* Building AOSP code for CTS:

For building source code for CTS run the following script from source code directory with mentioned parameters:

# ./build.sh N1 user cts

* Building AOSP code for VTS:

For building source code for VTS run the following script from source code directory with mentioned parameters:

# ./build.sh N1 user vts

* Setting-up CTS:
* Host Machine Setup:
* Note: CTS currently supports 64-bit Linux and Mac OS host machines. CTS will not work on Windows OS.
* Download [Android Debug Bridge ( HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html"adb HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html" HYPERLINK "http://developer.android.com/tools/help/adb.html")](http://developer.android.com/tools/help/adb.html)  – # sudo apt-get install android-tools-adb
* Download [Android Asset Packaging Tool (AAPT)](http://developer.android.com/guide/topics/manifest/uses-feature-element.html) – # sudo apt-get install aapt
* Download Java Development Kit (JDK) – <http://openjdk.java.net/install/>
* Download and open the latest version of the [CTS Media Files](https://source.android.com/compatibility/cts/downloads.html) – [https://source.android.com/compatibility/cts/downloads.html#cts-media-files](https://source.android.com/compatibility/cts/downloads.html)
* Setting-up VTS:
* Host Machine Setup:
* <https://source.android.com/compatibility/vts/systems>
* sudo apt-get install python-dev
* Install Protocol Buffer tools (for Python):
* sudo apt-get install python-protobu
* sudo apt-get install protobuf-compiler
* Install Python virtual environment-related tools:
* sudo apt-get install python-virtualenv
* sudo apt-get install python-pip
* Device/Emulator Setup:
* Make sure that your device has been flashed with a user build (Android 4.0 and later) and running fine.
* Copy the CTS media files to your device:
* Navigate (cd) to the path where media files are downloaded and unzipped to.
* Change the file permissions – # chmod u+x copy\_media.sh
* Run – # ./copy\_media.sh all
* Factory data reset the device: Settings > Backup & reset > Factory data reset

Warning: This will erase all user data from the device.

* Set your device's language to English (United States) from: Settings > Language & input > Language
* Turn on the location setting if there is a GPS or Wi-Fi / Cellular network feature on the device: Settings > Location > On
* Make sure no lock pattern or password is set on the device: Settings > Security > Screen lock > None
* Make sure the time is set to 12-hour format: Settings > Date & time > Use 24-hour format > Off
* Select: Settings > Developer options > Stay Awake > On
* Launch the browser and dismiss any startup/setup screen
* Enable USB debugging on your device: Settings > Developer options > USB debugging

Note: On Android 4.2 and later, Developer Options is hidden by default. To make them available, go to Settings > About phone and tap Build number seven times. Return to the previous screen to find Developer options.

* Connect the desktop machine that will be used to test the device with a USB cable

Note: When you connect a device running Android 4.2.2 or later to your computer, the system shows a dialog asking whether to accept an RSA key that allows debugging through this computer. Select Allow USB debugging.

Running CTS:

Connect at least one device.

Press the home button to set the device to the home screen at the start of CTS.

While a device is running tests, it must not be used for any other tasks and must be kept in a stationary position (to avoid triggering sensor activity) with the cameras pointing at an object that could be focused.

Do not press any keys on the device while the CTS is running. Pressing keys or touching the screen of a test device will interfere with the running tests and may lead to test failures.

Navigate (cd) to this path – aosp-neo-n1/out/host/linux-x86/cts/android-cts/tools

Change the file permissions – # chmod 777 cts-tradefed

Run – # ./cts-tradefed

Give command – run cts --skip-preconditions --disable-reboot --skip- device-info

Running VTS:

run vts For default VTS tests

run vts-hal For default VTS HAL (hardware abstraction layer) tests

run vts-kernel For default VTS kernel tests

View test progress and results reported on the console.