

Source Control Tools

Working with Android code requires using both **Git** (an open-source version-control system) and **Repo** (a Google-built repository-management tool that runs on top of Git). See the [Source Control Workflow](#) page for a summary of regular actions you can take, such as uploading changes for review.

[Git]

Git handles large projects that are distributed over multiple repositories. Android uses Git for local operations such as local branching, commits, diffs, and edits. One of the challenges in setting up the Android project was determining how to best support the outside community---from the hobbyist community to the large OEMs building mass-market consumer devices. Google wanted components to be replaceable, and wanted interesting components to have a life of their own, outside of Android. Google first chose a distributed revision control system, then narrowed it down to Git.

For more details on Git, refer to this [Git Documentation](#).

[Repo]

[Repo](#) unifies Git repositories when necessary, performs uploads to the [Gerrit revision control system](#), and automates parts of the Android development workflow.

The Repo Launcher provides a Python script that initializes a checkout and downloads the second part, the full Repo tool. The full Repo tool is included in an Android source code checkout. It's located, by default, in

```
$SRCDIR/.repo/repo/... and it receives forwarded commands from the downloaded Repo Launcher.
```

Repo doesn't replace Git, it only makes it easier to work with Git in the context of Android. Repo uses [manifest files](#) to aggregate Git projects into the Android superproject. You can put the `repo` command, which is an executable Python script, anywhere in your path. In working with the Android source files, you can use Repo for across-network operations such as with a single Repo working directory.

In most situations, you can use Git instead of Repo, or mix Repo and Git commands to form complex commands. However, using Repo for basic across-network operations makes your work much simpler. For more details on Repo, see the [Repo Command Reference](#), [Repo README](#), the [Preupload Hooks](#) (tests) that can be enabled in Repo, and [general docs in AOSP](#).

To download and install the **Repo Launcher** from *git-repo- downloads*, see [Installing Repo](#).

[Gerrit]

[Gerrit](#) is a web-based code review system for projects that use Git. Gerrit encourages a more centralized use of Git by allowing all authorized users to submit changes, which are automatically merged if they pass code review. In addition, Gerrit makes reviewing easy, displaying changes side by side in the browser and enabling inline comments.

Find the Android Gerrit review interface at android-review.googlesource.com and the code navigation interface at android.googlesource.com.

[Installing Repo]

Follow these steps to install Repo.

Run these commands to use the official package from your Linux distribution:

```
sudo apt-get update
sudo apt-get install repo
```

After installation, verify that `repo version` reports something similar to the following when it's running in a regular directory. (A regular directory isn't part of a repo client; for example, it's your home directory.)

Run this command:

```
repo version
```

Expect a report similar to this one:

```
<repo not installed>
repo launcher version 2.15
(from /usr/bin/repo)
```

- The `repo launcher version` number reporting as 2.15 or higher indicates a correct version number and proper installation.
- `(from /usr/bin/repo)` indicates installation from a package.
- `(from /home/<>/bin/repo)` indicates manual installation.