/Hi all,

The below steps are used to apply privileges to pull and push codes on acs-main. The red parts are commands or content can be directly use, the yellow parts need to replace with your own info.

If something confused, please checking in the <https://securewiki.ith.intel.com/display/GMIN/Ubuntu+14.04+Workstation+Setup> (Most steps come from this link) for more details,

or discuss with me directly.

1. http://eam.intel.comEnterprise Applications  DevTools  ACS BUNDLE  Software Developer

2.  $sudo vim /etc/environment, add below proxy

     http\_proxy=http://proxy.cd.intel.com:911

     https\_proxy=https://proxy.cd.intel.com:911

     no\_proxy=localhost,.intel.com,127.0.0.0/8,172.16.0.0/20,192.168.0.0/16,10.0.0.0/8

     source /etc/environment

3.$sudo vim /etc/sudoers, add below

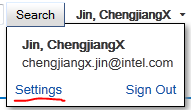
  Defaults  env\_keep += "http\_proxy https\_proxy no\_proxy"

  Use : w !sudo tee % to save

4.$java –version, to check if java version >=1.7, if <1.7 or not installed, please install the >=1.7 version jdk, such as $sudo apt-get install openjdk-7-jdk

5.$sudo apt-get install bison g++-multilib git gperf libxml2-utils ccache lib32z1 lib32ncurses5 lib32bz2-1.0 dos2unix

6. $cat ~/.ssh/id\_rsa.pub, copy it to [https://android.intel.com/#/q/status:open,n,z](https://android.intel.com/" \l "/q/status:open,n,z) Sign In(Use windows log in account to sign in)Settings(as below)SSH Public KeysAdd Key…



If no ~/.ssh/id\_rsa.pub, generate it using $ssh-keygen

7.$vim ~/.ssh/config, add below, and change id\_robot05\_rsa to your key, usually id\_rsa, change the jinchenx to your windows log on accout(IDSID).

StrictHostKeyChecking no

IdentityFile=~/.ssh/**id\_robot05\_rsa**

host otc-android.intel.com

     port 29418

     user **jinchenx**

host review.otc-android.intel.com

     port 29418

     user **jinchenx**

host aia-review.intel.com

     port 29418

     user **jinchenx**

host android.intel.com

     port 29418

     user **jinchenx**

host jfumg-gcrmirror.jf.intel.com

     port 29418

     user **jinchenx**

host android-mirror-sh.devtools.intel.com

     port 29418

     user **jinchenx**

host git-ccr-1.devtools.intel.com

     port 29418

     user **jinchenx**

host android-mirror-or.devtools.intel.com

     port 29418

     user **jinchenx**

host ncsgit002.nc.intel.com

     port 29418

     user **jinchenx**

**9.**$ chmod 600 ~/.ssh/config

10.$sudo vim ~/.gitconfig , add blow, and change the yellow part to your corresponding IDSID or email.

[review "<https://android.intel.com>"]

        username = **jinchenx**

[user]

        email = [**chengjiangx.jin@intel.com**](mailto:chengjiangx.jin@intel.com)

        name = **jinchenx**

[color]

        ui = auto

[url "ssh://**jinchenx**@android-mirror-sh.devtools.intel.com:29418/"]

        insteadOf=git://android.intel.com/

        insteadOf=ssh://android.intel.com/

        insteadOf=ssh://android.intel.com:29418/

        insteadOf=git://android-mirror-sh.devtools.intel.com/

        insteadOf=ssh://android-mirror-sh.devtools.intel.com/

        insteadOf=git://android-mirror.devtools.intel.com/

        insteadOf=ssh://android-mirror.devtools.intel.com/

        insteadOf=ssh://android-mirror.devtools.intel.com:29418/

        insteadOf=git://jfumg-gcrmirror.jf.intel.com/

        insteadOf=ssh://jfumg-gcrmirror.jf.intel.com/

        insteadOf=ssh://jfumg-gcrmirror.jf.intel.com:29418/

[url "ssh://**jinchenx**@android.intel.com:29418/"]

        pushInsteadOf=ssh://android.intel.com/

        pushInsteadOf=ssh://android.intel.com:29418/

        pushInsteadOf=ssh://jfumg-gcrmirror.jf.intel.com/

        pushInsteadOf=ssh://jfumg-gcrmirror.jf.intel.com:29418/

        pushInsteadOf=ssh://android-mirror.devtools.intel.com/

        pushInsteadOf=ssh://android-mirror.devtools.intel.com:29418/

        pushInsteadOf=ssh://android-mirror-sh.devtools.intel.com/

        pushInsteadOf=ssh://android-mirror-sh.devtools.intel.com:29418/

[cat]

[diff]

         tool = meld

[merge]

         tool = meld

11.$ ssh-keyscan -p 29418 android.intel.com >> ~/.ssh/known\_hosts;ssh-keyscan -p 29418 android-mirror-sh.devtools.intel.com >> ~/.ssh/known\_hosts

12.$sudo mkdir /usr/share/ca-certificates/intel;sudo wget <http://certificates.intel.com/repository/certificates/Intel%20Root%20Certificate%20Chain%20Base64.zip> -O /usr/share/ca-certificates/intel/IntelChain.zip;cd /usr/share/ca-certificates/intel/;sudo unzip IntelChain.zip;sudo find -name "\* \*" -type f | sudo rename 's/ /\_/g';sudo update-ca-certificates

# If it does not work with previous line, try

$sudo dpkg-reconfigure ca-certificates

13.$ mkdir -p ~/bin;curl --noproxy android.intel.com <https://android.intel.com/static/repo> > ~/bin/repo;chmod a+x ~/bin/repo

14. mkdir -p ~/bin/ccache\_dir

15. vim ~/.bashrc, add below

export PATH=$PATH:$HOME/bin

export USE\_CCACHE=1

export CCACHE\_DIR=$HOME/ccache\_dir

16.$source ~/.bashrc

17. $sudo apt-get install preload prelink zram-config;sudo prelink -amR

18.$mkdir -p ~/acs\_main;cd

19. cd ~/acs\_main

20.The below command will create a *.repo* folder

$repo init -u ssh://android.intel.com/manifest -b platform/android/main -m acs-main

21.The below command will pull asc-main code to your current directory.

$ repo sync -j8

22.To submit a commit, we need to start a new branch, below command started a new branch named PRC\_branch.

$repo start *PRC\_branch* --all

23. Now we can make some changes to our code, such as add, delete, update, etc.

24. After we made a change, we need add a comment to the change, as below command

$git add .

$git commit -am “xxxx”

25.The above step23 added change to local, use the below command to push the change to remote repo

$repo upload

26. After step24, there will be link generated to point to the change in gerrit, something like this:

<https://android.intel.com/#/c/378033/>

27. Open the link, add the code reviewers, such as [shuang.wan@intel.com](mailto:shuang.wan@intel.com) or [shaofeng.tang@intel.com](mailto:shaofeng.tang@intel.com)

run acs

python ACS.py --device\_models |grep ECS

sudo easy\_install lxml

sudo easy\_install arrow

cd acs/src/\_Tools/setup\_manager/setups

sudo python setup\_mgr.py install devel

sudo pip uninstall uiautomator

acs\_test\_suites/OTC/TC/PRC/TC/

acs\_test\_suites/OTC/CAMPAIGN/SystemFunctional/LOLLIPOP/MR\_1/PRC/

cp -r acs\_test\_suites/OTC/ acs/src/\_ExecutionConfig/

cp -r otcqa-oat-l /testplan/acs/TC/ acs/src/\_ExecutionConfig/OTC/TC/PRC/testplan/

cp -r otcqa-oat-l/testplan/\*/\*.conf acs/src/\_ExecutionConfig/OTC/TC/PRC/testplan/\*/\*

https://tlsstor001.tl.intel.com/artifactory/webapp/browserepo.html?6

Artifactory/acs\_test\_artifacts/[OTC\_Relases   OTC\_AFT\_ww\_Release]

<https://mcg-depot.intel.com/artifactory/acs/releases/OTC/>

acs====unzip acs\_core.zip=>acs\_fwk /\_ExecutionConfig/CAMPAIGN+TC+bench\_config\_BT\_HW\_campaigns.xml+myBench\_usb.xml

acs====unzip acs\_core.zip=> acs\_fwk/src

acs\_test\_script=unzip acs\_core.zip----acs\_fwk====acs\_test\_script

acs\_test\_suites====unzip campaigns.zip----- BENCHCFG NDG PCCG SI FT OSBV PM ST WWAN

====BOOT \_Catalogs conf \_Configs ExtraLibs \_\_init\_\_.py OTC SCRIPT\_HELPERS

acs\_test\_suites/OTC/TC/PRC/

acs\_test\_suites/OTC/

上传包：<https://mcg-depot.intel.com/artifactory/webapp/deployartifact.html?5>

<http://artifactory/acs_test_artifacts/OTC_AFT_ww_Release/test/>

python ACS.py -d ECS-Android-LLP -c OTC/CAMPAIGN/SystemFunctional/LOLLIPOP/MR\_1/PRC/ Sensor.MALATA10.xml -b OTC/BENCHCFG/myBench\_usb.xml --dev --user=chengjiangx.jin@intel.com --awr\_server=acs.tl.intel.com

下载包：<https://mcg-depot.intel.com/artifactory/cactus-absp-tl/acs-engineering/200/>

<http://acs.tl.intel.com:8080/acs-bundle-documentation/acs_developer_doc/acs_ci/release_builders/release_builder.html>

https://buildbot.tl.intel.com/absp/builders/acs-engineering

./pretest\_campaign\_generator\_v1.py ../../testplan

<http://campaign.tl.intel.com/#/>

Here the wiki of TCR: <https://wiki.ith.intel.com/display/DRD/Campaign+-+Rest+API>

: <http://acs.tl.intel.com:8080/acs-bundle-documentation/acs_user_doc/getting_started/reporting/TCR-TAAS.html>

engineerbuild

<https://buildbot.tl.intel.com/absp/builders/acs-engineering/>

engineerbuild result

https://buildbot.tl.intel.com/absp/builders/acs-engineering/builds/323

<https://mcg-depot.intel.com/artifactory/cactus-absp-tl/acs-engineering/323>

add chrome\_cast into query

copy oat campaign\_generator\_v1.py to acs code

Add pretest, ww02 release, latest folder to release

Add Davinci and CTS packages to release

Upload package

<https://tlsstor001.tl.intel.com/artifactory/webapp/home.html?1>

[https://tlsstor001.tl.intel.com/artifactory/simple/acs\_test\_artifacts/](https://tlsstor001.tl.intel.com/artifactory/simple/acs_test_artifacts/OTC_Android_Auto_Test_Suite/pre-release/)[OTC\_Android\_Auto\_Test\_Suite/pre-release/](https://tlsstor001.tl.intel.com/artifactory/simple/acs_test_artifacts/OTC_Android_Auto_Test_Suite/pre-release/)

curl -i -u jinchenx:AP7unxFDqUTtjX8s38JxiGwRifY -T ACS\_v.16.14.4\_1.zip "https://tlsstor001.tl.intel.com/artifactory/acs\_test\_artifacts/OTC\_Android\_Auto\_Test\_Suite/pre-release/" -k

Official link: <https://mcg-depot.intel.com/artifactory/acs_test_artifacts/OTC_Android_Auto_Test_Suite/latest/>

Backup link: <https://tlsstor001.tl.intel.com/artifactory/acs_test_artifacts/OTC_Android_Auto_Test_Suite/latest/>

Livereport

http://otc-livereport.sh.intel.com:8080/smartserver/index.html#/smartserver/group/1

hotfix sheet:

<https://docs.google.com/spreadsheets/d/1pzTSkP3g355J55DRDD-P_8ahdV046-EMrggVM4qA4LY/edit#gid=1317489657>

lsusb

udevadm info -a -n /dev/bus/usb/001/012

/devices/pci0000:00/0000:00:1a.0/usb1/1-1/1-1.4

AdevicesApci0000B00A0000B00B1a.0Ausb1A1-1A1-1.4

sudo docker run -dit --name AdevicesApci0000B00A0000B00B1a.0Ausb1A1-1A1-1.4 --device=/dev/bus/usb/001/011 auto/testbot

sudo docker ps -a

sudo docker rm -f AdevicesApci0000B00A0000B00B1a.0Ausb1A1-1A1-1.4

acs wiki

http://otcqa.sh.intel.com/wiki/Automation\_test\_suites\_for\_system\_functional\_validation#Environment\_configuration

./acs\_release\_management/external\_release/aahm/clean\_rules.txt  
flash sofia 3grmrd6s image

~/bin/DownloadTool/DownloadTool -c1 --erase-mode 2 --library ~/bin/DownloadTool/libDownloadTool.so ~/Downloads/s3gr10m6s-flashfiles-L1l000068/\*.fls

L： <http://otcqarpt.jf.intel.com/irda/>

M: <https://otcqarpt.jf.intel.com/OAM/>

Kvm

apt-get install qemu-kvm libvirt-bin virt-manager bridge-utils

sudo virt-install --name ubuntu14\_vm\_1 --hvm --ram 1024 --vcpus 1 --disk path=/home/irdatest/kvm/ubuntu14\_vm\_1/disk.img,size=10 --network network:default --accelerate --vnc --vncport=5911 --cdrom /home/irdatest/kvm/ubuntu-14.04-desktop-amd64.iso –d

virsh list --all

add m plans:

BYT\_EMDOOR\_I8170

BYT\_EMDOOR\_I8880

BYT\_Tongfang\_16

IP3\_T15

Delete plans:

BYT\_EMDOOR\_I8811

BYT\_EMDOOR\_I8889

ECSEDU

rerun ACS failed case flow as below:

1.       Generate rerun campaign

$ python ACS.py --eft  $CTEASSET/\_Reports

2.       Start to rerun campaign like below:

$ python ACS.py -d ECS-Android-LLP -c Multimedia\_Camera.SOFIA3GR\_MRD6S.merge\_FAILED\_TESTS.xml -b OTC/BENCHCFG/myBench\_usb.xml --dev [--user=aft.cti@intel.com](mailto:--user=aft.cti@intel.com) --awr\_server=acs.tl.intel.com

live report server

<http://otc-livereport.sh.intel.com:8080/smartserver/index.html#/smartserver/index>

hpalm

<https://hpalm.intel.com/qcbin/start_a.jsp>

<https://hpalm-pre.intel.com/qcbin/start_a.jsp>

ywang111   
Testpre

3GRMRD6S L

BYT CR ET

T3 bios

<https://wiki.ith.intel.com/display/ANDROIDSI/CherryTrail+T3+MRD>

adb reboot fastboot

fastboot flashing unlock

fastboot flash sfu xxx.fv

fastboot reboot

ART

<http://otcqa.sh.intel.com/IRDA/ART/>

metrics server:

username/passwd: sdk-test/123456

mysql ： root,  123456

t4 image

https://mcg-depot.intel.com/artifactory/cactus-absp-jf/build/eng-builds/m\_mr1/PSI/weekly/

<https://mcg-depot.intel.com/artifactory/cactus-absp-jf/build/eng-builds/m_mr1/PSI/weekly/2016_WW09/r2_cht_ffd/userdebug/>

dediprog stage2

fastboot flash fw\_stage2 xxxxxx.bin

for i in `ls \_\*`;do endline=`cat $i |wc -l`;let endline=endline-2;sed -i 50','$endline'd' $i;done