When new Android versions become available, the following steps should be taken in order to fully support the new API version in the Flutter Engine. These steps only change the Engine to build against and target the new API and does not guarantee everything works with the changes in the new android API.

Buildroot:

build/config/android/config.gni: Edit default_android_sdk_version and default_android_sdk_build_tools_version

SDK and other dependencies:

Flutter now includes a script to download, package, and upload the Android SDK to CIPD. These CIPD packages are then used as dependencies by the Flutter engine and recipes so that there is a stable archived version of the Android SDK to depend on. The script is located in the Flutter engine repo under `tools/android-sdk/create_cipd_packages.sh`.

Edit `tools/android-sdk/packages.txt` to refer to the updated versions you want. The format for each line in packages.txt is `<package_name>:<subdirectory_to_upload>`. Typically, each should be updated to the latest available version which can be found with the `sdkmanager --list --include_obsolete`. `sdkmanager` can be found in your `commandline-tools` package of the android sdk.

The script must be run on a Linux or Mac host. Run:

`` `$ ./tools/android-sdk/create_cipd_packages.sh <new_version_tag> <path_to_your_local_androi ``

This script will download and re-upload the entire SDK, so it may take a long time to complete. `cmdline-tools` should be installed in your local sdk as the script uses `sdkmanager`. Once the CIPD packages are finished uploading, you can update the sdk version tag used in `.ci.yaml`, `DEPS`, and elsewhere.

It is no longer recommended to upload CIPD android sdk packages manually, but if it must be done, run the following commands to zip and upload each package to CIPD:

`` `$ cipd create -in <your-android-dir>/Android/sdk/<some_package> -name flutter/android/sdk/< ``

Typically, `<your-android-dir>` is in your home directory under `~/Library/Android`. The `<new-version-tag>` is what you will use to specify the new package you uploaded in the Flutter engine DEPS file.

Engine:

- DEPS: Roll buildroot hash
- DEPS: Change the version parameter under `flutter/android/sdk/platforms/${{platform}}` to the new build-tools version tag. Eg, `'version': 'version:30r2'`
- DEPS: Change the version parameter under `flutter/android/sdk/platform-tools/${{platform}}` to the new build-tools version tag. Eg, `'version': 'version:30.0.4'`

- DEPS: Change the version parameter under `flutter/android/sdk/build-tools/${{platform}}` to the new build-tools version tag. Eg, `'version': 'version:30.0.1'`

Flutter LUCI Recipes:

Scenario tests recipes/engine/scenarios.py: Update the CIPD hash to the latest version that contains the configuration for the Android Virtual Device (AVD) desired `generic_android<API#>.textpb` and change the script to use the new textpb config and change the API number in the logs. The CIPD package for the AVD launcher can be found at https://chrome-infra-packages.appspot.com/p/chromium/tools/android/avd/+/ and updating the packages uploaded there is tracked in https://bugs.chromium.org/p/chromium/issues/detail?id=1112429#c7

Framework, examples and samples

- Templates in tooling: Change `targetSdkVersion` in various `build.gradle.tmpl` files to use the new API version
- Examples, samples, gallery, etc: Change `targetSdkVersion` in `android/app/build.gradle` for each project to the new API version.