How to build chromium on Linux

Jump to bottom

Bin Miao edited this page on 22 Jun · 7 revisions

System requirements

- A 64-bit Intel machine with at least 8GB of RAM. More than 16GB is highly recommended.
- At least 100GB of free disk space.
- You must have Git and Python v2 installed already.

Most development is done on Ubuntu (currently 14.04, Trusty Tahr). There are some instructions for other distros below, but they are mostly unsupported.

Step-by-step guide

1. Install depot_tools

- Clone the ĶǴẏṙ ẤṙẤ ṙ ķẎ repository:
  `$ git clone https://chromium.googlesource.com/chromium/tools/depot_tools.git`
- Add depot_tools to the end of your PATH (you will probably want to put this in your ą ǒǘl ẎDŵ or ą ǒŊẎDŵ). Assuming you cloned depot_tools to /path/to/depot_tools:
  `$ export PATH="$PATH:/path/to/depot_tools"`
- When cloning depot_tools to your home directory **do not** use ~ on PATH, otherwise `gclient runhooks` will fail to run. Rather, you should use either $HOME or the absolute path:
  `$ export PATH="$PATH:${HOME}/depot_tools"`

2. Get the code

- Create a chromium directory for the checkout and change to it (you can call this whatever you like and put it wherever you like, as long as the full path has no spaces):
  `$ mkdir chromium && cd chromium`

- Clone the chromium-src repository:
  `$ git clone https://github.com/otcshare/chromium-src.git`
- Rename the chromium-src to src:
  `$ mv chromium-src src`
- Create .gclient file and edit the file to contain the following arguments:
  ```
  $ vim .gclient
  solutions = [
  {
  "url": "http://github.com/otcshare/chromium-src.git@webml",
  "managed": False,
  "name": "src",
  "deps_file": ".DEPS.git",
  "custom_deps": {},
  },
  ]
  ```

3. Config the proxy (proxy needed only)

- Make sure the proxy settings is correct in terminal:
  ```
  $ export http_proxy=http://proxy_addr:proxy_port
  $ export https_proxy=https://proxy_addr:proxy_port
  ```
- Create .boto file and edit the file to contain the following arguments:
  ```
  $ vim .boto
  [boto]
  proxy=http://proxy_addr
  proxy_port=proxy_port
  ```
- Add .boto config to system enviroment:
  `$ export NO_AUTH_BOTO_CONFIG="/path/to/.boto"`

4. Install additional build dependencies

- Once you have checked out the code, and assuming you're using Ubuntu, run build/install-build-deps.sh
  `$ cd src && ./build/install-build-deps.sh`
  > You may need to adjust the build dependencies for other distros. There are some notes at the end of this document, but we make no guarantees for their accuracy.

5. Update your checkout and run the hooks

- To update an existing checkout, you can run:
  `gclient sync`
- Once you've run install-build-deps at least once, you can now run the Chromium-specific hooks, which will download additional binaries and other things you might need:
  `$ gclient runhooks`

> Optional: You can also install API keys if you want your build to talk to some Google services, but this is not necessary for most development and testing purposes.

6. Setting up the build

- Chromium uses Ninja as its main build tool along with a tool called GN to generate .ninja files. You can create any number of build directories with different configurations. To create a build directory, run:
  `$ gn gen out/Default`
- To config the flags for WebNN, run gn args out/Default and edit the file to contain the following arguments:
  `$ gn args out/Default`
  `target_os = "linux"`
  `target_cpu = "x64"`
  `is_debug = false`
  `is_component_build = false`

> You only have to run this once for each new build directory, Ninja will update the build files as needed.
> You can replace Default with another name, but it should be a subdirectory of out.

7. Build Chromium

- Build Chromium (the “chrome” target) with Ninja using the command:
  `$ autoninja -C out/Default chrome unstable_deb`
  > (autoninja is a wrapper that automatically provides optimal values for the arguments passed to ninja.)

8. Run Chromium

- Once it is built, you can simply run the browser:
  `$ out/Default/chrome-wrapper`

Clone this wiki locally

```
https://github.com/intel/webml-polyfill.wiki.git
```