Introduction

This page is about how to design a project that can build independently with GN but also be brought into the Chrome build.

GN is in principle no different than GYP in that there is some core configuration that must be the same between both the standalone build and the Chrome build. However, GN is much more explicit in its naming and configuration, so the similarities between the two builds are also much more explicit and there is less flexibility in how things are configured.

What you need for a minimal GN build

Requirements:

- A master build config file. Chrome's is `//build/config/BUILDCONFIG.gn`
- A separate build file for the toolchain definition. It's not a good idea to put these in a BUILD.gn file shared with any target definitions for complex reasons. Chrome's are in `//build/toolchain/<platform>/BUILD.gn`.
- A `BUILD.gn` file in the root directory. This will be loaded after the build config file to start the build.

You may want a `.gn` file in the root directory. When you run GN it recursively looks up the directory tree until it finds this file, and it treats the containing directory as the "source root". This file also defines the location of the master build config file:

- See Chrome's `src/.gn` file.
- Unlike Chrome, you probably don't need to define a secondary root.
- see `gn help dotfile` for more.

Adding a `.gn` file in a repository that is pulled into Chrome means that then running GN in your subdirectory will configure a build for your subproject rather than for all of Chrome. This could be an advantage or a disadvantage.

If you would rather avoid using this file, you can use the command-line flags `--root` and `--dotfile` to set these values.

How the standalone and Chrome builds interact

Powered by Gitiles source log blame