**SOP- Android APK Build Automation in Jenkins**

**Jenkins URL:** <http://hme-build.westus.cloudapp.azure.com:8080/>

Machine generated alternative text:
4hrOmin-#100 NIA 17mm
4hr19min-jfl 2mo1Odays-. 18mm
aaa aescnpjjjj
Jenkins
2. Android_APK_Build_LIVE is part of Post-Build action in UAT(1) Build and it’s getting triggered automatically once UAT gets completed
New item
People
Build History S Last Success Last Failure Last Duration
Project Relationship
Ci Check File Fingerprint
0
0
1. Android_APK_Build_UAT build is running on it’s own as it has been scheduled in Periodic Build Option and it get’s triggered at 6:00 PM PST everyday

**Android APK Build Jobs**

1. **Android\_APK\_Build\_UAT** (**\***This build is scheduled as **Cronjob** and it runs first, *as soon as UAT gets completed LIVE gets triggered automatically* so we don't need to run LIVE manually, means both the job are getting triggered here in Jenkins automatically at 6:00 PM PST)

1. **Android\_APK\_Build\_LIVE** (**\***This Build is under *post-build action in UAT build Job so we don't need to trigger this build neither manually nor automatically*)

**Step 1:**

**Android Automation Build Job Name:** Android\_APK\_Build\_UAT

e Jenkins 
Source Code Management 
Build Triggers 
Build Environment 
Build 
_ 4 search 
Post-build Actions 
Suresh Kumar Maurya 
Advanced... 
I log out 
Jenkins 
Android APK Build LIAT 
General 
Description 
[Plain text] Preview 
CJ Discard old builds 
CJ GitHub project 
CJ Notify when Job configuration changes 
CJ This build requires lockable resources 
CJ Restrict build execution causes 
CJ This project is parameterised 
CJ Throttle builds 
CJ Disable this project 
CJ Execute concurrent builds if necessary 
Apply 

**Step 2:**

**We are using SVN (Subversion) to Download the Android Code:**

**URL :** <https://192.175.52.181/svn/HME/QSR/AZURE/HMEPulse_MobileApp/HMEPulse_MobileApp_Client/trunk>

Jenkins 
Android APK Build LIAT 
General 
Source Code Management 
Build Triggers 
Build Environment 
Build 
Post-build Actions 
Source Code Management 
O 
None 
O 
Git 
Subversion 
Modules 
Repository URL 
Credentials 
Local module directory 
Repository depth 
Ignore externals 
Cancel process on externals fail 
Add module. 
d additional credentials... 
https:/\192.17S.52.181\svn/HME/QSR\AZURE/HMEPulse MobileApp,'HMEPulse Mob 
Ad* 
infinity 
Apply 
u 
'svn update' as much as possible 

**Step 3:**

Android Build is scheduled at 6:PM PST which runs automatically every day.

H 18 \* \* \*

Jenkins 
Android APK Build LIAT 
General 
Source Code Management 
Build Triggers 
Build Environment 
Build 
Post-build Actions 
Build Triggers 
CJ Trigger builds remotely (e.g„ from scripts) 
CJ Build after other projects are built 
Build periodically 
Schedule 
Would last have run at Sunday, December 9, 2018 6:17:45 PM PST; would next run at Monday, December 10, 2018 6:17:45 PM PST 
CJ GitHub hook trigger for GITScm polling 
D Poll scb,l 

**Step 4:**

**All the android build's batch commands and steps are mentioned here in the Build Environment Field:**

cd android

gradlew clean && gradlew assembleRelease --console plain

Jenkins 
Android APK Build LIAT 
General 
Build 
Source Code Management 
Build Triggers 
Build Environment 
Build 
Post-build Actions 
Execute Windows batch command 
Command cd android 
c lean 
assembleReIease 
- -console plain 

**Step 5:**

**Post-build Actions:**

**Archive the Artifacts:**

**Files to Archive:** android/app/build/outputs/apk/\*.apk

Jenkins 
Android APK Build LIAT 
General 
Source Code Management 
Build Triggers 
Build Environment 
Build 
Post-build Actions 
Post-build Actions 
Archive the artifacts 
Files to archive .apk 
Build other projects 
Projects to build Android APK Build LIVE 
@ Trigger only if build is stable 
Trigger even if the build is unstable 
O 
Trigger even if the build fails 
O 
Add post-build action 
Apply 
Advanced... 

**Commands for Android\_APK\_Build\_LIVE:**

**Files which have been edited before build command are being executed in Android\_APK\_Build\_LIVE:**

cd src/services/

(Get-Content ".\webServices.js") |

**1)**

Foreach-Object {$\_.replace("https://hme-uat2-leaderboard.azurewebsites.net", "https://hme-live2-leaderboard.azurewebsites.net")} |

Foreach-Object {$\_.replace("kZMb80K91$Q1", "Kfd149M$19\*1")} |

Foreach-Object {$\_.replace("Basic aG1lX3RlYW06a1pNYjgwSzkxJFEx", "Basic aG1lX3RlYW06S2ZkMTQ5TSQxOSox")} |

Foreach-Object {$\_.replace("https://registerpushnotification.azurewebsites.net/api/hme/store/aggregate/registernotification", "https://hme-live-registerpushnotification.azurewebsites.net/api/hme/store/aggregate/registernotification")} |

Foreach-Object {$\_.replace("971161549671", "860124029336")} |

Set-Content ".\webServices.js"

**2)**

cd android/app/src/main/java/com/hm/

(Get-Content .\NotificationSettings.java) |

Foreach-Object {$\_.replace("971161549671", "860124029336")} |

Foreach-Object {$\_.replace("hme-uat-notificationhub", "hme-live-notificationhub")} |

Foreach-Object {$\_.replace("Endpoint=sb://hme-pulse-notif-uat.servicebus.windows.net/;SharedAccessKeyName=DefaultListenSharedAccessSignature;SharedAccessKey=Ui+HpTNvTzGsyxeQoO+xhfcF/tkoD46eNuMjHjnPGJk=", "Endpoint=sb://hme-live-notificationhub-namespace.servicebus.windows.net/;SharedAccessKeyName=DefaultListenSharedAccessSignature;SharedAccessKey=YhyjJo2cw6sQ9cTgxlG0VZN86yk6EL5Qu5X3LmdkHyc=")} |

Foreach-Object {$\_.replace("Endpoint=sb://hme-pulse-notif-dev.servicebus.windows.net/;SharedAccessKeyName=DefaultFullSharedAccessSignature;SharedAccessKey=BnWWIwN+lz+Ebs2OVFPEJoWJNwtJkF6Aq2oOxKXWHmI=", "Endpoint=sb://hme-live-notificationhub-namespace.servicebus.windows.net/;SharedAccessKeyName=DefaultFullSharedAccessSignature;SharedAccessKey=Az73/k+aq2cW2ADjkeIANZoej9IJXFwEZ5woKNz90os=")} |

Set-Content ".\NotificationSettings.java"

**Command which have been used for LIVE:**

cd android

gradlew clean && gradlew assembleRelease

cd android/app/build/outputs/apk/

ren app-release.apk live-app-release.apk

**Monitoring Android\_APK\_Build\_UAT and LIVE both:**

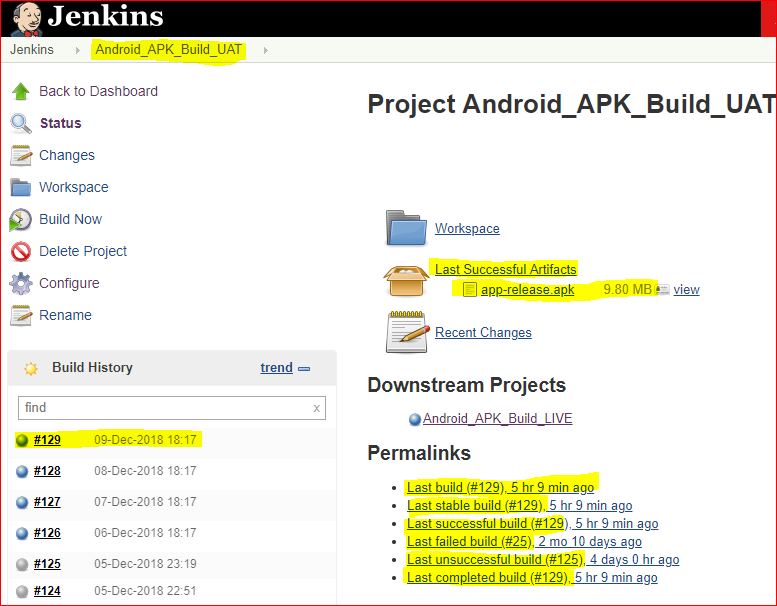
**To check console log:**

**Here you can easily see Last Successful Build Details:**

**#129 is successful build has been completed.**

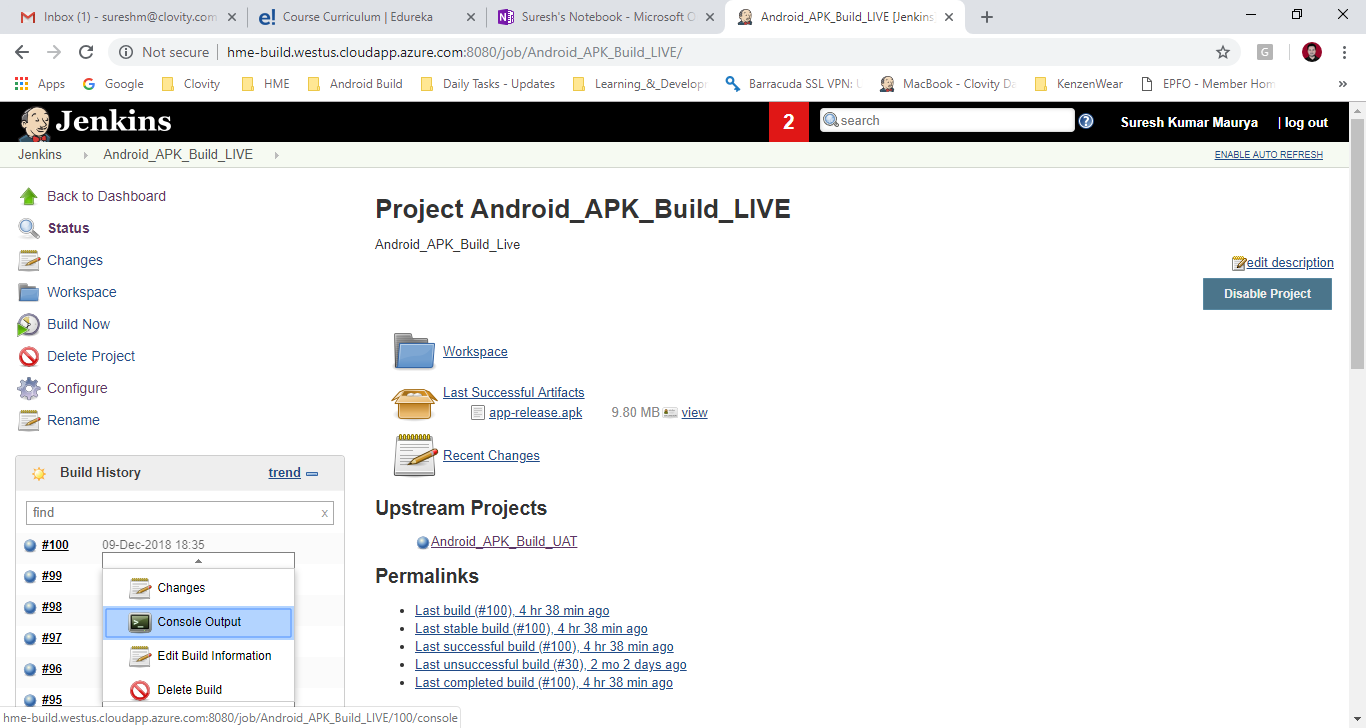
**Last Successful Artifacts 9.80 MB which can be downloaded.**

**Downstream Project Detail is also mention in this window which you can see easily**



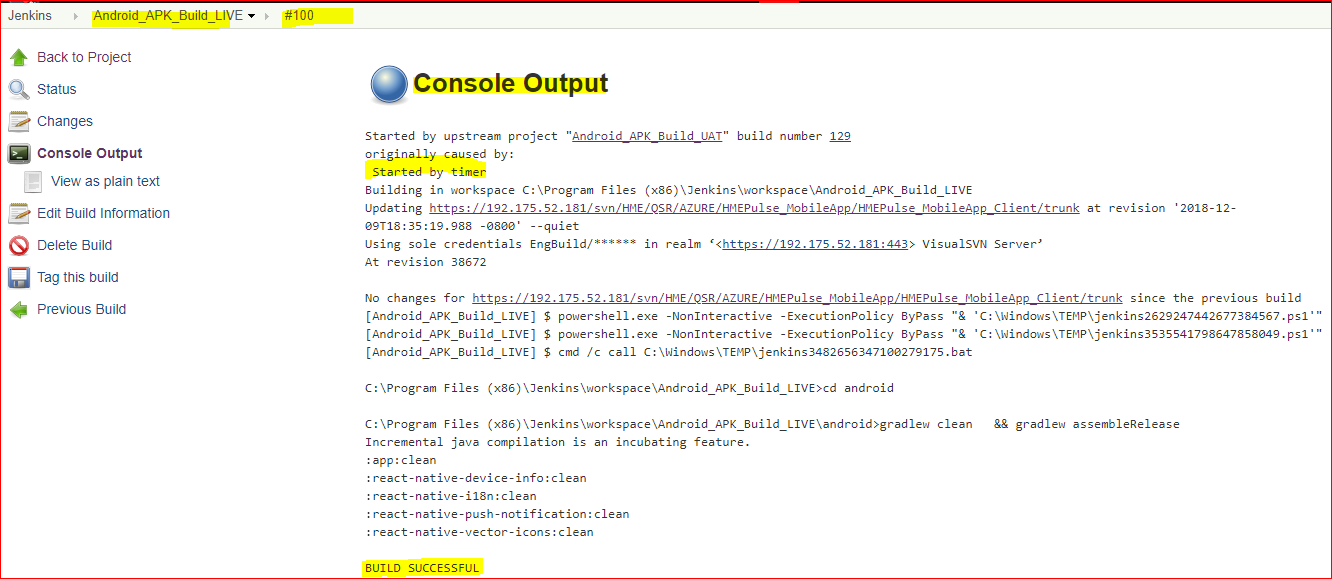
**1 Click on the Build Job – Android\_APK\_Build\_UAT or LIVE**

**2 Go to Build History and find out the Build No. starting with #100 and Click drop-down on Console Output**



You can easily see and identify that the build has been **started by timer**, **BUILD SUCCESSFUL**

**\*If Build has been completed successfully you will see a message: BUILD SUCCESSFUL**



**\*If Build has been completed as Failed you will see a message: BUILD FAILED**

