common/jolla repos for other packages. Also build_packages -i/--mic overrides your *.ks file. One way to use your new OBS repo is: download droid-config-$DEVICE-ssu-kickstarts*.rpm: https://build.sailfishos.org/package/binaries/nemo:testing:hw:fairphone:fp2-sibon/droid-config-fp2-sibon?repository=sailfishos_4.2.0.21 and unpack *.ks with your new adaptation-community-$DEVICE repo, update droid-local-repo metadata then run mic manually:

```
EXTRA_NAME="extra"
sudo mic create fs --arch=$PORT_ARCH \
 --tokenmap=ARCH:$PORT_ARCH,RELEASE:$RELEASE,EXTRA_NAME:$EXTRA_NAME \
 --record-pkgs=name,url \
 --outdir=sfe-$DEVICE-$RELEASE$EXTRA_NAME \
 --pack-to=sfe-$DEVICE-$RELEASE$EXTRA_NAME.tar.bz2 \
 $ANDROID_ROOT/Jolla-@RELEASE@-$DEVICE-@ARCH@.ks #--verbose --debug
```

* .ks file change sometimes together with new target and droid-config so download new one for every release
* mic installs only one package patterns-sailfish-device-configuration-$DEVICE defined in *.ks file when sailfish image is created. That means every other package installed in image is pulled by droid-config/patterns/patterns-sailfish-device-configuration-$DEVICE.inc. For optional packages which can be removed later without uninstalling other adaptation packages use weak dependencies like: "Recommends: jolla-developer-mode" instead "Requires: jolla-developer-mode".
* Ports do not recieve updates in Settings page. For upgrade disable 3th party repos and use "ssu release $RELEASE; version --dup". If it fails for some unknown reason try zypper ref; zypper dup --details to figure out what is broken.
* When using OBS web interface to upload rpm package with '+' in file name OBS will rename plus sign to space and copy will fail. To solve this explicitly set correct "Filename" at "Add File" page when you upload such rpms.

* The OBS API url used to osc changed, it's now https://build.sailfishos.org . If you have local checkout and you want to replace to url in those use something like this:

```
grep --include=_apiurl -rl build.merproject.org . | xargs sed -i 's/build.merproject.org/build.sailfishos.org/g'
grep --include=_apiurl -rl api.merproject.org . | xargs sed -i 's/api.merproject.org/build.sailfishos.org/g'
```

depending on which url you had been using before. In addition to changing it in local checkouts also fix the url in .oscrc
* _pattern is no longer used for new releases, forget about it

---

Revision #89
Created 21 September 2023 17:59:54 by poetaster
Updated 31 October 2026 22:55:44 by kdr