Task Manager Android Application

User Manual

November 12, 2020

Slippery Rock University

Xavier Julius

Olivia Warner

Table of Contents

About Android Task Manager

The application gathers the details, services, and CPU utilization of a rooted device. It will take that information , break it up, and display it to the user with a scrollable user interface. From this application, you can also explore the different permissions, memory and even force stop or completely uninstall it. The advantage with this application is that you are able to gather the details, services, and CPU information for a device. The disadvantage of this is that you have to be a root user to be able to perform these tasks on the device. A rooted emulator we found easiest for all users instead of putting their phone at risk by trying to root it.

1.0 Setting Up Android Studio

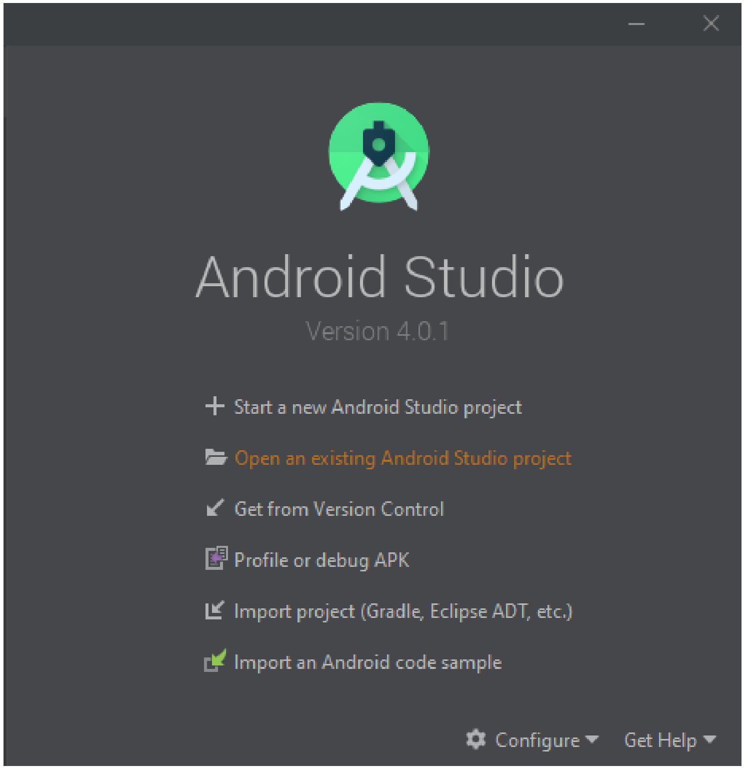
Android Studio is the IDE platform used to create this application. It is important to understand how to open the project for building and running purposes. This section shows the steps of downloading and opening up an existing project.

1.1 Install Android Studio

1. Download Android Studio
2. Download Link: https://developer.android.com/studio/
3. Select the download appropriate for your device (Windows, Mac, etc.)
4. Run the installation (.exe) file

1.2 Open Task Manager Project

1. When opening Android Studio for the first time, you should see a screen similar to this:


Android Studio
Version 4.0.1
+ Start a new Android Studio project
Open an existing Android Studio project
Get from Version Control
Profile or debug APK
Import project (Gradle, Eclipse ADT, etc.)
Import an Android code sample
Configure
Get Help

2. Select "`Open an existing Android Studio Project`"
3. Find and Select the Task Manager Project
4. This will open the project window where you may navigate through the different files and soon run the application.

2.0 Setting Up MEmu Play Emulator

To run the Task Manager application, you need to be able to install and root an emulator. The different tasks that are being done in the application require root privileges. This section will go through the details instructions to installing, setting up and running the MEmu Play Emulator.

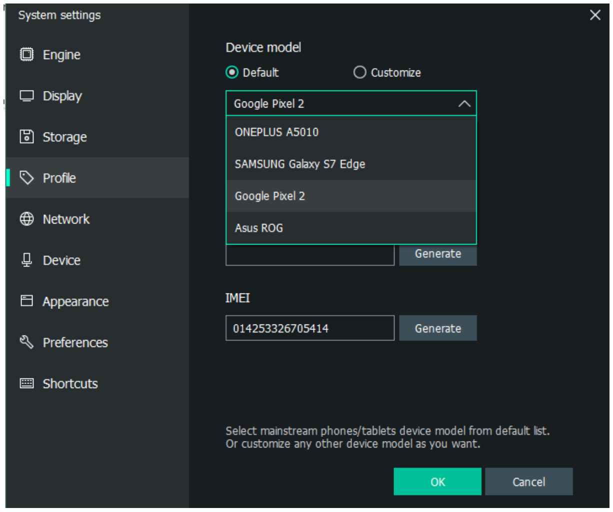
2.1 Install MEmu Play Emulator

1. Download the MEmu Play Emulator
   a. Download Link: https://www.memuplay.com/
   b. Save the Memu-Installer.exe file
   c. Run the Memu-Installer.exe file
      i. Running this file prompt your computer to begin the installation. This may take a few minutes.
2. Finish Installation
   a. Select your language > English (United States)
   b. Select your keyboard layout > English (US)
   c. Your emulator should now be installed correctly, where you may navigate around the device and get a feel of all of its functions and capabilities.

2.2 Change MEmu Device

In this emulator, you have the ability to change what device you are going to use for your application.

1. You can change the device by going your device `settings` on the right tool bar. The icon will look like this: ⚙
2. From there navigate to "`Profile`"



Device model

● Default ○ Customize

Google Pixel 2

ONEPLUS A5010
SAMSUNG Galaxy S7 Edge
Google Pixel 2
Asus ROG

IMEI

014253326705414

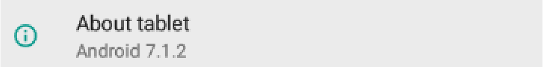
Select mainstream phones/tablets device model from default list.
Or customize any other device model as you want.

3. There you can change the device model. You have 4 different options to choose from.

4. Click `OK` to save the changes.

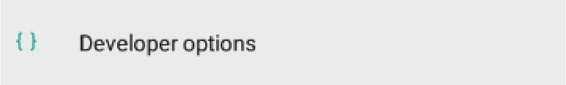
2.3 Enable USB Debugging

1. Go to the `Settings` App on MEmu
2. Scroll down all the way down to `About Tablet`

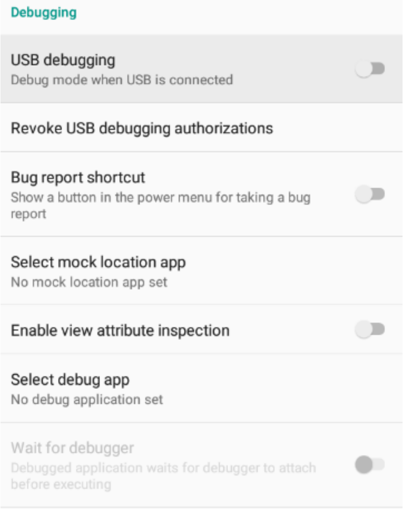


About tablet
Android 7.1.2

3. Find the `Build Number` and tap/click it 7 times to enable `Developer Mode`
   a. You will get a message that tells you `“You are a developer”` at the bottom of the screen.
4. Select `Developer Options` that is above `About Tablet`



5. Scroll down until you find the different `Debugging Options`
6. Select `USB Debugging to ON`



Debugging

USB debugging
Debug mode when USB is connected

Revoke USB debugging authorizations

Bug report shortcut
Show a button in the power menu for taking a bug report

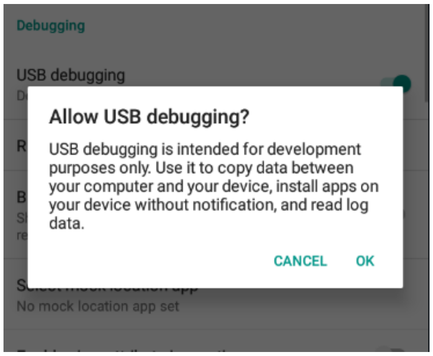
Select mock location app
No mock location app set

Enable view attribute inspection

Select debug app
No debug application set

Wait for debugger
Debugged application waits for debugger to attach before executing

7. You will then be greeted with a pop-up message that asks if you allow this feature



Allow USB debugging?

USB debugging is intended for development purposes only. Use it to copy data between your computer and your device, install apps on your device without notification, and read log data.

8. Select `OK` to allow `USB Debugging`
    a. USB Debugging allows Android Studio to communicate with MEmu Play Emulator. This will give you the option to run the application on the emulator now.

2.4 Enable Root

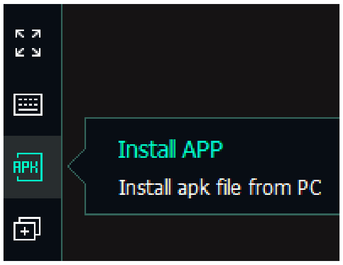
1. Select the `Emulator Settings` button on the toolbar to the right
2. Under the `Engine Tab`, enable `Root Mode` to `ON`
3. Select `OK`

**To enable a complete root, we are going to install 2 (.apk) files. This extension is used when installing applications on an Android device.*

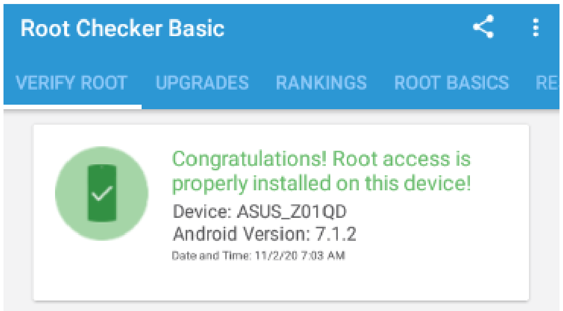
4. Install Root Check APK file on your PC
    a. Download Link: https://apkpure.com/root-checker/com.joeykrim.rootcheck

5. Install SuperSU APK file on your PC
    a. Download Link: https://apkpure.com/supersu/eu.chainfire.supersu

6. Launch the MEmu Emulator
7. Navigate to the toolbar to the right and select the `APK` button



8. Open the 2 apk files that you just downloaded (Root Check &
9. Your applications should now be found on the home screen
10. Open the SuperSU app on the emulator
11. Update the SU Binary by selecting `Continue`
12. Install the SU Binary by selecting `Normal`
13. You will get an "`Installation Success`" message
14. Open the RootChecker app on the emulator and verify the root is complete



Congratulations! Root access is properly installed on this device!

Device: ASUS_Z01QD

Android Version: 7.1.2

Date and Time: 11/2/20 7:03 AM

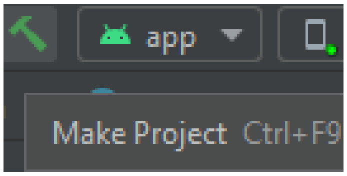
15. Your emulator is now rooted, so any application being ran on the application with permission will have access to root permissions!

3.0 Running the Task Manager Application

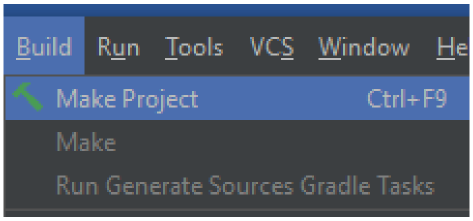
It is important to understand how to build and run your application. When it comes to building and running the application, it can be done in several ways. This section will analyze the different ways that you may build and run the application.

3.1 Building the App

1. It is important to build the application before running it for the first time. To build the application, you find the green hammer icon and click on it.

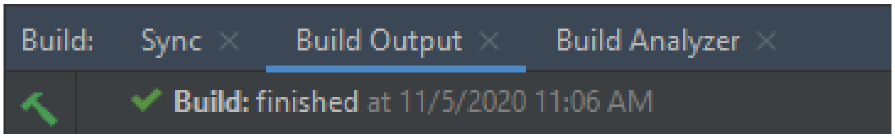


2. You may also build the application from the top toolbar you will select `Build > Make Project.`



3. You can also use the keyboard shortcut: > `CTRL + F9`

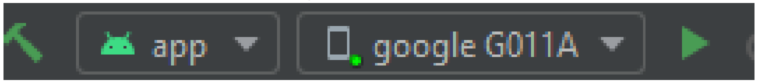
4. After the build, it may warn you of any errors or simply give you a build success message at the bottom.



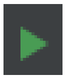
Build: finished at 11/5/2020 11:06 AM

3.2 Running the App

1. Before running the application be sure that the select device is the MEmu Player. The device name should be the one that you selected from step (2.1.2) Change MEmu Device. If your device is not there, check and make sure you have USB Debugging enabled from step (2.1.3) USB Debugging. If USB Debugging is already enabled, try restarted the MEmu Play Emulator.



2. To run the application, you can click the `green play button` next your selected device. When you run the application is builds it as well.



3. Another way to run your application is by finding the `Run` tab on the top toolbar. Select `Run ‘run app’`.

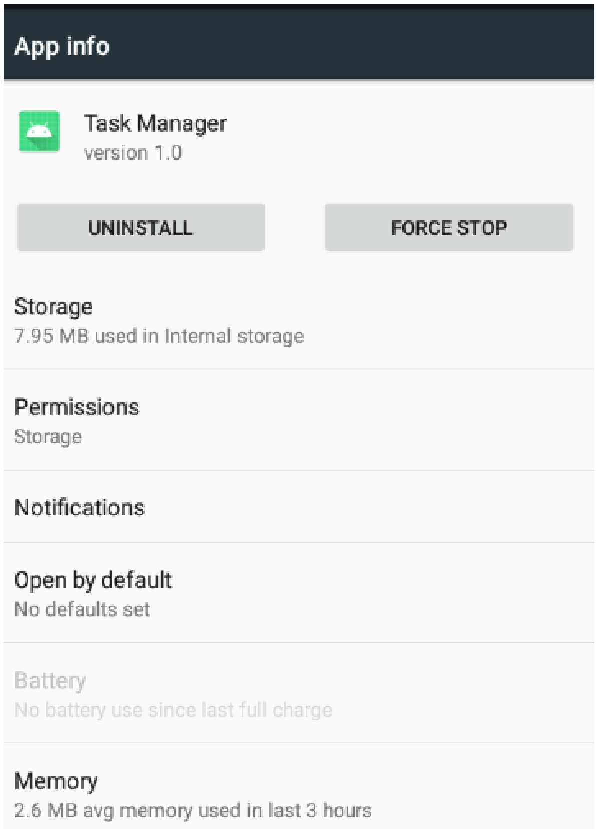
4. You can also use the keyboard shortcut: > `Shift + F10`

4.0 <u>Analyzing Output</u>

There are 3 different tabs of information in the Task Manager Application. The 3 different tabs show 3 different key pieces of information. This section will analyze the different output in each tab and other features.

4.1 App Details Button

The `App Details` button shows you all the available information about the Task Manager Application. This is including if the user would like to know more about the application’s `storage, permissions, notifications, and memory`. This allows the user to also `force stop` or `uninstall` the application.



App info

Task Manager
version 1.0

Storage
7.95 MB used in Internal storage

Permissions
Storage

Notifications

Open by default
No defaults set

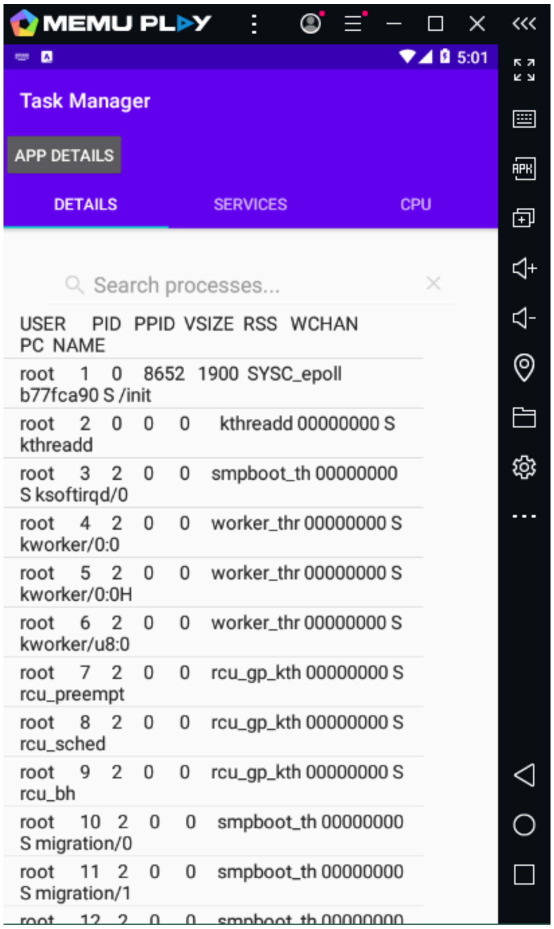
Battery
No battery use since last full charge

Memory
2.6 MB avg memory used in last 3 hours

4.2 Details Tab

- The `Details Tab` shows all processes from all available users. To retrieve the running processes from all the available users, the command "`su -c ps`" is being run.
- The `first line` shows the format of information `USER > PID > PPID > VSIZE > RSS > WCHAN > PC > NAME` [1].
  - `USER` shows the user in which is running that process
  - `PID` shows the process id, it will be in the form of an integer
  - `PPID` shows the parent process PID for that process
  - `VSIZE` shows the virtual memory size
  - `RSS` shows the resident size memory
  - `WCHAN` shows is the address in the kernel where the process is sleeping
  - `PC NAME` shows the name of the process
  - This format will remain the same through the list of all the processes.
- The processes are all shown in a `ListView`. This `ListView` allows for a scrollable list experience where each process is on its own line.

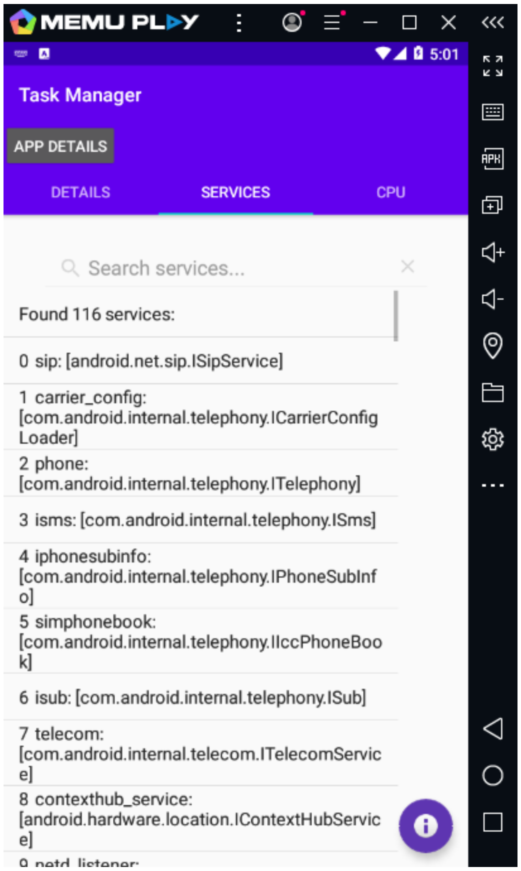
- A `SearchView` is found at the top to filter the different processes if you are looking for a specific one without having to spend time finding it.



```
USER    PID  PPID  VSIZE  RSS   WCHAN
PC  NAME
root     1    0    8652   1900  SYSC_epoll
b77fca90 S /init
root     2    0    0      0     kthreadd 00000000 S
kthreadd
root     3    2    0      0     smpboot_th 00000000
S ksoftirqd/0
root     4    2    0      0     worker_thr 00000000 S
kworker/0:0
root     5    2    0      0     worker_thr 00000000 S
kworker/0:0H
root     6    2    0      0     worker_thr 00000000 S
kworker/u8:0
root     7    2    0      0     rcu_gp_kth 00000000 S
rcu_preempt
root     8    2    0      0     rcu_gp_kth 00000000 S
rcu_sched
root     9    2    0      0     rcu_gp_kth 00000000 S
rcu_bh
root     10   2    0      0     smpboot_th 00000000
S migration/0
root     11   2    0      0     smpboot_th 00000000
S migration/1
root     12   2    0      0     smpboot_th 00000000
```

4.3 Services Tab

- The `Services Tab` shows all available services, as well as the currently running ones. To retrieve all available services the command "`su -c service list`" is being run. To retrieve only the running services, the command "`cmd -l`" is being run.
- The services are presented within a `ListView`. This `ListView` is scrollable and it shows each service name on its own line.
- A `SearchView` can be found at the top to filter out a particular service.
- The button in the bottom right-hand corner when clicked will bring another button to view. ⓘ This button, ⇄ when clicked will show you the list of running services. Click the original button to go back to the whole list of services.



Found 116 services:

0 sip: [android.net.sip.ISipService]

1 carrier_config: [com.android.internal.telephony.ICarrierConfigLoader]

2 phone: [com.android.internal.telephony.ITelephony]

3 isms: [com.android.internal.telephony.ISms]

4 iphonesubinfo: [com.android.internal.telephony.IPhoneSubInfo]

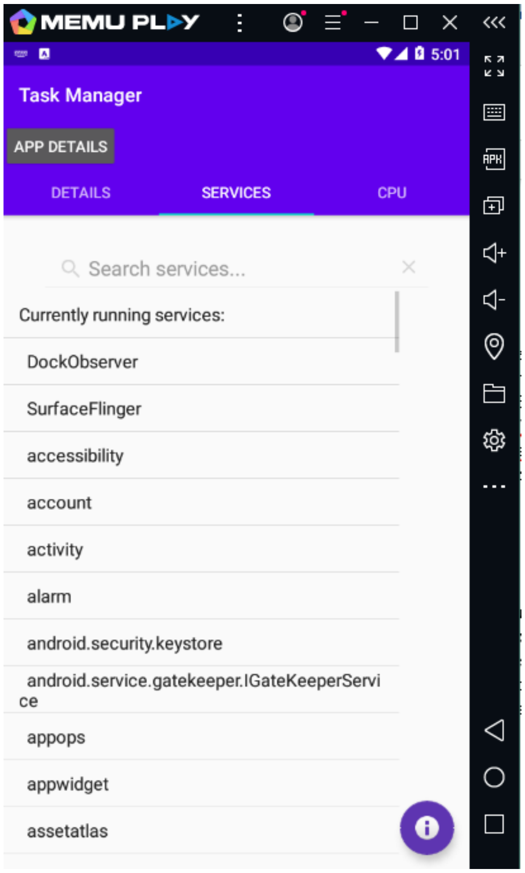
5 simphonebook: [com.android.internal.telephony.IIccPhoneBook]

6 isub: [com.android.internal.telephony.ISub]

7 telecom: [com.android.internal.telecom.ITelecomService]

8 contexthub_service: [android.hardware.location.IContextHubService]

9 netd_listener:



Currently running services:

DockObserver

SurfaceFlinger

accessibility

account

activity

alarm

android.security.keystore

android.service.gatekeeper.IGateKeeperService

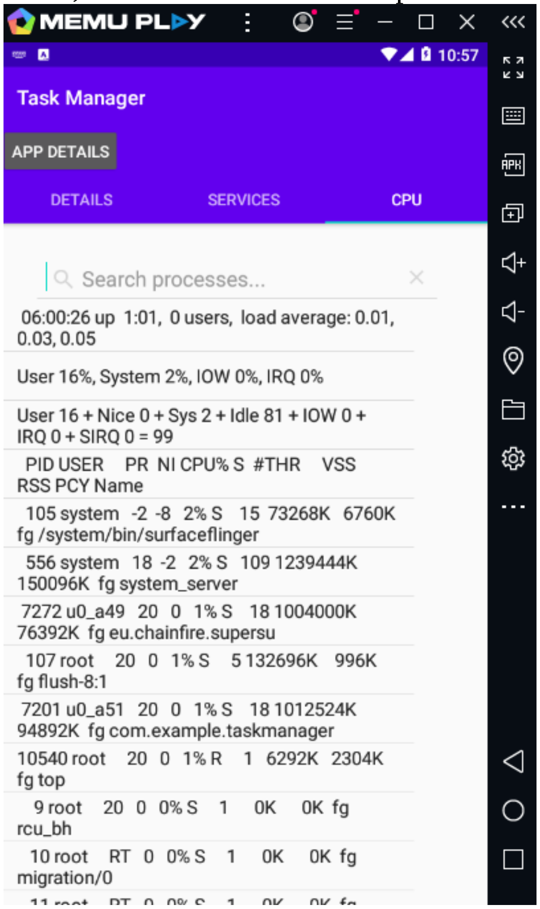
appops

appwidget

assetatlas

4.4 CPU Tab

- The CPU tab shows a lot of useful information about the CPU. This would include uptime, load average, CPU utilization, and the CPU % of each process.



```
 06:00:26 up  1:01,  0 users,  load average: 0.01,
0.03, 0.05
User 16%, System 2%, IOW 0%, IRQ 0%
User 16 + Nice 0 + Sys 2 + Idle 81 + IOW 0 +
IRQ 0 + SIRQ 0 = 99
  PID USER     PR  NI CPU% S  #THR     VSS
RSS PCY Name
  105 system   -2  -8   2% S    15  73268K   6760K
fg /system/bin/surfaceflinger
  556 system   18  -2   2% S   109 1239444K
150096K  fg system_server
 7272 u0_a49   20   0   1% S    18 1004000K
76392K  fg eu.chainfire.supersu
  107 root     20   0   1% S     5 132696K    996K
fg flush-8:1
 7201 u0_a51   20   0   1% S    18 1012524K
94892K  fg com.example.taskmanager
10540 root     20   0   1% R     1   6292K   2304K
fg top
    9 root     20   0   0% S     1      0K      0K  fg
rcu_bh
   10 root     RT   0   0% S     1      0K      0K  fg
migration/0
   11 root     RT   0   0% S     1      0K      0K  fg
```

- The first line shows the `current system time, system up-time`, `number of users`, and `CPU load average`. The information on the first line is separated by a comma (`,`).
    - The first item shows the time on the system. The hours, minutes and seconds are separated by a simple colon (`:`). The second item on the first line shows how long the system has been up. If the system has been up in minutes it will show the number followed by the units, “min”. If it has been up more than an hour it output in the format of (# of hours:# of minutes).
    - The third item shows the number of users on the system. It will say 0 users because we are working in an emulator.
    - The fourth item is the CPU load average. There are 3 values associated with the load average because it shows the average in 3 different time intervals (1 minute, 5 minutes and 15 minutes).

- The second line shows the CPU utilization statuses.
    - The first item (`User %`) shows % of utilization while executing at user level.

    - The second item (`System %`) shows % of utilization while executing at kernel level.
    - The third item (`IO Wait` (`IOW`) %) shows % operating system is waiting for input output.
    - The fourth item (`Interrupt Request` (`IRQ`) %) shows the interrupt signal sent to the processor.

- The third line shows more CPU utilization % statuses. Its line repeats some information while giving us added information.
    - The first item (`User`) shows % of utilization while executing at user level.
    - The second item (`Nice`) shows % of utilization at user level and nice priority.
    - The third item (`Sys`) shows % of utilization while executing at kernel level.
    - The fourth item (`Idle`) shows the % of how the operating system being idle.
    - The fifth item (`IO Wait` (`IOW`)) shows % of operating system waiting for input output.
    - The sixth item (`Interrupt Request` (`IRQ`)) shows the interrupt signal sent to the processor.
    - The seventh item (`Softirqs` (`SIRQ`)) shows that the processor has its own thread and states the number of processors [3].

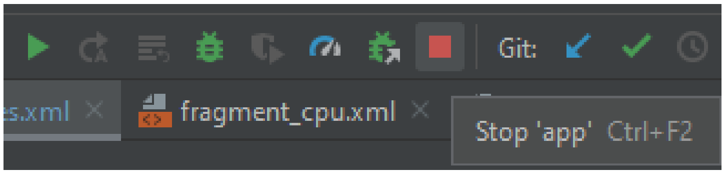
- The fourth line shows the format in which the processes are shown in the lines below. This format contains: `PID > USER > PR > NI > CPU% > S > #THR > VSS > RSS > PCY > Name` [2].
    - `PID` shows the process id
    - `User` shows the user in which is running that process
    - `PR` shows the priority of the process
    - `NI` is the nice value
    - `CPU%` shows how much CPU power the process is using
    - `S` shows process status code
    - `#THR` shows the number of threads
    - `VSS` shows virtual memory
    - `RSS` shows the real memory usage
    - `PCY` shows the policy of the Android memory manager
    - `Name` shows the name of the process
- A `SearchView` can be found at the top of the tab, which grants the ability to search for a specific process.

5.0 <u>Stopping the Task Manager Application</u>

There are several ways that you can stop the Task Manager Application. These different sections will give steps to the different ways to stopping the application and allow yourself to find the easiest way for you.

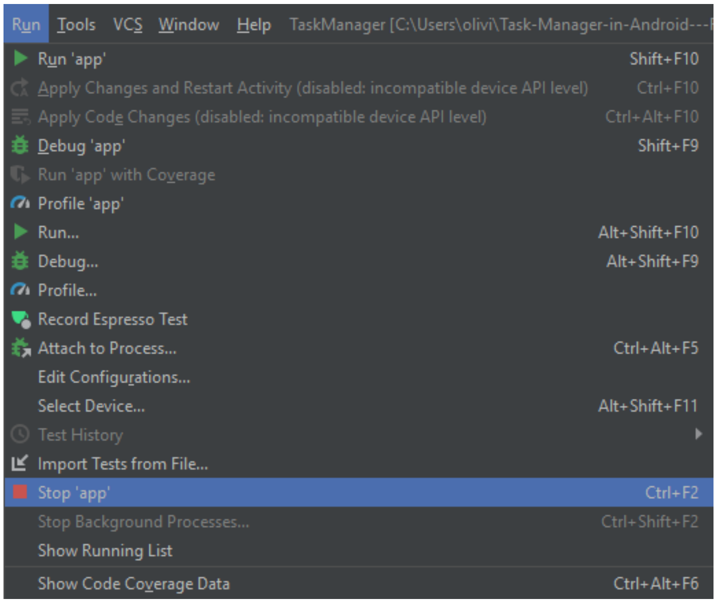
5.1 Stop Button

The stop button can be found at the top in the different tools to run, debug, and profile the application. The stop button is a red square button, and if you click it you can stop the application.



5.2 From Toolbar

You can stop the application from the top toolbar. You can find this option from: `Run > Stop`.

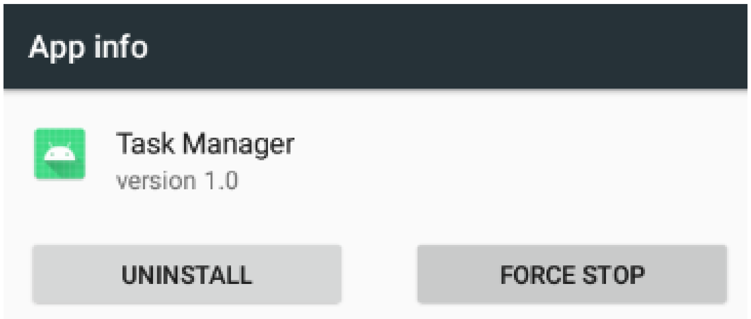

Run Tools VCS Window Help TaskManager [C:\Users\olivi\Task-Manager-in-Android---
Run 'app' Shift+F10
Apply Changes and Restart Activity (disabled: incompatible device API level) Ctrl+F10
Apply Code Changes (disabled: incompatible device API level) Ctrl+Alt+F10
Debug 'app' Shift+F9
Run 'app' with Coverage
Profile 'app'
Run... Alt+Shift+F10
Debug... Alt+Shift+F9
Profile...
Record Espresso Test
Attach to Process... Ctrl+Alt+F5
Edit Configurations...
Select Device... Alt+Shift+F11
Test History
Import Tests from File...
Stop 'app' Ctrl+F2
Stop Background Processes... Ctrl+Shift+F2
Show Running List
Show Code Coverage Data Ctrl+Alt+F6

5.3 Keyboard Shortcut

You may stop the application from a keyboard shortcut defined in Android Studio.
> `CTRL + F2`

5.4 Force Stop

Force stopping the application is not preferred, but it is a way to stop the application. This force stop can be done within the Task Manager application. Click the "`App Details`" `button`. From there you will see a window with the application details. A `Force Stop` button can be found in on the top right. Click this button and verify that you want to force stop and it will give you a warning message. Say OK and it will force stop the running application.



Resources

[1] UNDERSTANDING OUTPUT FIELDS IN PS COMMAND:

This command explains the output fields of the ps command. There were some output fields that were unfamiliar.

[1] https://kb.iu.edu/d/afnv

[2] UNDERSTANDING OUTPUT FIELDS IN TOP COMMAND:

This command explains the output fields within the top command. *There were some output fields that were unfamiliar.*

[2] https://stackoverflow.com/questions/10051152/does-any-one-know-the-top-command-s-results-meaning-in-android

[3] SOFTIRQ OUTPUT FIELD:

This command explains what a SIRQ (Softirq) is and its purpose in the top command output field.

[3] https://0xax.gitbooks.io/linux-insides/content/Interrupts/linux-interrupts-9.html