NOTE: You will need to get the ChromeOS **internal** manifest and repo (to access the stumpy overlays). Using the mini-layout seems fine.

Prepare your Newton machine:

- Flip the switch to developer mode. Switch is inside the Kensington lock port on the back. Direction seems different on PVT (dev = away from power) than DVT (dev = toward power).
- Turn it on and let the device erase its stateful partition.
- Reboot. It won't wait for you to hit Control-D -- that's OK.
- Get into a shell (Ctrl-Alt-right-arrow)
- Switch the firmware to developer mode. This will allow you to boot off a USB stick.
    - sudo chromeos-firmwareupdate --mode=todev

Build and install an image:

- Build a stumpy image for Aura:
    - Run "set_shared_user_password" with the usual "test0000". You'll need it.
    - Run "setup_board --board=stumpy --default", no "x86" required.
    - Run "export USE=aura" then "build_packages".
    - Run "build_image --noenable_rootfs_verification dev". Dev images are required, test isn't.
- Plug in your USB stick, turn on the Newton and hit Control-U at the startup screen to boot off USB.
- Run /usr/sbin/chromeos-install to install the image.
- Reboot
- Hit Ctrl-Alt-right-arrow and get to a shell.
- Make your filesystem writable: sudo mount -o remount,rw /
- Modify /etc/init/openssh-server.conf to remove the "#for_test " part of start on line, so that ssh is always running (basically the line should read as "start on...")
- Modify /etc/init/ui.conf to comment out all the lines in the ***post-stop script*** to not kill / unmount things.
    - This way your ssh sessions won't get killed.
    - Also, if session _manager crashes on boot, X will keep running.
- Reboot
- chmod 666 /home/chronos/.Xauthority

At this point you can use start_devserver and gmerge to develop, which is slow but simple (in jamescook's experience). Or use davemoore's more efficient method below:

Building chrome on your Z600 (not in chroot)

- Use ninja, goma and the component build to speed up builds (while these are optional, you will need to adjust the instructions below accordingly). Full builds from new source take about 7 minutes, full rebuilds of recently built source take about 3 minutes and compile / link of one file at about 5 seconds. So to generate a cros compatible image use

these with {chromiumos_root} substituted.
GYP_GENERATORS=ninja
GYP_DEFINES="chromeos=1 component=shared_library disable_webrtc=1 disable_nacl=1"
or in ~/.gyp/include.gypi

```
{
 'variables': {
  'disable_nacl': 1,
  'component': 'shared_library',
  'disable_webrtc': 1,
  'chromeos': 1,
  }
}
```

- To generate your ninja build setup
  ./build/gyp_chromium
- Debug is also supported,
- To build your chrome
  ninja chrome -C out/Release -j 750

Running chrome on cros

- ssh from your desktop to your cros as user "chronos"
- Then on cros copy / paste this set of lines (w/ {user} substituted)

```
sudo touch /var/run/disable_chrome_restart
# You'll get prompted here

bash
echo 1 > /var/lib/power_manager/disable_idle_suspend      ????
pkill -9 -P `pgrep session_manager` chrome
export DISPLAY=:0.0
alias cm="/tmp/c/src/out/Release/chrome --disable-seccomp-sandbox --parallel-auth --login-manager
--login-profile=user --log-level=0 --enable-logging=stderr --no-first-run --user-data-dir=/home/chronos
--in-chrome-auth --scroll-pixels=3 --enable-smooth-scrolling --force-compositing-mode"
alias cl="/tmp/c/src/out/Release/chrome --disable-sync --login-user=testcros2@gmail.com --log-level=0
--enable-logging=stderr --no-first-run --user-data-dir=/home/chronos --login-profile=test
--scroll-pixels=3 --test-type=foo --enable-smooth-scrolling --force-compositing-mode"
# Debug versions
alias dm="/tmp/c/src/out/Debug/chrome --disable-seccomp-sandbox --parallel-auth --login-manager
--login-profile=user --log-level=0 --enable-logging=stderr --no-first-run --user-data-dir=/home/chronos
--in-chrome-auth --scroll-pixels=3 --enable-smooth-scrolling --force-compositing-mode"
alias dl="/tmp/c/src/out/Debug/chrome --disable-sync --login-user=testcros2@gmail.com --log-level=0
--enable-logging=stderr --no-first-run --user-data-dir=/home/chronos --login-profile=test
--scroll-pixels=3 --test-type=foo --enable-smooth-scrolling --force-compositing-mode"
mkdir /tmp/c
```

- If you have a password on your chronos user account you'll be prompted for it now.
- Then copy/paste this line (modified for your desktop {user} / {host} / {chrome_root})

  ```
  sshfs {user}@{host}:{path_to_chrome_root} /tmp/c
  # e.g. sshfs jamescook@jamescook.mtv.corp:/w/chrome /tmp/crm -
  ```
- If this is the first time you've done this since creating your build output directory you'll want to do this to allow chrome to find libcros.

  ```
  ln -s /opt/google/chrome/chromeos /tmp/c/src/out/Release
  ln -s /opt/google/chrome/chromeos /tmp/c/src/out/Debug
  ```
- Then run chrome with

  ```
  USER=chronos cl
  ```
- That will launch a chrome with your logged in user. Chrome will start in 5 seconds or so. cm will launch chrome with the login manager.

Debugging on stumpy

- Do the above but build the Debug version and then in your ssh session

  ```
  cd /tmp/c/src/out/Debug
  gdb chrome
  ```
  This will take longer, but still only about 30 seconds. You can set breakpoints, view code, etc... Others may prefer gdbserver / eclipse.

Running browser_tests

1. You may need to kill the session manager once (jamescook needs this, others don't):

   ```
   sudo killall session_manager
   ```
2. Then you can run tests normally

   ```
   cd /tmp/c/src
   out/Debug/browser_tests
   ```

You may need an updated (i.e. newer than the one in gLucid) version of Xvfb if you want to be able to use it to run browser_tests. See https://groups.google.com/a/google.com/forum/?fromgroups#!topic/chrome-aura/O8PyrhYZL6E .