

# ***Treppn™ Profiler Starter Edition***

## ***User Guide***

***80-N2762-1 J***

***September 12, 2013***

---

**Submit technical questions at:**  
<https://support.cdmatech.com/>

Qualcomm is a trademark of QUALCOMM Incorporated, registered in the United States and other countries. All QUALCOMM Incorporated trademarks are used with permission. Other product and brand names may be trademarks or registered trademarks of their respective owners.

This technical data may be subject to U.S. and international export, re-export, or transfer (“export”) laws. Diversion contrary to U.S. and international law is strictly prohibited.

# Contents

---

<b>1 Introduction.....</b>	<b>5</b>
1.1 Trepn™ Profiler.....	5
1.1.1 Snapdragon MDP.....	5
1.2 Conventions .....	6
1.3 Updates to this version.....	6
1.4 Trepn Profiler download.....	7
<b>2 Getting Started.....</b>	<b>8</b>
2.1 Profiling your application .....	8
2.2 Graph .....	9
2.3 Stats.....	10
<b>3 Using Trepn Profiler .....</b>	<b>11</b>
3.1 Launching the application.....	11
3.2 Trepn Profiler navigation.....	11
3.3 Profile System.....	12
3.4 Profile App.....	13
3.5 Stop Profiling.....	14
3.6 Analyze Run .....	15
3.7 Using the Graph.....	15
3.7.1 Viewing the legend.....	17
3.7.2 Changing the Y-axis .....	18
3.7.3 Graphing Application State .....	19
3.7.4 Zooming and scrolling in the graph.....	20
3.7.5 Understanding the Graph.....	21
3.8 Using the Stats .....	22
3.8.1 Understanding the Stats .....	25
3.9 Preferences.....	26
3.9.1 General Preferences .....	27
3.9.2 Using Overlays .....	28
3.10 Action Overflow .....	29
3.10.1 Help .....	29
3.11 Deleting Profiles .....	30
3.12 Exporting .csv .....	30
3.13 Using status bar controls.....	30
3.13.1 Notification controls .....	31
3.14 Upgrading Trepn Profiler.....	31
3.15 Profiling Battery Power .....	32
3.15.1 Graphing Battery Power .....	33

<b>4 Using Application States .....</b>	<b>34</b>
4.1 Using Android Intents to Identify Application State changes in Trepn Log .....	34
4.1.1 Manipulating Application State in your app.....	34
4.1.2 Code for changing Application State.....	35
4.1.3 Application State example.....	36
<b>5 Using Intents for Automated Testing.....</b>	<b>37</b>
5.1 Automated testing with Trepn Profiler .....	37
5.1.1 Commands for sending Intents from an external .bat script .....	37
5.1.2 Code for sending Intents from another application.....	38
5.1.3 Intents example using an external .bat script.....	40
<b>6 Using Trepn Profiler Efficiently .....</b>	<b>41</b>
6.1 Managing Trepn Profiler's Impact.....	41
<b>A What the Tool Profiles/Captures.....</b>	<b>43</b>
A.1 Data Points .....	43
A.1.1 Power Stats .....	43
A.1.2 CPU Stats .....	43
A.1.3 Network Stats.....	45
A.1.4 Other Stats .....	46
A.2 Normalized CPU load.....	46

## Revision history

Revision	Date	Description
A	Nov 2010	Initial release
B	Jun 2011	Updated for Ver 1.2 and to incorporate Snapdragon™ MDP MSM8660 devices
C	Sep 2011	Updated for Ver 2.0; substantial changes to UI made with this revision
D	Dec 2012	Updated for Ver 3.4
E	Jan 2013	Updated for Ver 3.5
F	Mar 2013	Updated for Ver 3.6
G	Apr 2013	Updated feature set for Ver 3.6
H	Jul 2013	Updated for Ver 3.7
J	Sep 2013	Updated for Ver 4.0

**Note:** There is no Rev. I, O, Q, S, X, or Z per Mil. standards.

# 1 Introduction

---

## 1.1 Trepn™ Profiler

As devices become more advanced and more capable, application developers need to have greater control of the various hardware components. Since battery power is still limited, it is becoming increasingly important to ensure that applications are optimized for minimal power consumption.

The Trepn Profiler included with the Snapdragon™ MDP is a diagnostic tool for providing visibility into the power impact of applications running on top of the OS. It provides the ability to analyze CPU usage, network data, and other items.

The Trepn Profiler features the ability to visualize data in a real-time graph and provide statistical information on the values collected. In addition, overlays allow you to view power and other values dynamically while using the app being profiled. Files can also be exported for offline analysis.

### 1.1.1 Snapdragon MDP

The Snapdragon MDP is Qualcomm's development platform, providing access to cutting-edge hardware in a commercial-like form factor.



For more information on the Snapdragon MDP, see <https://developer.qualcomm.com/mobile-development/development-devices-boards>.

## 1.2 Conventions

This guide uses the convention wherein items that you tap are displayed in **boldface** type. This guide also operates under the assumption that users of the Trepn Profiler/Snapdragon MDP are familiar with basic touchscreen usage, such as scrolling, tapping/double-tapping, and entering text on a touchscreen keyboard.

Shading indicates content that has been added or changed in this revision of the document.

## 1.3 Updates to this version

NOTE: Numerous changes were made in this section.

With Ver 4.0, the Trepn Profiler includes the following upgrades:

- n Full support for the Trepn plug-in for Eclipse. If you are an Android developer, visit the Qualcomm Developer Network to learn more about this exciting software.
- n New Holo-themed interface for a modern, more-refined look
- n New Split Graph View shows multiple graphs on the screen at the same time – Split Graph View works in Portrait view to ensure at least four data points are visible at once. When more than four data points are selected, you can scroll up or down using your finger. You can enable Split Graph View in following ways:
  - .. Go to General Settings and select it under Graph View
  - .. After you have started profiling you can switch views from the Graph View by either:
    - Touching the Menu key and selecting Split View
    - Touching the three Action Overflow dots in the upper right and selecting Split View
- Graph height varies depending on the number of data points selected
- n Added battery power monitoring support in devices that did not have it before, such as the HTC One and new Nexus 7. All Android devices that support this feature should now display battery power.
- n Added the ability to show actual battery power in addition to the delta between the baseline battery power and current battery power. This can be enabled using the Show Deltas option on the General Settings screen.
- n The appearance of the Merged Graph View has been improved for a cleaner look with a new sans serif font.
- n Overlay-related features:
  - .. Added an Overlays button to the main profiling screen, so it is easier for new users to profile with real-time overlays.
  - .. You can now have up to six real-time overlays visible at once on any device
  - .. You can now set the transparency level of a real-time overlay on the Overlay settings page by moving the slider to the right to make an overlay more opaque or left to make it more transparent.
  - .. The overlay title is automatically pre-filled in when the first data point selected. If this does not occur, just clear the current name and try again.

- n Added an Action Overflow button to the Action bar. This contains frequently accessed items such as Stop Profiling, Settings, About, Help and the ability to exit the app. If you are currently profiling, you can also stop profiling or switch views from the Action bar
- n Added a dialog to notify first-time users when they have entered overlay mode and tell them how to get back to the Trepn home screen. If you do not want to see this warning again, touch Don't Show Again.
- n Removed the Are you sure you want to stop profiling? dialog that appeared every time you stopped profiling and replaced it with some new time-saving options. From a single dialog you can now exit without saving, save as .csv, save as .db, or press back to return to profiling.
- n The name of the app being profiled is inserted in the auto-generated filename.
- n A warning is now displayed when you select more than six different data points. Touch OK to continue or touch Don't Show Again. For best performance, touch OK and make sure less than seven data points are checked.
- n The new Snapdragon-branded About Box has a link to our QDevNet developer's forum and a private email where you can submit feature requests and report bugs.
- n Other bug fixes and optimizations

## 1.4 Trepn Profiler download

To download the latest version, visit <http://developer.qualcomm.com/Trepn>.

## 2 Getting Started

---

NOTE: Numerous changes were made in this chapter.

### 2.1 Profiling your application

The Trepn Profiler allows you to profile any Android™ application.



1. Start the Trepn Profiler from the App Launcher or Home screen.
2. Select **Profile App**.
3. Select your application from the list. The Trepn Profiler is now profiling your app. Test any functionality of the app while the Trepn Profiler collects data in the background.
4. Return to the Trepn Profiler and select **Graph** or **Stats** to view the Graph or Stats, respectively. If Overlays have been set up in Preferences, this allows you to return to the App and monitor the Trepn data while using the App.
5. Select **Stop Profiling** to save.

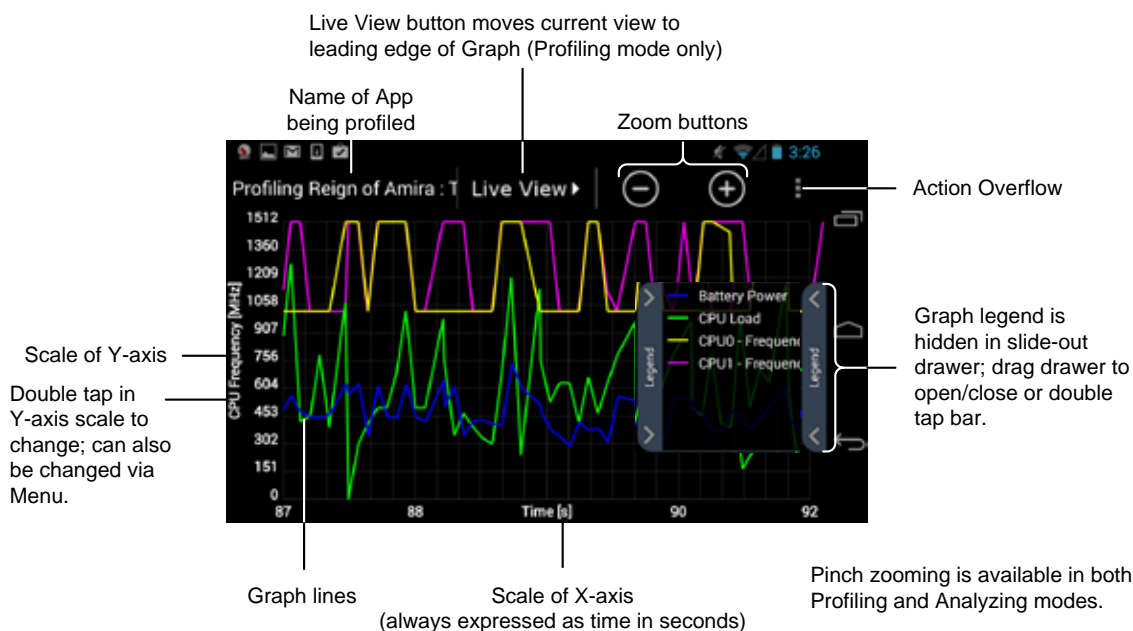
The Trepn Profiler profiles a default set of data points. You can modify the points captured via **Preferences→Data Points**.



**NOTE:** Trepn Profiler automatic Help is disabled by default. Going to Preferences→General Preferences and checking the Enable Help Dialogs option will enable automatic Help. This will display Help screens whenever available. In addition, you can view Help topics where available by tapping Action Overflow and selecting Help. Go to [Help](#) for more information.

## 2.2 Graph

A functional representation of the Graph is given below:



For more information on the Graph, go to [Understanding the Graph](#).

## 2.3 Stats

The Stats screen is given below:



**NOTE:** For the tab bar to appear, Show Per-Application Statistics must be enabled in [General Preferences](#).

For more information on the Stats, go to [Using the Stats](#).

# 3 Using Trepro Profiler

---

NOTE: Numerous changes were made in this chapter.

This chapter provides detailed instructions on the using the Trepro Profiler included with the Snapdragon MDP device.

## 3.1 Launching the application

From the main home screen or the application launcher on the Snapdragon MDP device or other supported device, tap the **Trepro** icon to launch the program.

## 3.2 Trepro Profiler navigation

When starting the program, the Trepro Profiler home screen gives the following options:

- [Profile System](#) – Starts profiling the system (without running a target App).
- [Profile App](#) – Starts profiling a given application.
- [Analyze Run](#) – Loads a saved session to review the data.

The action bar has two buttons to access common option.

- [Preferences](#) – Allows changing general preferences, data points, and overlays.
- [Action Overflow](#) – Gives context-sensitive access to frequently accessed items such as Stop Profiling, Settings, About, Help and the ability to exit the Trepro Profiler. From the Trepro Profiler home screen, Action Overflows has the following options:
  - Settings – Same as tapping the Preferences button
  - About – Gives version number/additional information about the Trepro Profiler and links to important items such as Send feedback for submitting feature requests and reporting bugs and a link to the QDevNet Developer Forum.
  - [Help](#) – Provides context-sensitive Help topics on screens where help is available.
  - Exit

When profiling or analyzing a run, other options appear:

- [Overlays](#) – Allows you to return to the app/home screen and monitor the Trepro data while using the app (not applicable to analyzing a run)
- [Graph](#) – Opens the Graph in Profiling or Analyzing modes. See [Using the Graph](#) for details.
- [Stats](#) – Opens the Stats in Profiling or Analyzing modes. See [Using the Stats](#) for details.

Depending on whether you are profiling or analyzing a run, other options appear:

- n [Stop Profiling](#) – Opens a Save Session dialog and ends the session (in Profiling mode).
- n Close Session – Closes the session (in Analyzing mode).

### 3.3 Profile System

This section describes profiling the system on the Android device.

**NOTE:** In most use cases, you would want to set [Preferences](#) before beginning the profile.

To start profiling:

1. Tap **Profile System** from the main menu screen. If connected to a power supply and Battery Power is selected, this opens a Power Source Warning dialog. It is recommended that you do *not* plug in a power source while profiling Battery Power. The Trepro Profiler will start logging data about the device.
2. Tap **Home** to return to the device home screen.
3. Use the device as a user normally would.
4. Return to the Trepro Profiler to view data logged by opening the Trepro Profiler, where the following options are available:
  - .. Overlays – Allows you to return to the app/home screen and monitor the Trepro data while using the device
  - .. Graph – Opens the Graph
  - .. Stats – Opens the Stats
  - .. Stop Profiling – Opens a Save Session dialog and ends the session

**NOTE:** An easy way to return to Trepro Profiler while profiling is to open the Android notification bar and tap the Trepro Profiler icon.

For each start and stop profiling event, Trepro will print logs to logcat to correlate the start and stop times with any other logcat output.

When Trepro starts profiling, it will output the following message to logcat with the tag “Trepro”:

- n “Trepro has Started Profiling.”

When Trepro stops profiling, it will output:

- n “Trepro has Stopped Profiling.”

## 3.4 Profile App

This section describes profiling a specific app. In most use cases, you would want to set [Preferences](#) before beginning the profile.

To begin profiling an app:

1. Tap **Profile App** from the Trepn main menu screen.
2. In the Select an application to profile dialog, scroll to the desired application and tap to select it. If connected to a power supply and Battery Power is selected, this opens a Power Source Warning dialog. It is recommended that you do *not* plug in a power source while profiling Battery Power.
3. Trepn Profiler may collect baseline data for a period specified by the Baseline Interval in General Preferences and then launch the application. Once the app launches, Trepn Profiler begins capturing data.
4. Use the application as a user normally would. Note that if overlays have been selected, a Graph, Voltmeter, or Bar Chart will appear in the desired portion of the screen while the application is running. This can be helpful in looking for power spikes during normal usage of the application. Individual overlays can be hidden/shown during profiling by selecting/unselecting the corresponding checkbox. The overlays can also be positioned on the screen by touching and dragging them individually.

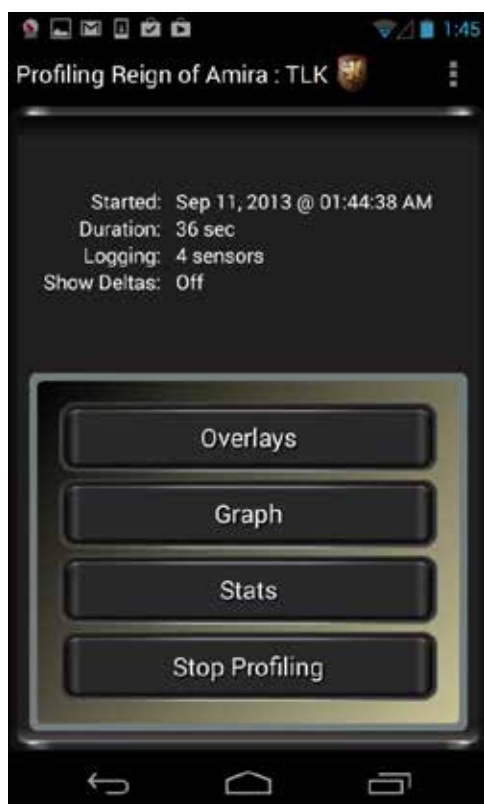


If a specific feature or portion of the application is to be tested, see [Using Android Intents to Identify Application State changes in Trepn Log](#).

5. Return to the Trepn Profiler to view the impact of the application, where the following options are available:
  - Overlays – Allows you to return to the app/home screen and monitor the Trepn data while using the app
  - Graph – Opens the Graph
  - Stats – Opens the Stats
  - Stop Profiling – Opens a Save Session dialog and ends the session
6. If necessary, return to your application in a similar fashion.

## 3.5 Stop Profiling

This section describes stopping the profile in the Trepn Profiler.



**NOTE:** It may be necessary to return to the profiling screen by tapping **Back**.

To stop profiling:

1. Tap **Stop Profiling** to bring up the Save Session dialog.
2. Select one of the following options:
  - Don't save – Stops the session without saving.
  - Save as .csv – Saves the session as a .csv file on the micro-SD card/internal storage. The .csv *cannot* be opened in the Trepn Profiler and is intended for offline analysis on a PC.
  - Save as .db – Saves the session as a database .db on the micro-SD card/internal storage.

Sessions saved as database .db files can be exported to .csv files as indicated in

[Exporting .csv](#).

## 3.6 Analyze Run

This section describes loading a saved session in the Trepn Profiler in Analyzing mode via Analyze Run.

To open a saved session:

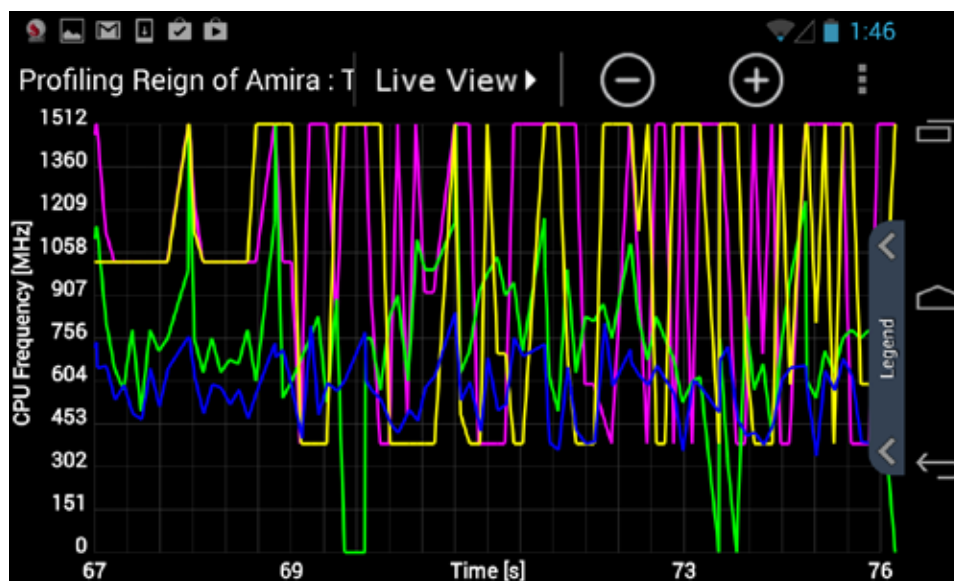
1. Tap **Analyze Run** from the Trepn Profiler main menu screen.
2. In the Select run to analyze dialog scroll to the desired session and tap the checkbox or name of the session to select it.
3. Press **Analyze**, and the session is loaded in the Trepn Profiler in Analyzing mode.

## 3.7 Using the Graph

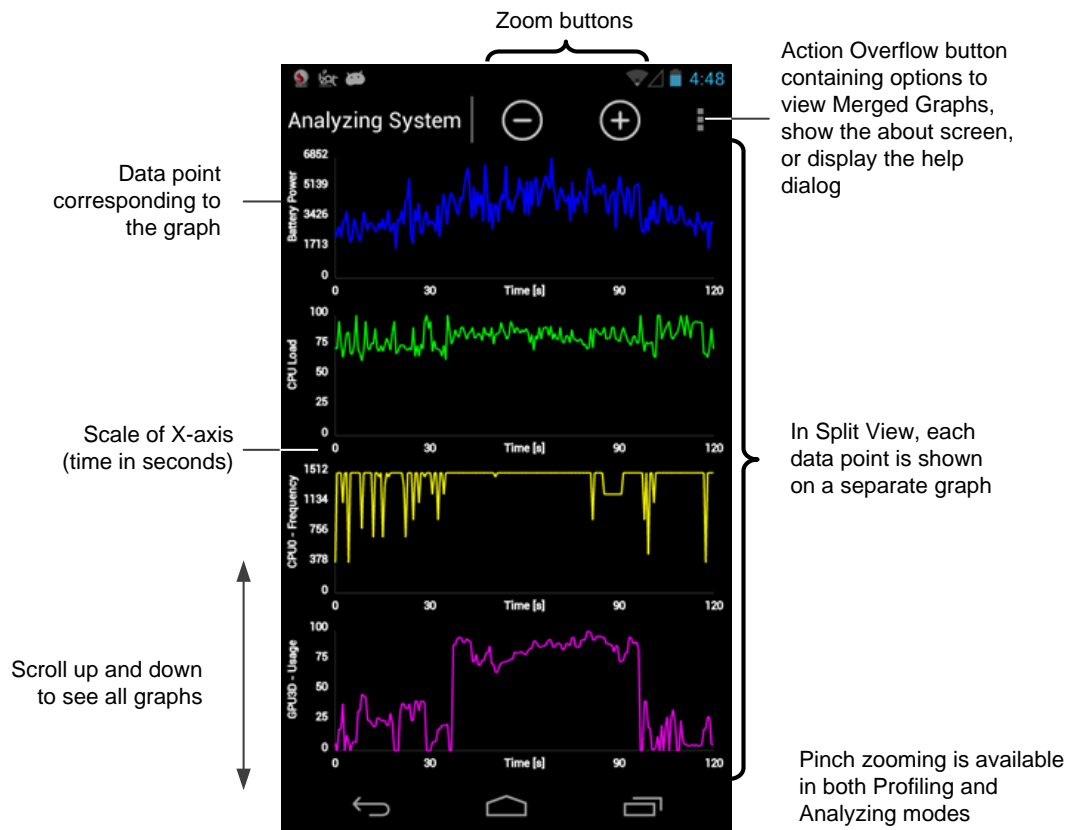
To view the graph while profiling in Profiling mode or with a run loaded in Analyzing mode, tap **Graph**.

The graph provides a visual depiction of the various data points being measured and other items being profiled. The Trepn Profiler graph is given in two views:

- n Merged View – All data points are plotted on the same graph. In this view, the legend is given to define the data points by color. However, the legend is hidden in a in a slide-out drawer. See [Viewing the legend](#) for details. The Merged View is given below:



- n Split View – Each data line appears in a separate graph for easier analysis. If not all data point graphs fit on the screen, you can scroll up or down to see all graphs. The Split View is given below:

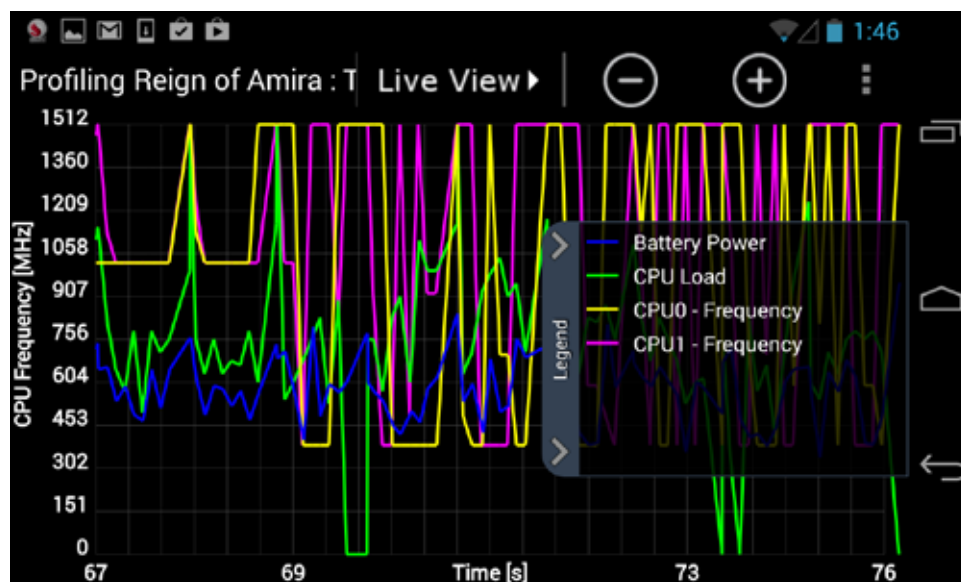


You can toggle between Merged and Split View from the Action Overflow button.

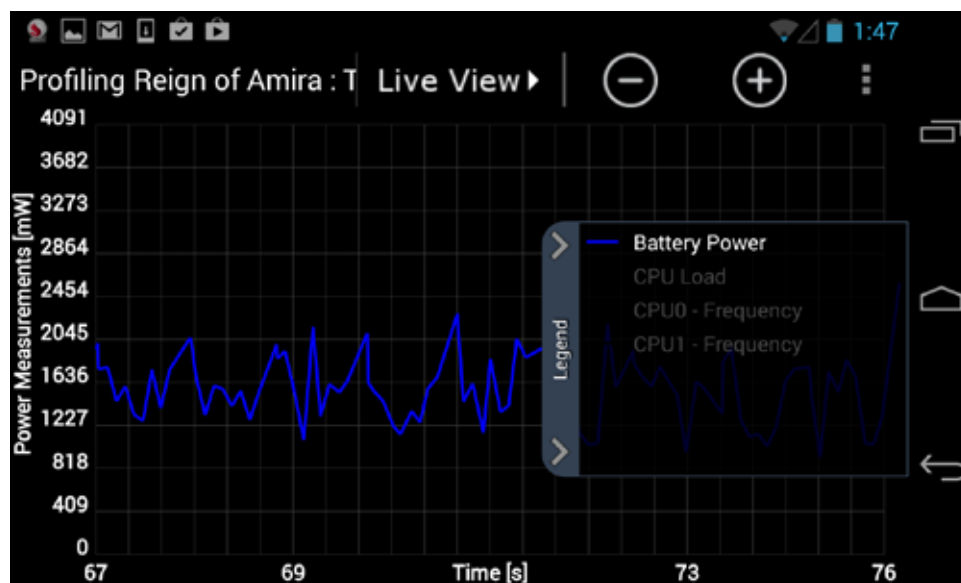


### 3.7.1 Viewing the legend

When first entering the graph, the legend is hidden in a slide-out drawer. To open and close the legend, drag the gray Legend bar on the touchscreen (you can also double-tap the Legend bar to open/close the legend):



When the legend is open you can show or hide graph lines by tapping on the item name. Tapping the item name grays out the item name in the legend and temporarily removes that line from the graph:



Tapping the item name in the legend again shows the hidden line.

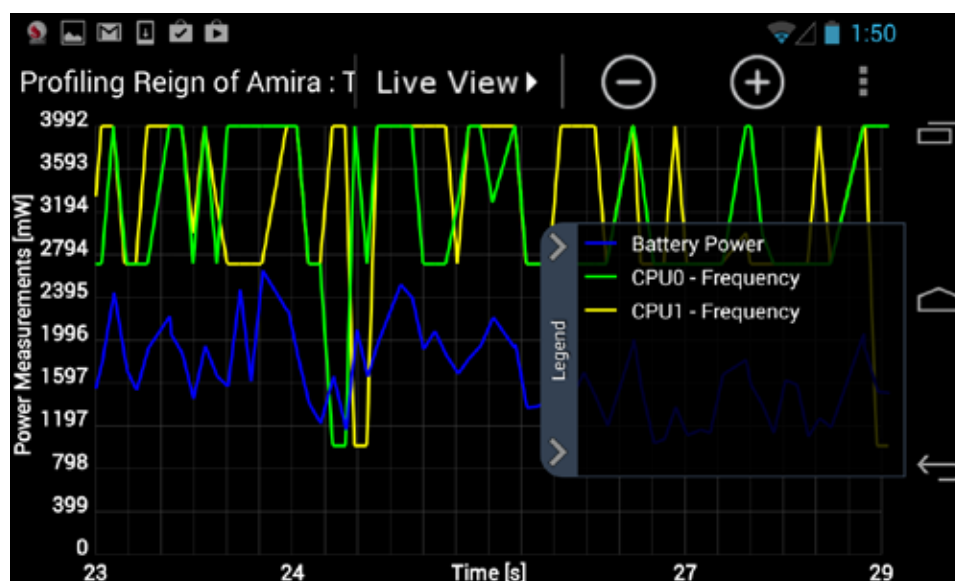
If there are too many items for the legend to display them all at the same time, it may be necessary to scroll up or down to view the other items.

**NOTE:** As the number of data points increases, the sampling rate decreases. Suggested usage is six or fewer data points at one time.

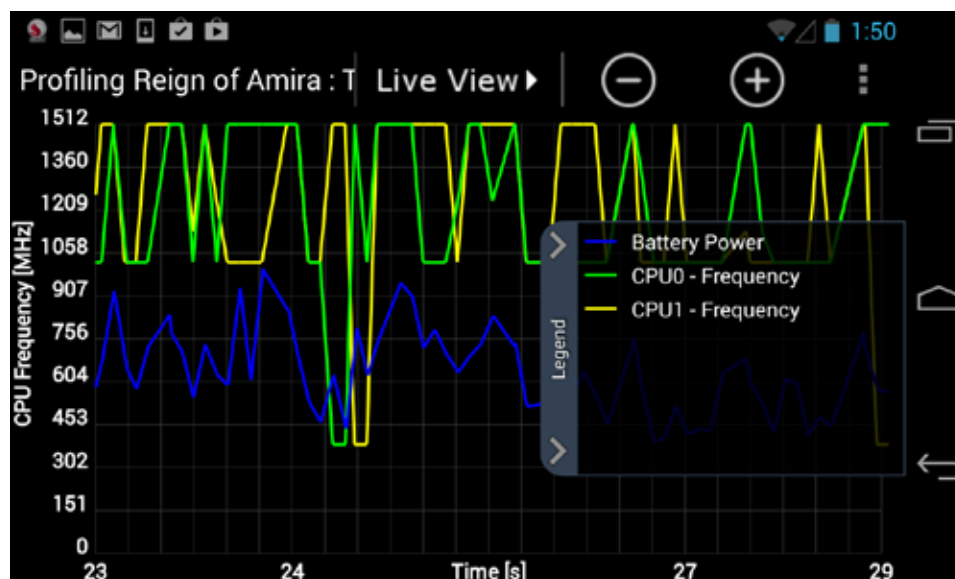
### 3.7.2 Changing the Y-axis

The Trepn Profiler can profile power, CPU usage, and other items at the same time. However, the graph Y-axis only displays one scale at a time. Therefore, it is often necessary to switch the Y-axis back and forth to monitor the values of different items.

In the following example, the Y-axis is displaying the scale for Battery Power (expressed in mW). It does not correspond to the values of CPU0 Frequency or CPU1 Frequency (green and yellow lines, expressed in MHz). To view the correct scale for these data points, you need to change the scale on the Y-axis.



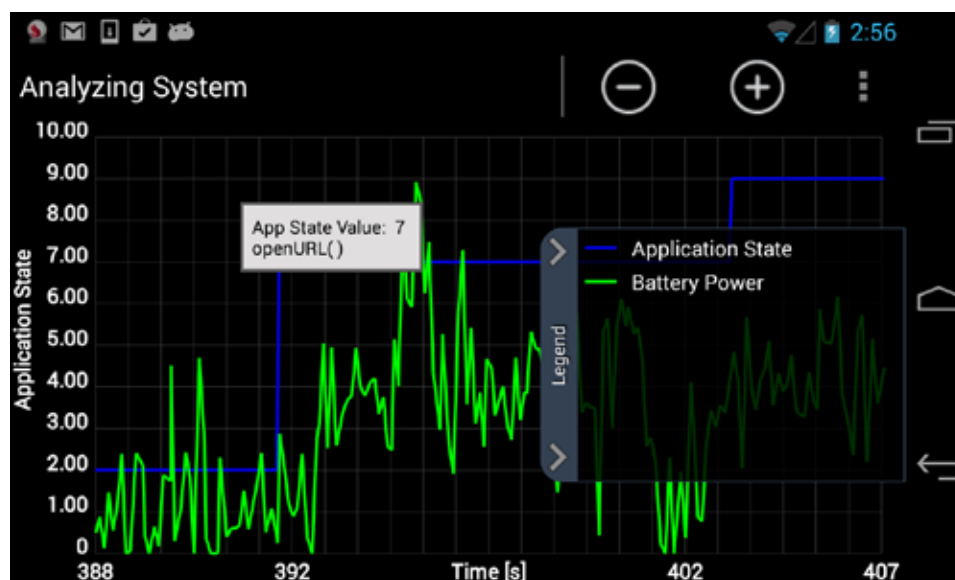
Change the Y-axis by double-tapping on the Y-axis scale. This will cycle through the scales for all data points in the run. The graph displays with the Y-axis changed.



**NOTE:** In this view, the scale on the Y-axis does not correspond to Battery Power (blue line). To view the proper scale for this, the scale on the Y-axis needs to be changed back by double tapping as above.

### 3.7.3 Graphing Application State

If the Application State data point has been enabled, it displays as a graph line in Profiling and Analyzing modes. Tapping this line triggers a pop-up that shows more detailed information about the point that was tapped.



See [Using Application States](#) for more details.

### 3.7.4 Zooming and scrolling in the graph

The Trepn Profiler graph uses pinch zooming and zoom buttons to zoom in and out in both Profiling and Analyzing modes. A pinch open gesture (apart with the fingers) zooms in, while a pinch close gesture (together with the fingers) zooms out. The zoom in/out occurs centered on the points on the touchscreen where the pinch occurred. Pressing the zoom-in (plus) button or zoom-out (minus) button in the title bar will also zoom in or out of the current view.

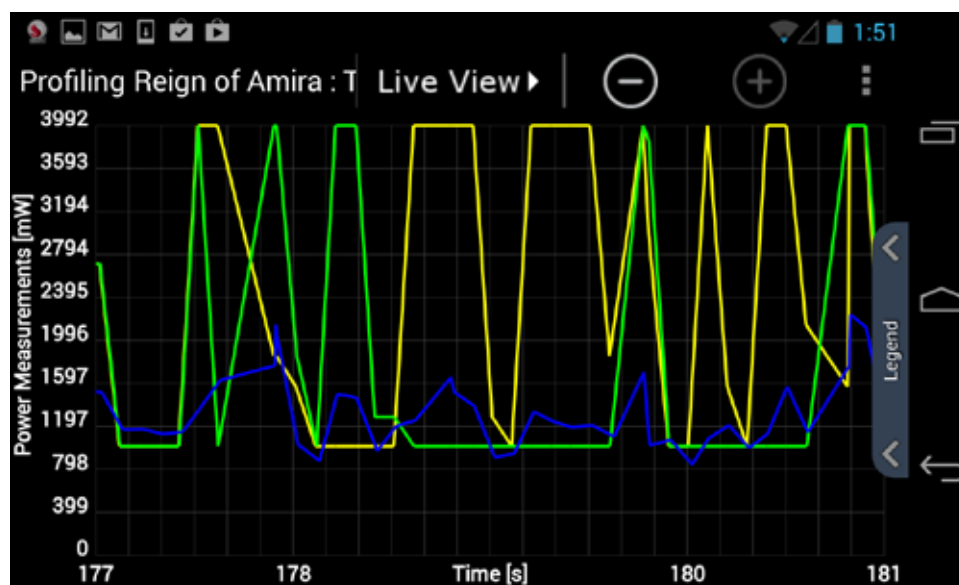
Scrolling is done by sliding a finger on the Graph left or right. Additionally, doing a quick swipe gesture left or right on the Graph will move the view one full page in that direction.

Both zooming and scrolling can have the effect of a soft pause on the Graph in Profiling mode. For example:

- When a graph is being generated, zooming to the point where the actively updating portion of the Graph *is no longer* displayed has the effect of pausing automatic scrolling. This allows the Graph to be viewed in a static mode, while the profile continues to run offscreen.
- Likewise, scrolling to the point where the actively updating portion of the Graph *is no longer* displayed also has the effect of pausing automatic scrolling.

Zooming out or scrolling to the point where the actively updating portion of the Graph *is* displayed has the effect of unpausing. This allows the Graph to be viewed in a dynamic mode where you can view the graph being updated in real-time.

Whenever the Graph is paused, a **Live View** button displays in the title bar. Pressing **Live View** will move the view to the leading edge of the graph and restart automatic scrolling.



**NOTE:** In this view, the graph has been zoomed in to view the data points in higher detail. The X-axis has updated to show the new time scale.

### 3.7.5 Understanding the Graph

The Trepn Profiler Graph gives a graphic visualization of the data collected during the session runtime. All data points selected in Data Point Preferences are displayed in this graph view.

In the Graph, pinch zooming and the zoom buttons can be used in both Profiling and Analyzing modes. Further details of the Graph are:

- n Name of application – Appears after “Profiling:” or “Analyzing:” in Profiling or Analyzing modes, respectively.
- n X-axis – Always labeled Time with units in seconds.
- n Y-axis –The label is that of the first data point added to the graph.

## 3.8 Using the Stats

To view the usage statistics in Profiling and Analyzing mode, tap **Stats**.

**NOTE:** Depending on where you are in the Trepn Profiler, you may need to return to a higher level by tapping **Back**.

The Stats provides tables of the various data points selected.

Stats are only displayed for Battery Power Stats, CPU, Network, or Other data points being logged. There are no total values for CPU Frequency and Load data points since the values are expressed as average percentages and total consumption is not meaningful. Application State sections are included in Stats only if the profiled application utilizes the state change intent of the Application State data point being polled. See [Using Android Intents to Identify Application State changes in Trepn Log](#) for more details.

If Show Per-Application Statistics is enabled in [General Preferences](#), Stats will include a tab bar:

- n System gives stats on items selected via Preferences.
- n App List gives CPU stats on all apps running on the device. Tapping on the name of an app in this tab opens a new tab showing CPU and Memory Stats for that selected app.
- n <Name of App> (Reign of Amira in this example) gives CPU and Memory Stats on the profiled app.

**NOTE:** If Show Per-Application Statistics is *not* enabled before profiling starts, the System stats screen will be the only screen available (no tab bar).

Profiling Reign of Amira : TLK		
Overall	Average	Total
Power Measurements		
Battery Power	2,101.64 [mW]	29.54 [mWh]
CPU Load	Average	Total
CPU Load	45.99 [%]	N/A
CPU Frequency	Average	Total
CPU0 - Frequency	1,138.65 [MHz]	N/A
CPU1 - Frequency	1,187.72 [MHz]	N/A

System tab gives stats on items selected via Preferences

Profiling Reign of Amira : TLK	
Application	CPU [%]
Trepn Profiler	18.5
Reign of Amira : TLK	7.8
Settings	1.2
System UI	0.6
Launcher	0.3
Google+	0.2
Google Keyboard	0.2
Google Partner Setup	0.1
Google Account Manager	0.1
Search Applications Provider	0.1

App List tab gives CPU stats on all apps running on device

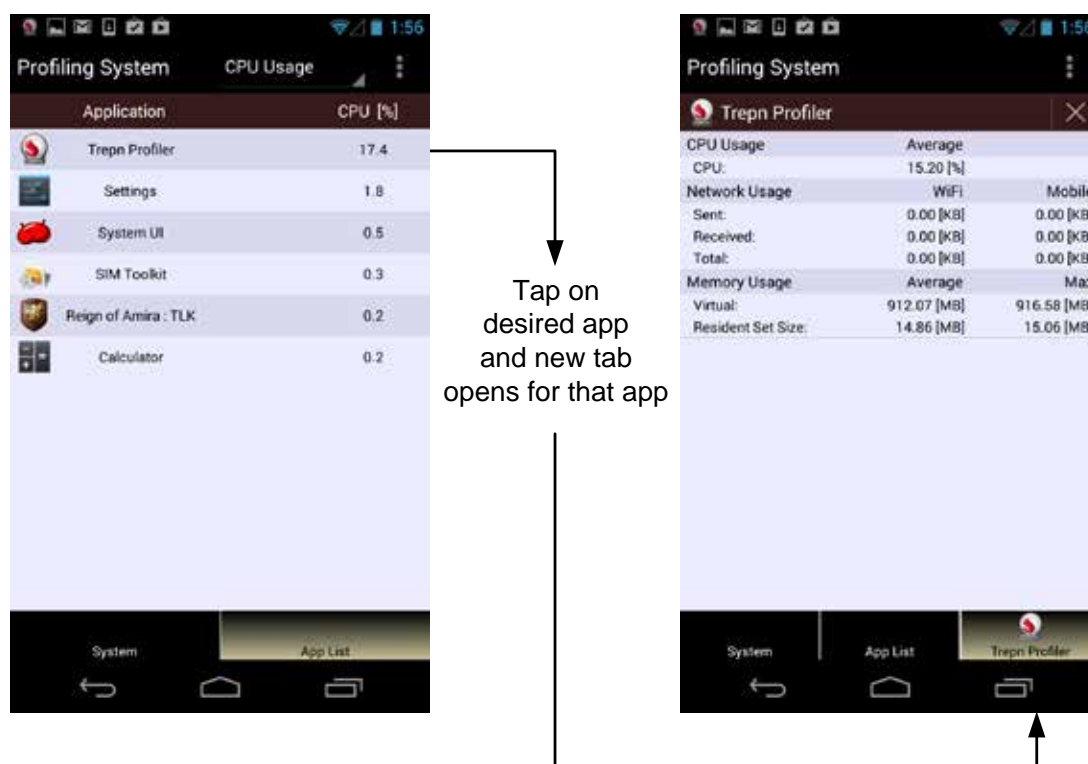
Profiling Reign of Amira : TLK		
CPU Usage	Average	
CPU	7.49 [%]	
Network Usage	WiFi	Mobile
Sent:	0.00 [KB]	0.00 [KB]
Received:	0.00 [KB]	0.00 [KB]
Total:	0.00 [KB]	0.00 [KB]
Memory Usage	Average	Max
Virtual:	1,199.15 [MB]	1,208.26 [MB]
Resident Set Size:	80.81 [MB]	80.87 [MB]

Last tab (Reign of Amira in this example) gives CPU and Memory Stats on profiled app

When profiling a certain app, a tab is shown giving CPU and Memory Stats for that app. In addition, the Trepn Profiler has the ability to give these application stats for all apps currently running on the device, and this can be done while profiling the system (Profile System) or an app (Profile App).

To view CPU and Memory Stats for individual apps:

1. From the main Trepn Profiler screen, tap **Profile System**.
2. Tap **Stats**. The Stats screen opens with the System tab highlighted.
3. Tap the **App List** tab. The App List displays showing a list of all apps currently running on the device.
4. Tap the desired app and a new tab immediately opens for that app:

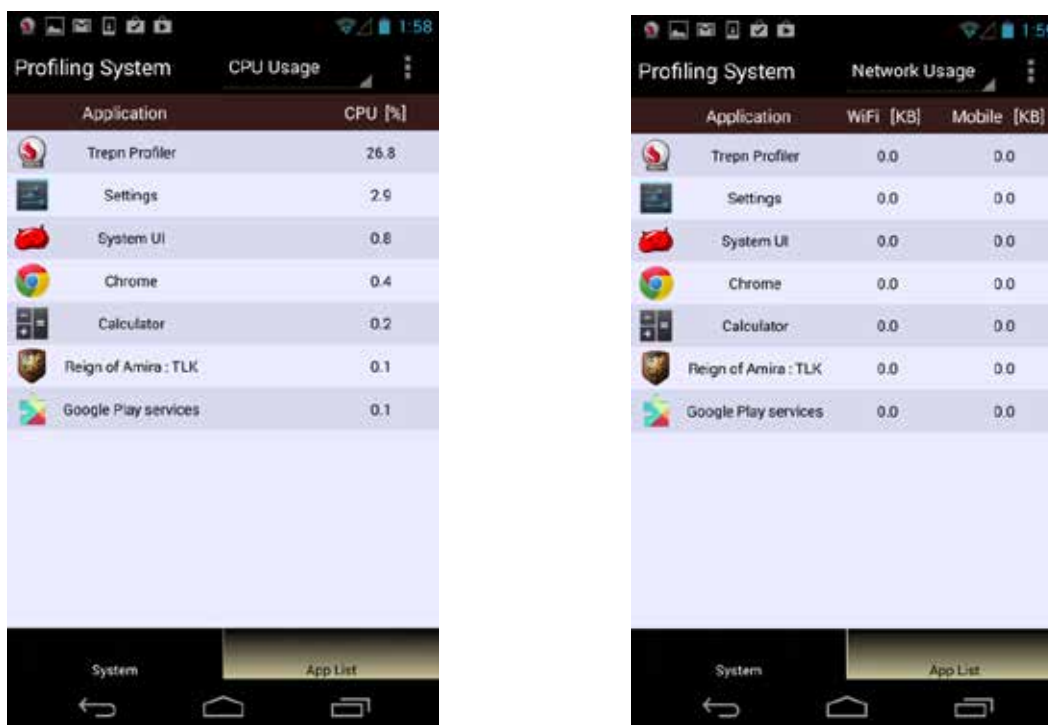


5. Repeat Steps 3 and 4 for any other desired apps.

In this example, Profile System was used. Instructions for Profile App would be similar; however, the target app would already have a tab open.

**NOTE:** CPU and Memory Stats are only available if Show Per-Application Statistics is enabled in [General Preferences](#).

If the device supports per-application network statistics, the App List tab will allow you to switch between CPU and network usage:



- n The **CPU Usage** option shows the CPU usage of all applications running on the device.
- n The **Network Usage** option shows Wi-Fi and mobile data usage of all applications running on the device.

**NOTE:** The `/proc/uid_stat` directory must be present on the device to support per-application network statistics.

The per-application network statistic is a cumulative value over the duration of the run. Alternatively, per-application statistics with timestamps can be accessed when the run is exported to a .csv file.



### 3.8.1 Understanding the Stats

The Trepro Profiler Stats displays computed values for the Power and CPU usage data points included in the session. The Battery Power sensor includes both an average value (mW) and total consumption (mWh), while the CPU Load points include only an average (%) because total consumption is not meaningful.

The sections of the statistics view include:

- n Overall – Statistics from profiled application launch to the current runtime of the session in Profiling mode or to the end of the session in Analyzing mode.
- n Application State – Statistics for any developer-assigned Application State found during the run of the session. If the profiled application utilizes the Application State Intent, and the Application State data point is being polled, then statistics will be calculated for each state change. Application State sections will not be included in Stats if the Application State data point is not being polled. For more information on Android Intents and Application States, see [Using Android Intents to Identify Application State changes in Trepro Log](#).

## 3.9 Preferences

To access Preferences, from the Trepro Profiler main menu, tap the **Preferences** action bar button in the upper-right part of the screen or tap the **Action Overflow** action bar button and select Settings.



The Trepro Profiler Preferences screen has the following tabs:

- n General – Settings for enabling help dialogs, storage mode, graph view, and other options. Preferences can be saved and loaded from this tab.
- n Data Points – For complete details on Data Points, see [Appendix A](#).
- n Overlays – For details on using overlays, see [Using Overlays](#).

### 3.9.1 General Preferences

The Trepn Profiler preferences General tab includes the following settings:

- n Enable Help Dialogs – Enables/disables automatic Help dialog screens.
  - n Storage Mode – Sets where Trepn Profiler saves data during profiling.
    - .. Database – Sessions are stored as database .db files on the micro-SD card/internal memory in the /trepn/ directory. If no storage is available, Trepn can still profile, but no data is saved.
    - .. Main Memory – Sessions are stored in RAM. There is less system overhead using this option but not all data is guaranteed to be saved. For a typical session, approximately the last hour of data will be saved, but this can vary depending on the amount of data profiled in the session. Sessions can be saved when profiling is stopped as either a database .db file or a .csv file on the micro-SD card/internal memory. A micro-SD card/available space on internal storage does not have to be present in the device to profile using this option but needs to be present to save the data.
    - .. None – Data is not saved. This option significantly reduces Trepn overhead during profiling.
  - n Graph View:
    - .. Merged – All data lines appear on the same graph.
    - .. Split – Each data line appears in a separate graph.
  - n Baselining Interval – Changes the Baselining Interval. The Baselining Interval is expressed in seconds and is the amount of time in seconds that Trepn Profiler will collect baseline data before the start of profiling. The baseline timer will only run if a Power Stat (Battery Power) is selected in the Data Points preferences. This improves how accurately Trepn can estimate the actual power usage, if available, of a profiled application. A larger baselining interval provides a more accurate estimate.
  - n Show Deltas – Works in conjunction with the Baselining Interval to estimate the power usage of a profiled app by taking the Trepn Profiler/Android OS overhead into account:
    - .. Turning Show Deltas off shows the Battery Power as actual power usage of the battery
    - .. Turning Show Deltas on shows Battery Power as estimated power usage of the profiled application (by removing an estimate of the power drain of the Android OS and the Trepn Profiler).
- NOTE:** Previous versions of Trepn Profiler Starter Edition only showed deltas and never showed the battery power as the actual power usage of the battery. As a result, effectively, Show Deltas functionality was always checked on for previous versions of the Trepn Profiler even though it was not visible in General Preferences.
- n Show Frequency – When there is a Graph overlay with a single frequency point is selected, this option enables/disables showing the current frequency on the overlay.
  - n Show Per-Application Statistics – Collects data for all applications that run during the profiling session. The statistics are shown on the **Stats→App List** tab in Profiling and Analyzing modes. Turning this option off will reduce Trepn overhead while profiling.

- n Acquire Wakelock while Profiling – Prevents the CPU from sleeping while Trepro is profiling. This option may influence the CPU behavior and may not be representative of the normal use case under test. Use it only when appropriate for your test case.
- n Save Preferences – Saves the selected General, Data Points, and Overlays preferences as a .pref preferences file.
- n Load Preferences – Loads a saved preferences .pref file.
- n Reset Preferences – Restores all Trepro settings to default values.

## 3.9.2 Using Overlays

Overlays add small graphs to the touchscreen, letting you see broad readings of data points as you use the app. Overlays can be helpful in looking for power spikes and Application State changes during normal usage of the application.

**NOTE:** There are three default overlays, CPU 0 Frequency, CPU 1 Frequency, and a CPU 0 and CPU 1 comparison Bar Chart. These default overlays can be modified as desired or removed entirely. Reset Preferences returns them to their original settings. In addition, the default Overlay settings enable an Overlay Mode dialog. You can disable this by selecting, Don't Show Again.

To add a new overlay or edit an existing one:

1. Tap **Preferences**→**Overlays**
2. Tap one of the overlays on the screen to bring up the overlay chooser dialog for that overlay.
3. Select the desired settings for that overlay:
  - Type – Type of the overlay:
    - None – Do not show an overlay in this position.
    - Graph – Allows up to two data points to be displayed.
    - Voltmeter – Allows only one data point to be displayed.
    - Bar Chart – Allows up to two data points to be displayed.
  - Title – Trepro automatically pre-fills in the overlay title when the first data point selected. However, it may be necessary to just clear the current name and reselect. Alternatively, you can enter your own name for the overlay.
  - Transparency – Sets the transparency level of the overlay. Move the slider to the right to make an overlay more opaque or to the left to make it more transparent. Note that the transparency level set last is applied to all overlays in a session.
  - Data Points – Select the data point(s) to display in the overlay. For Graph and Bar Chart overlays, a colored box appears to the right of the data point when selecting. Tap this box to select the color for the line or bar chart of that data point.
4. Tap **OK** to go back to the screen showing all the overlays.
5. The preview of the overlay will change to match the selected settings.
6. To position the overlay on the screen, touch the overlay and drag it to the desired location. This position is where the overlay will be displayed while profiling.

The overlay settings can be modified when Trepn is profiling. While Trepn is profiling and an overlay is visible, double-tap the overlay. A preferences dialog is displayed that allows changes to each of the items described in step 3 above. The choice of data points is limited to the data points that were enabled before profiling began.

The overlays can also be repositioned on the screen while Trepn is profiling by touching the overlay and dragging it to the desired location.

## 3.10 Action Overflow

The Action Overflow button appears as three vertical dots at the top right of the action bar and contains frequently accessed items, depending on the context within the Trepn Profiler:

- n Settings – Accesses the Preferences (similar to the Preferences button)
- n About – Give information about the Trepn Profiler version and access to the following:
  - .. License agreement
  - .. Send feedback – Links to a private email where you can submit feature requests and report bugs
  - .. QDevNet – Links to the QDevNet developer’s forum
  - .. Device Info
- n Help – Accesses context sensitive Help where available
- n Exit – Returns to the Android home screen/open App
- n When Profiling, gives the option to Stop the profile
- n When viewing the Graph, gives the option to toggle the view between Merged and Split views

### 3.10.1 Help

To access Help where available, tap the **Action Overflow** button and select Help. The Trepn Profiler automatic Help is disabled by default. When enabled, Trepn Help presents context-sensitive Help topics on all screens where help is available. When automatic Help is disabled, Help screens present only one option, **OK** (tapping **OK** closes the Help window and leaves Help disabled).

To enable automatic Help, tap **Preferences**→**General**→**Help Dialogs** to select the checkbox.

When automatic Help is enabled, Help screens present two options, **OK** and **Disable Help**.

- n Tapping **OK** closes the Help window and leaves automatic Help enabled.
- n Tapping **Disable Help** closes the Help window and turns off the automatic Help.

**NOTE:** Tapping **Back** has the same effect as tapping OK on all Help screens (closes the window without changing the status).

## 3.11 Deleting Profiles

To delete unwanted Profiles:

1. Tap **Analyze Run** from the Trepn Profiler main menu screen.
2. In the Select run to analyze dialog scroll to the desired session(s) and tap the checkboxes/names of the sessions to select them.
3. Press **Delete**.
4. In the confirmation dialog, select **OK** to proceed with the deletion. There is no undo.

In addition, you can delete a profile that you are viewing (in Analyzing mode), by tapping the **Trashcan** icon in the action bar and tapping **Yes** to confirm in the Delete? dialog.

## 3.12 Exporting .csv

Trepn Profiler provides the option to export saved data as a .csv file directly from the UI in Analyzing mode.

To export a run to a .csv file:

1. From the Trepn Profiler main menu, tap **Analyze Run**.
2. In the Select run to analyze dialog, scroll to the name of the desired session, select it, and press **Analyze** to open the session. The session is loaded in the Trepn Profiler in Analyzing mode.
3. Tap the **Save** icon button in the upper-right corner of the action bar.
4. In the Export to .csv dialog, choose the default name or tap into field and enter a file name using the touchscreen keyboard. Press **Export** to save the run as a .csv file in the /trepn/ directory on the micro-SD card/internal storage.

**NOTE:** It is not necessary to enter the .csv file extension when choosing the file name.

## 3.13 Using status bar controls

When Trepn Profiler is launched, a Trepn Profiler icon is added to the notification area of the Android status bar. The icon will stay in the status bar while the application is running, even if it is pushed to the background.

This feature allows for easier access to the Trepn Profiler from other Android screens or applications. By opening the notification drawer and tapping on the Trepn Profiler icon, Trepn Profiler is brought to the foreground.

If Trepn Profiler is not profiling and it is exited by pressing the Android **Back** button, the application will be removed from the status bar. To navigate away from Trepn Profiler without the notification icon being removed, press the Android **Home** button instead.

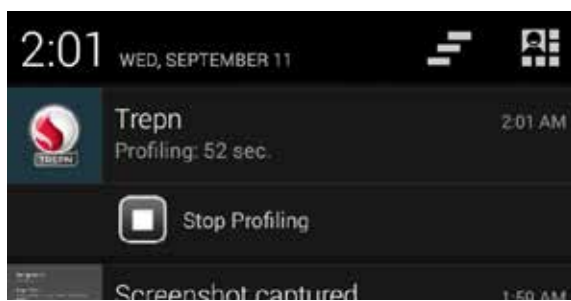
### 3.13.1 Notification controls

**NOTE:** Notification controls are only supported on Android 4.1 (Jelly Bean) or higher.

On supported Android versions, an additional button is included below the main Trepn Profiler notification when the drawer is open. If Trepn Profiler is idle, a **Profile System** button appears. If the application is currently profiling, a **Stop Profiling** button appears.

- n Tapping **Profile System** starts a profiling session in the background using the current preferences.
- n Tapping **Stop Profiling** opens Trepn Profiler and triggers the Save Session confirmation dialog.

If the **Profile System** or **Stop Profiling** button is not displayed, it may be necessary to expand the Trepn Profiler notification. Place two fingers on the Trepn Profiler notification item and spread the fingers apart in a pinch open gesture.



### 3.14 Upgrading Trepn Profiler

Upgrading the Trepn Profiler requires the following software:

- n Java Development Kit (JDK)
- n Android™ SDK

Most users of Trepn will already have this installed.

To install the new version of Trepn Profiler:

1. Download the latest version at <http://developer.qualcomm.com/Trepn> and note the location of the apk file, e.g., C:\trepn\_x.x.apk.
2. Open a command window.
3. Make sure that the Snapdragon MDP or other supported device is connected to the computer via USB.
4. In the command window, enter

```
adb install -r <full path to apk>
```

**NOTE:** If you have issues with Preferences in a newly installed version of Trepn, try uninstalling either via the command line or via Settings (as indicated below) and installing with **adb install <full path to apk>**.

To uninstall Trepn via command line:

1. Open a command window.
2. Make sure that the Snapdragon MDP or other supported device is connected to the computer via USB.
3. In the command window, enter:

```
adb uninstall com.quicinc.trepn
```

To uninstall Trepn via Settings:

1. Navigate to **Settings**→**Applications**→**Manage Applications**.
2. Tap **Trepn Profiler**.
3. Tap **Uninstall**.
4. At the Uninstall application dialog, tap **OK** to confirm.

## 3.15 Profiling Battery Power

Battery Power is measured in mW or mWh. Battery Power values are given as adjusted values relative to the baseline measurement. For more details, see Baseline Interval in the [General Preferences](#) section.

Connecting a power source to the device while profiling the Battery Power data point can lead to inaccurate data for that data point. It is recommended that you do not plug in a power source while profiling.

If the Battery Power data point is enabled and a power source is connected when **Profile System** or **Profile App** is selected, the prompt message will warn that a charger is connected and suggest removing it. Both prompts give the options **OK** to start profiling anyway, or **Cancel** to return to the home screen.

If a power source gets plugged in while a profiling session is already running and Trepn Profiler is the foreground application, a prompt may be displayed to warn that a charger has been connected. The prompt can be dismissed by pressing the **OK** button to continue.



**NOTE:** A power source can include a wall-charger, USB cable, docking station, or any other powered connection.



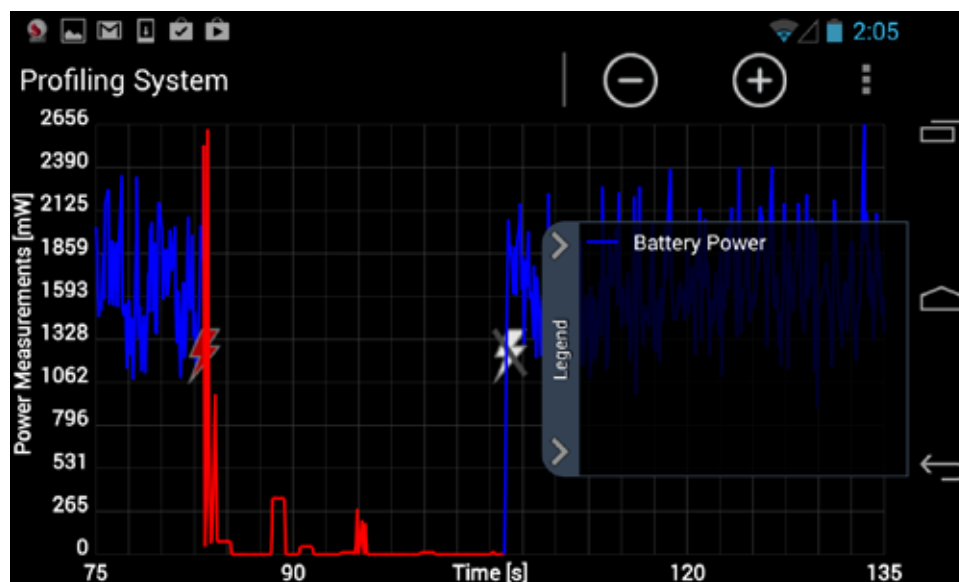
### 3.15.1 Graphing Battery Power

If the Battery Power data point has been enabled in the Data Points preferences list, then it appears on Trepn's main Graph view while profiling or analyzing a session. Use the graph's legend to identify the assigned line color. For details on using the graph, see [Using the Graph](#).

Since power sources can affect the value of the Battery Power data point, the graph has features that identify when a power source is connected, disconnected, and the time during which the data point was being affected:

- n A red lightning icon appears when a power source is attached: 
- n A white lightning icon with a cross appears when a power source is detached: 
- n The color of the line changes to red for the duration of the time that a power source is detected.
- n While the data line is red and a power source is being detected, the red lightning icon will also be shown in the Action Bar above the graph. Tapping this button will display a dialog showing the current charging status.

While the Battery Power data line is the color red, it should not be considered as an accurate reading. It is recommended that you do not plug in a power source while profiling.



## 4 Using Application States

---

### 4.1 Using Android Intents to Identify Application State changes in Trepn Log

In its basic form, Trepn Profiler can be used as-is without making any changes to an application. The Trepn Profiler can be used to launch an application and take note of the graphs that illustrate how much CPU and battery power are being used through the lifetime of the app.

For best results, it is recommended to instrument your code to correlate application behavior with graph activity in the Trepn Profiler.

#### 4.1.1 Manipulating Application State in your app

One important feature of the Trepn Profiler is its ability to capture Android Intents from an application. The Trepn Profiler allows the user to view the Application State with the other selected data points (battery power, CPU loads, etc.), which can be extremely valuable in diagnosing a problem in the application. Developers can add the code snippet below into their applications to send an Intent for the Trepn Profiler to change the Application State at that point. This allows for a definitive breakdown of usage consumption during specific portions of the application.

For example, say the application being profiled is communicating with a web server and a local SQLite database to get/send data. If the application is performing several requests and a spike in power consumption is seen, then it may not be clear which task in the application caused this spike in consumption. The Application State allows the developer to track down exactly which part of the code is causing this spike.

Essentially, Application State aides in the diagnosis of resource consumption problems, so that the developer can move on to solving the problem more efficiently.

## 4.1.2 Code for changing Application State

The Trepn Profiler allows for simply sending a call to update the state, where the Trepn Profiler will increment it by 1, or the Intent can include an integer value. When this integer value is included, the Trepn Profiler will directly assign the state to this supplied integer value. This is useful if the application has many states which can be confusing for users to monitor with auto-incremented values or if the application is alternating between the same states while running (going from state=1 to state=2 and back to state=1, etc.).

The applicable code for an auto-incrementing state is:

```
Intent stateUpdate = new Intent("com.quicinc.Trepn.UpdateAppState");  
sendBroadcast(stateUpdate);
```

The applicable code for a developer-assigned state with a state description is:

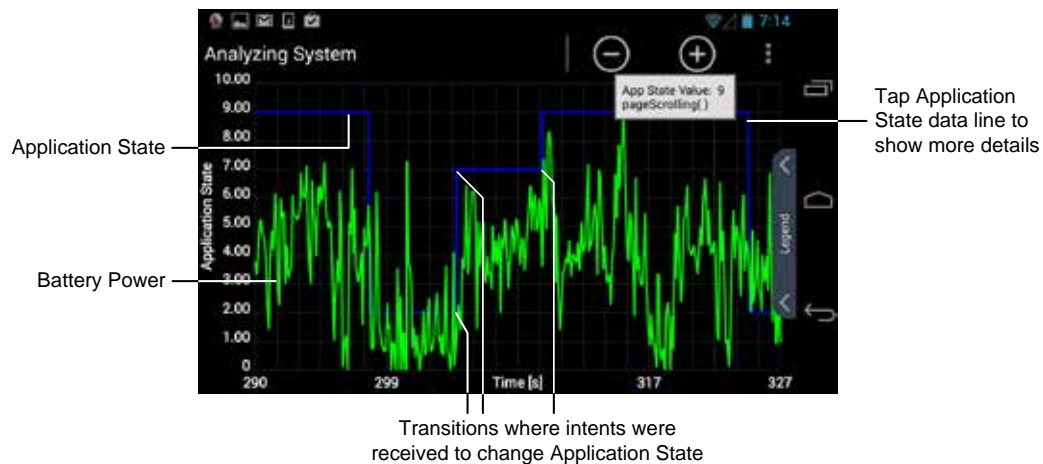
```
Intent stateUpdate = new Intent("com.quicinc.Trepn.UpdateAppState");  
stateUpdate.putExtra("com.quicinc.Trepn.UpdateAppState.Value",  
<int_value>);  
stateUpdate.putExtra("com.quicinc.Trepn.UpdateAppState.Value.Desc",  
<string_value>);  
sendBroadcast(stateUpdate);
```

In this code, change <int\_value> to an integer value to correspond to what you want the Application State in the Trepn Profiler to be when that Intent is received. Change <string\_value> to a string value corresponding to the description of that Application State when the Intent is received.

### 4.1.3 Application State example

This section shows how a developer can use the Trepro Profiler to analyze changes in application behavior using the intent mechanism above.

The user can tap the Application State data line in the graph to display a pop-up with more details about the state at that time. The pop-up shows the Application State's value and the state's description. If no description was provided by the Intent, only the value is displayed.



The following shows the Stats when Application State changes are used.

Analyzing System		
Power Measurements	Average	Total
Battery Power	859.88 [mW]	169.54 [mWh]
openURL() Start: 2.66s End: 2.71s		
Power Measurements	Average	Total
Battery Power	1,142.10 [mW]	1.92 [mWh]
pageScrolling() Start: 8.85s End: 8.86s		
Power Measurements	Average	Total
Battery Power	908.39 [mW]	3.46 [mWh]
scrollingPaused() Start: 22.75s End: 0.00s		
Power Measurements	Average	Total
Battery Power	344.78 [mW]	0.56 [mWh]
openURL() Start: 28.78s End: 28.86s		
Power Measurements	Average	Total
Battery Power	587.17 [mW]	2.11 [mWh]
pageScrolling() Start: 41.89s End: 42.00s		
Power Measurements	Average	Total
Battery Power	1,107.47 [mW]	4.41 [mWh]

**NOTE:** The Application State intents sent more frequently than every 2 or 3 seconds are unlikely to provide meaningful data, due to the limits in the frequency in which the data points can be accessed.

# 5 Using Intents for Automated Testing

---

## 5.1 Automated testing with Trepro Profiler

Android Intents can be used to send commands to start profiling, stop profiling, and load preferences to Trepro Profiler. This feature allows Trepro Profiler to be used in an automated testing environment by providing the ability to start and stop profiling from an external script or another Android app.

In order to successfully use Intents, Trepro Profiler needs to be started before sending any Intents.

**CAUTION:** When using the commands in the following sections, do not copy and paste them directly from this document. There may be character encoding issues that will make the command not execute properly.

### 5.1.1 Commands for sending Intents from an external .bat script

The applicable commands for sending Intents from an external .bat script are:

- n To start Trepro Profiler:

```
adb shell am startservice com.quicinc.trepro/.TreproService
```

A Trepro Profiler icon displays in the Android notification area if TreproService is running.

If the command does not start the Trepro Profiler service, it may be necessary to include the “--user 0” argument within the startservice command or to issue an “adb root” command first.

- n To start profiling:

```
adb shell am broadcast -a com.quicinc.trepro.start_profiling
```

In this code, if Storage Mode is set to Database in [General Preferences](#), the run is given a default name. If Storage Mode is not set to Database, the run will not be saved.

- n It is possible to choose a name to give to the run when starting profiling. To start profiling with a chosen name:

```
adb shell am broadcast -a com.quicinc.trepro.start_profiling -e  
com.quicinc.trepro.database_file "<string_value>"
```

In this command, change <string\_value> to the name you would like to give to the run. The run can be accessed by pressing **Analyze Run** from the Trepro Profiler main menu screen.

- n To stop profiling:

```
adb shell am broadcast -a com.quicinc.trepro.stop_profiling
```

n To load preferences:

```
adb shell am broadcast -a com.quicinc.trepn.load_preferences -e
com.quicinc.trepn.load_preferences_file "<string_value>"
```

In this code, change <string\_value> to the path of the preferences file on the device you would like to load. The default location for preferences is /sdcard/trepn/saved\_preferences. The Intent to load preferences should be sent before sending the Intent to start profiling.

n To save preferences:

```
adb shell am broadcast -a com.quicinc.trepn.save_preferences -e
com.quicinc.trepn.save_preferences_file "<string_value>"
```

In this code, change <string\_value> to the desired name of the preferences file you would like to save on the device. The default location for saved preferences is /sdcard/trepn/saved\_preferences.

If any of the adb shell commands above fail, it may be necessary to enter them directly into the shell.

For example:

```
>adb shell am broadcast -a com.quicinc.trepn.start_profiling
```

This command may need to be entered as:

```
>adb shell
$ am broadcast -a com.quicinc.trepn.start_profiling
```

## 5.1.2 Code for sending Intents from another application

The applicable commands for sending from another application are:

n To start Trepn Profiler:

```
Intent trepnProfiler = new Intent();
trepnProfiler.setClassName("com.quicinc.trepn",
"com.quicinc.trepn.TrepnService");
startService(trepnProfiler);
```

n To start profiling:

```
Intent startProfiling = new Intent("com.quicinc.trepn.start_profiling");
sendBroadcast(startProfiling);
```

This section shows how a developer can use Trepn Profiler to analyze changes in application behavior using the intent mechanism above. In this command, change <string\_value> to the name you would like to give to the run. The run can be accessed by pressing **Analyze Run** from the Trepn Profiler main menu screen.

n To stop profiling:

```
Intent stopProfiling = new Intent("com.quicinc.trepn.stop_profiling");
sendBroadcast(stopProfiling);
```

n To load a .pref preferences file:

```
Intent loadPreferences = new
Intent("com.quicinc.trepn.load_preferences");
loadPreferences.putExtra("com.quicinc.trepn.load_preferences_file",
<string_value>);
sendBroadcast(loadPreferences);
```

In this code, change <string\_value> to the path of the preferences file you would like to load. The default location for preferences is /sdcard/trepn/saved\_preferences. The Intent to load preferences should be sent before sending the Intent to start profiling.

n To save a .pref preferences file:

```
Intent savePreferences = new
Intent("com.quicinc.trepn.save_preferences");
savePreferences.putExtra("com.quicinc.trepn.save_preferences_file",
<string_value>);
sendBroadcast(savePreferences);
```

In this code, change <string\_value> to the desired name of the preferences file you would like to save on the device. The default location for saved preferences is /sdcard/trepn/saved\_preferences.

### 5.1.3 Intents example using an external .bat script

The following example shows how a developer can load preferences, start profiling, and stop profiling using the intent mechanisms above.

The code for the script is:

```
adb shell am startservice com.quicinc.trepn/.TrepnService
ping 123.45.67.89 -n 1 -w 1000 > NUL
adb shell am broadcast -a com.quicinc.trepn.load_preferences -e
com.quicinc.trepn.load_preferences_file
"/sdcard/trepn/saved_preferences/external_script_preferences.pref"
ping 123.45.67.89 -n 1 -w 1000 > NUL
adb shell am broadcast -a com.quicinc.trepn.start_profiling
ping 123.45.67.89 -n 1 -w 20000 > NUL
adb shell am broadcast -a com.quicinc.trepn.stop_profiling
```

The commands are explained line-by-line as follows:

```
adb shell am startservice com.quicinc.trepn/.TrepnService
```

Starts the Trepn Profiler from the external script

```
ping 123.45.67.89 -n 1 -w 1000 > NUL
```

Causes the script to wait for 1 sec (1000 ms) while Trepn Profiler is starting

```
adb shell am broadcast -a com.quicinc.trepn.load_preferences -e
com.quicinc.trepn.load_preferences_file
"/sdcard/trepn/saved_preferences/external_script_preferences.pref"
```

Loads the “/sdcard/trepn/saved\_preferences/external\_script\_preferences.pref” preferences file

```
ping 123.45.67.89 -n 1 -w 1000 > NUL
```

Causes the script to wait for 1 sec (1000 ms) while Trepn Profiler loads the preferences

```
adb shell am broadcast -a com.quicinc.trepn.start_profiling
```

Sends Intent to Trepn Profiler to start profiling

```
ping 123.45.67.89 -n 1 -w 20000 > NUL
```

Causes the script to wait for 20 seconds (20000 ms) while profiling

```
adb shell am broadcast -a com.quicinc.trepn.stop_profiling
```

Sends Intent to Trepn Profiler to stop profiling



# 6 Using Trepn Profiler Efficiently

---

Trepn Profiler is an on-target profiler. While this offers the benefit of being easy to use, it also uses system resources on the target while profiling. The additional overhead may impact the performance of the device or the application being tested. If the overhead is too high, Trepn Profiler may also begin to capture data at less frequent intervals.

## 6.1 Managing Trepn Profiler's Impact

Trepn Profiler includes many useful features but not all of them may be needed for every profiling session.

Here are some preference changes and tips to use Trepn Profiler more efficiently.

**NOTE:** For details on editing preferences, see [Preferences](#).

### General preferences

- n Storage Mode – Change the Storage Mode preference to None or Main Memory. Database mode continuously saves data to a database on the micro-SD card/internal memory. This leads to higher CPU usage overhead than the other modes. The None storage mode does not save any of the data points. The Main Memory storage mode saves data in RAM and can be saved to a database or .csv file when profiling is complete. Due to the limited amount of RAM provided to Trepn Profiler by Android, not all data is guaranteed to be saved. For a typical session, approximately the last hour of data will be saved, but this can vary depending on the amount of data profiled in the session. For more information on Storage Mode settings, see [General Preferences](#).
- n Show Per-Application Statistics – Turn this feature off to reduce overhead. This preference determines whether CPU and Memory Stats are collected for all applications that were active during the profiling period. The amount of overhead varies depending on the number of active applications.

### Data points

- n Turn off any data points that are not necessary. Each additional data point contributes to the CPU usage overhead.
- n Although there is no theoretical maximum to the number of data points that can be traced, the frequency of polling decreases as an inverse of the number of data points selected. Therefore, a small number of data points will produce a more frequent polling interval. If Trepn Profiler appears to slow down or produce sparse data points, it is recommended to turn off unnecessary data points from the polling set and to trace one targeted subset at a time.

## Overlays

- Turn off any overlays that are not necessary. Turning off unused overlays (by pressing on an overlay in preferences and choosing overlay type, None) helps reduce overhead when Trepro Profiler is running in the background. For more information on overlays, see [Using Overlays](#).

## Additional usage tips

- Do not use the Graph or Stats during profiling. These screens are frequently refreshed during profiling and can lead to additional CPU usage overhead. For best performance, navigate away from Trepro by tapping **Home** after starting profiling. Trepro will continue profiling data in the background. Combine with the suggestions above for maximum efficiency.

# A What the Tool Profiles/Captures

---

## A.1 Data Points

This section outlines the data points that Trepn Profiler can profile.

### A.1.1 Power Stats

Power Stats are measured in mW or mWh. Power Stat values are given as adjusted values relative to the baseline measurement. Power values are available for the following:

- n Battery Power – Power consumed by the battery
- n Battery Status – Charging status of the battery, automatically selected when the Battery Power data point is selected, but not selectable individually. This data point appears in the .csv file if the Battery Power data point is profiled.
  - .. 0 – Not charging
  - .. 1 – Charging (USB)
  - .. 2 – Charging (AC)
  - .. 3 – Charging (generic source)

### A.1.2 CPU Stats

CPU Stats include the following:

- n CPU Load – Load across all cores
- n CPU0 Frequency – Clock speed for first CPU core
- n CPU0 Interrupts – Number of interrupts on first CPU core
- n CPU0 Online State – Indication of whether first CPU core is powered up
  - .. 0 – Offline
  - .. 1 – Online
- n CPU1 Frequency – Clock speed for second CPU core
- n CPU1 Interrupts – Number of interrupts on second CPU core
- n CPU1 Online State – Indication of whether second CPU core is powered up
  - .. 0 – Offline
  - .. 1 – Online

- 1       n   CPU2 Frequency<sup>1</sup> – Clock speed for third CPU core
- 2       n   CPU2 Interrupts<sup>1</sup> – Number of interrupts on third CPU core
- 3       n   CPU2 Online State<sup>1</sup> – Indication of whether third CPU core is powered up
- 4           ..   0 – Offline
- 5           ..   1 – Online
- 6       n   CPU3 Frequency<sup>1</sup> – Clock speed for fourth CPU core
- 7       n   CPU3 Interrupts<sup>1</sup> – Number of interrupts on fourth CPU core
- 8       n   CPU3 Online State<sup>1</sup> – Indication of whether fourth CPU core is powered up
- 9           ..   0 – Offline
- 10          ..   1 – Online
- 11       n   GPU2D0 Usage<sup>2</sup> – Usage for the first GPU core
- 12       n   GPU2D1 Usage<sup>2</sup> – Usage for the second GPU core
- 13       n   GPU3D Frequency – Clock speed for the 3D GPU core
- 14       n   GPU3D Usage – Usage for the 3D GPU core
- 15       n   Normalized CPU Load – Normalized CPU load across all cores; see [Normalized CPU loads](#)
- 16           for more information

---

<sup>1</sup>Quad-core devices only.

<sup>2</sup>Not available on all devices.

### A.1.3 Network Stats

Network Stats include:

n Bluetooth® State – State of the Bluetooth radio

.. 0 – Off

.. 1 – Turning on

.. 2 – On

.. 3 – Turning off

n Data State – State of the cellular radio

.. 0 – Disconnected

.. 1 – Connecting

.. 2 – Connected, Dormant

.. 3 – Connected, No traffic

.. 4 – Connected, Sending traffic

.. 5 – Connected, Receiving traffic

.. 6 – Connected, Sending and receiving traffic

n WiFi RSSI Level – RSSI level for the Wi-Fi connection; values are expressed as dBm

n WiFi State – State of the Wi-Fi connection:

.. 0 – Wi-Fi currently being disabled

.. 1 – Wi-Fi disabled

.. 2 – Wi-Fi currently being enabled

.. 3 – Wi-Fi enabled

.. 4 – Wi-Fi in an unknown state; this state occurs when an error happens while enabling or disabling

## A.1.4 Other Stats

Other Stats include the following:

- n Application State – Data point for developers to instrument their application/code; Broadcast Intent for the Trepro Profiler to receive and update this value, allowing a more precise understanding of data points; value is a signed 32-bit integer. See [Using Android Intents to Identify Application State changes in Trepro Log](#).
- n Percent Battery Remaining – Remaining charge in the battery
- n GPS State – Current state of the GPS system
  - .. 0 – GPS Stopped
  - .. 1 – GPS Unknown State
  - .. 2 – GPS Running
- n Memory Usage – Total memory usage of the system
- n Screen Brightness – Brightness level of the screen's backlight
- n Screen State – State of the screen:
  - .. 0 – Off
  - .. 1 – On
- n Wakelocks – Instantaneous number of wakelocks being held
- n Wifilocks – Instantaneous number of Wifilocks being held

## A.2 Normalized CPU load

The operating frequency of CPUs can be scaled down depending on the workload. CPU Load reports the usage with respect to the operating frequency. Normalized CPU Load reports the usage with respect to the maximum frequency. As a result, CPU Load shows the load with respect to the scaled potential and Normalized CPU Load shows the load with respect to the maximum potential.

Consider the following example:

Take a device that has a single CPU with maximum frequency of 1000 MHz. Consider the case where the workload is low and the operating frequency is scaled down to 500 MHz. In one second, there are 500M clock cycles. Say the device uses 100M of those cycles for work and is idle for the remaining cycles. CPU Load would show 20% (100/500). However, the maximum potential of the CPU was 1000M cycles, so Normalized CPU Load would show 10% (100/1000).