# Chromium build instruction

By Lin Dandan [dandan.lin@intel.com]

Steps:

1. Get the code
2. Install the depot\_tools!!

You must use git! So you must install the git

//sudo apt-get install git

When install done, fetch depot\_tools, which includes gclient, gcl, git-cl, repo and others.

When you want to learn how to use git, you can type the command as follows:

//sudo git help -a

//sudo git clone https://chromium.googlesource.com/chromium/tools/depot\_tools.git

Add depot\_tools to your PATH:

//sudo export PATH='pwd'/depot\_tools:"$PATH"

You can use path in your home, such as HOME/lindandan/ instead of ‘pwd’, the right path make sense!

The command export is for setting the environment!

(2)Initial checkout!

Install the git and the git-svn

//sudo apt-get install git-svn

Tell the git about yourself using setting some info!

// git config --global user.name "MyName" [for me, LinDandan]

//git config --global user.email "MyEmail"[for me, dandan.lin@intel.com]

// git config --global core.sutocrlf false

// git config --global core.filemode false

// git config --global branch.autosetuprebase always

(3)Actual checkout

Fetch the tools in depot\_tools and create a new folder named src

If not using git-svn, type the command as follows:

//fetch --nohooks chromium --nosvn=True

Else type the command as follows:

//fetch --nohooks chromium

Then do next:

//cd src

//git checkout master

If you are building for Linux only:

//build/install-build-deps.sh

If you are building for Android also:

//build/install-build-deps-android.sh

Then do next:

//gclient sync –nohooks

(4)Using the last known good revision(LKGR)

Editing the .gclient file in your checkout directory. At the bottom!

“safesync\_url”:”https://chromium-status.appspot.com/git-lkgr”

(5)Updating the code

Update your current branch with git pull followed by gclient sync. Note that if you are not on a branch, git pull won’t work, you need to use git fetch instead.

//cd “$CHROMIUM\_DIR”

//git pull

//gclient sync

To speed up updating, using more jobs, for example

//cd “$CHROMIUM\_DIR”

//git pull

//gclient sync –jobs=16

(6)Contributing

With git you should make all your changes in a local branch. Once your change is committed, you can delete this branch.

Create a local branch named “mywork” and make changes to it.

//cd src

//git new-branch mywork

//vi …

Commit your change locally (this doesn’t commit your change to the SVN or Git server)

//git commit –a

If you added new files, you should tell git so by running git add<files> before committing.

Upload your change for review

//git cl upload

Send a try job

//git cl try

1. First Time Build Bootstrap

Make sure your dependencies are up to date by running the install-build-deps.sh script.

//build/install-build-deps.sh

Before you build, you should also install API keys.

You can specify the API keys to use either when you build Chromium, or at runtime using environment variables.

1. GYP(configuring)

After gclient sync finishes, it will run gyp automatically to generate the ninja build files. For standard chromium builds, this automatic step is sufficient and you can start compiling.

To manually configure gyp, run gclient runhooks or run gyp directly via build/gyp\_chromium.

Not necessary.

//build/gyp\_chromium –Dflag1=value1 –Dflag2=value2

//export GYP\_DEFINES=”flag1=value1 flag2=value2”

//build/gyp\_chromium

1. Compilation

For me, just build the libraries and tests for chrome! You can change it!

The weird “src/” directory is an artifact of gclient.

//cd src

**Build just chrome**

//ninja –C out/Debug chrome

**Faster builds**

**//**

**A lot of instructions!!**

**Build every test**

//ninja –C out/Debug

The above builds all libraries and tests in all components. It will take hours.

**Clang builds**

**Output**

Executables are written in src/out/Debug/ for Debug builds, and src/out/Release/ for Release builds.

**Release mode**

//ninja –C out/Release chrome

**Seeing the commands**

//ninja –v –C out/Debug chrome

Using –v can see the actual commands that ninja is invoking.

**Clean builds**

All built files are put into the out/directory

//rm –rf out

Or

//ninja –C out/Debug –t clean

**Linker Crashes**

运行chromium需要sandbox

//Ninja –C out/Debug chrome chrome\_sandbox

还要设置好环境变量

//sudo cp out/Debug/chrome\_sandbox /usr/local/sbin/chrome-devel-sandbox

//sudo chown root:root /usr/local/sbin/chrome-devel-sandbox

//sudo chmod 4755 /usr/local/sbin/chrome-devel-sandbox

//export CHROME\_DEVEL\_SANDBOX=/usr/local/sbin/chrome-devel-sandbox

如果不进行设置sandbox的话，直接通过运行chrome

// src/out/Debug/chrome –no-sandbox