# Contributing to Open GApps

:+1::tada: First off, thanks for taking the time to contribute! :tada::+1:

The following is a set of guidelines for contributing to Open GApps on GitHub.

These are just guidelines, not rules, use your best judgment and feel free to propose changes to this document in a pull request.

## What should I know before I get started?

### Behave

We could write a big chapter here about how to behave; but we expect everybody to act as good-natured grown-ups and to be respectful.

A good atmosphere and taking into account that we are all volunteers and that can have many off-line responsibilities too is key.

Demanding a feature or functionality for your 1% use-case is not appreciated. But you are free to donate source code or resources.

### The Open GApps Project vs pre-built OpenGApps.org packages

The Open GApps Project focuses on writing the scripts that enables everyone to build their own GApps package.

The OpenGApps.org pre-built packages by the buildbot is an easy-to-use service but not the main focus of the project.

### POSIX-compatible shell \*will\* hurt your brain

The majority of the Open GApps Project consists of POSIX-compatible shell code. That means there are limitations in how well code

can be structured (e.g. there are not even arrays available). This limits readability, so take into account when writing new code

that it will by default already be hard to read by other developers, so don't make it more complicated than necessary.

### Code structure

\* Only commands meant for the end-user go in the root directory

\* The bulk of the code goes into the scripts/ directory

\* All support-files are marked as inc.\* and are organized to be sourced in a logical place (i.e. ONLY the root script OR a low-level function)

\* Try to avoid mixing \*layers\* of the function-waterfall within one file, best to use a new support-file (i.e. buildgapps->buildtarget->buildhelper)

\* Binaries and databases for the scripts are allowed but should be put as much as possible in their own sub-folders, with

a clear description where they were obtained and how they can be compiled (for future updates or other platforms).

\* Try to use functions whenever it can make otherwise repetitive statements more readable

\* APKs are stored in separate repositories, named after their CPU architecture in the sources/ directory

### Code Styleguide

\* Use only POSIX-compatible code (but you can place markers where in the future you want to use bash)

\* Try to stick to latest Open Group specifications for command arguments, parameters and their structure

\* Try to use \*""\* as much as possible whenever passing an argument that should NOT be shell-globbed (i.e. 99% of the time)

\* Indentation is ONLY with spaces (no tabs) with 2 spaces per level

\* Try to \*"one-line"\* indenting statements if possible (and not bad for readability): i.e. `if [ TRUE ]; then`

\* Avoid superfluous statements like \*;\* at the end of lines (we are still converting legacy occurrences)

\* If using a \*;\* or similar statement please put a space after it for readability

\* Use a \*#\* to document lines in or near functions if it is not directly apparent how they work

### APK Uploads

\* The Open GApps project offers ready-made tools to create commits and upload APKs to the sources-repositories

\* If not using the tools, please do try to mimic their commit-style (one app per commit; rebase no merging; replace no removing/adding; git message)

\* If possible NEVER remove an APK in one commit and putting its update in another commit. (It hinders git delta-compression)