Little Eye Labs

Data Sheet

v 2.4

Last Updated: 7 Dec 2013

Table of Contents

Little Eye

Little Eye Labs builds mobile app analysis tools for app developers and testers. These tools provide detailed insights about the behavior of the app, enabling easy benchmarking and optimization of resources.

Supported Platforms

Little Eye is a desktop tool that works alongside the Integrated Development Environment used for developing Android Apps. The currently supported platforms for the Little Eye desktop tool are

1. Windows
    a. Windows XP
    b. Windows Vista (32bit, 64bit)
    c. Windows 7 (32bit, 64bit)
    d. Windows 8 (32bit, 64bit)
2. Mac OS X
    a. Snow Leopard
    b. Lion
    c. Mountain Lion
3. Linux

    Little Eye has been tested on Ubuntu 12.04 (32bit and 64 bit), but should work on all Linux platforms.

Supported Smartphone OSs

Little Eye works with these Android OS versions:

1. Android 2.3 (Gingerbread)
2. Android 3.0, 3.1, 3.2 (Honeycomb)
3. Android 4.0 (Ice Cream Sandwhich)
4. Android 4.1, 4.2, 4.3 (Jelly Bean)
5. Android 4.4 (Kit Kat)

Pre-requisites to running Little Eye

The following software is required to be installed on your desktop before Little Eye runs

1. Java JRE or SDK - V 1.6 + (Java 6 or above)
2. Android SDK
3. USB debugging is enabled on the phone
4. If you are using Windows, you may need to ensure that you are connecting as a "Camera (PTP)" rather than as a "Media Device (MTP)"
5. If you are using Windows, you may need to install your phone's drivers before Little Eye can connect. Please see this link for information on USB drivers.

Supported Performance Metrics

Little Eye automatically collects several metrics while it is monitoring an app.

1. Power Consumed by the App
    a. App Total Power
        i. CPU Power
        ii. Wifi Power
        iii. Display Power
        iv. GPS Power (Experimental)
        v. 3G Power
    b. System Total Power
2. Network Data consumption
    a. System total data (Sent/Received)
    b. App total data
        i. Total App Data Sent
        ii. Total App Data Received
3. Memory Consumption
    a. Memory consumed by App - Dalvik Memory
    b. Memory consumed by App - Native Memory
    c. App PSS Memory
    d. Dalvik Threshold - Max Dalvik memory available to an App on the current device
4. CPU
    a. CPU % cycles consumed at a user level
    b. CPU % cycles consumed at a system level

5. Threads
    a. List of active threads
    b. CPU consumed by each thread
6. Disk Space consumption

Supported Events List

Little Eye automatically tracks events that happen on the Android system. Here's a list of Events currently being collected by Little Eye.

1. Power
    a. Screen On/Off
    b. Wakelock
    c. Application State Change
    d. Brightness Change
    e. Process State Change
    f. Orientation Change
    g. Activity Change
    h. Battery State Change
2. Wireless Data
    a. Connection State Change
3. Memory
    a. GC Concurrent Trigger
    b. GC for Alloc Trigger
    c. GC Explicit Trigger
    d. GC External Alloc Trigger
    e. Heap Dump
4. Log
    a. Error
    b. Debug
    c. Info
    d. Warning
    e. Verbose
    f. Assert

Disk Consumption

Little Eye can track certain directories to watch how the disk consumed by these directories grows. You can configure the directories to be monitored under Preferences.

Understanding Events

Wakelock - This event is shown when there is a change in state of any wakelocks held by your app, including when your app acquires or releases a wakelock.

Application State Change - This event is shown when the state of your app changes between foreground, visible, background, service, stopped etc.

Process State Change - This event is triggered when a process belonging to your app (identified by the uid) is created or killed.

Orientation Change - This event is shown when the device's orientation changes.

Activity Change - This event is shown when the foreground activity on the device is changed, irrespective of whether it belongs to your app or not.

Brightness Change - This event is shown if the brightness on the device is changed, either by the app or by the user

Screen On/Off - This event is shown when the screen is turned on or off.

Battery State Change - This event is shown when the battery state changes between charging or discharging (in other words, whether you're testing the device plugged in or disconnected) or between LOW and HIGH.

Connection State Change - This event is shown whenever there is a change in the connection state eg., a radio getting disconnected or connected. The event shows whether it is Wifi, 2G or 3G radio along with the change.

GC Triggers - These events are shown whenever the Java Garbage Collector runs. The GC may be triggered for various reasons, and the type of trigger corresponds to the GC trigger events.

Log Events - An event is shown whenever there is a log message of corresponding level (Verbose, Debug, Error, Warning etc). In the default settings, only Error events are visible on the time line.

Custom Events

Apart from the standard events that Little Eye reports, you can also include your own, custom events. If you want to create an event to be identified in Little Eye, please log a message from your App into Logcat with the prefix **LEStartEvent** or **LEEndEvent**. These will be captured by Little Eye and shown in the event timeline along with the rest of the System and App events.

Power Model

Little Eye computes the power consumed by the App under test using a Power Model. Currently, there are two supported Power Models- Google Nexus One and Google Galaxy Nexus. The power model is automatically selected to be the closest one to the device that is currently being tested.

The Power consumed is calculated based on the consumption of the following parameters by the App under test, running on the device that is connected. Note that irrespective of which device is connected, the power model is computed using the power model.

1. CPU
    a. Number of Cores
    b. CPU Cycles consumed
    c. Frequency stepping of the processor
    d. Frequency scale
2. Display
    a. Brightness of the screen
    b. Actual color of pixels on the screen, based on a uniform sample
3. Wifi Radio
    a. Wifi radio state
    b. Power levels of the Wifi Radio
4. 3G Radio
    a. 3G radio state
    b. Power levels of the 3G Radio
5. GPS
    a. State of the GPS sensor (On/Off/Looking for Fix)

Note about 3G Power: There are multiple protocols for 3G data transfer - UMTS, HSPA, HSPA+ etc. Little Eye currently models power computation based on UMTS network. In this, the state

transitions of 3G cellular radio are determined by the network operator. Little Eye models one typical network operator model.

Note about the Display Power: The display power is always modeled for a OLED screen.

Understanding Threads

Little Eye will capture and show all the threads that the process creates. The threads view is updated to include the CPU consumed by the thread as a percentage of the App's total. Both instantaneous and cumulative consumption are available in the threads view. As with all other metrics, you can go back in your run and see what threads were active at any point in the App.

Release Apps vs Debug Apps

Little Eye can monitor any app, irrespective of whether it is a release app or a debug app. If the App is built using debug mode, some additional features are available.

Additional Features supported on Debug-build apps

1. Capture Heap Dumps
2. Set Heap Dump triggers for automatic Heap Dumps at regular intervals

Video Capture

As the App under test is being monitored, Little Eye collects the video of the screen in the background. The screens are transferred back to the desktop as soon as the test is stopped.

The screen is captured at 1/4th the original resolution of the phone, at 3 frames per second (3 FPS).

Reports

At the end of a run, Little Eye can generate a report based on the run. You can select any subset of runs for the given App to generate an aggregated report

The following data is available in the report

1. Performance Score
    a. The performance score is calculated on various attributes, and the dimensions they are calculated on are shown in the report
2. Power
    a. Foreground Battery Consumption estimate

   b. Background Battery Consumption estimate
   c. Breakdown of power consumed by component
   d. Time spent by the App in foreground and background
3. Wireless Data
   a. Total Data Sent/Received
   b. Cost of data transferred to the user (via configurable data plan)
   c. Total Data Sent/Received by the App in the foreground and background
4. Memory
   a. Peak/Average Memory consumption
   b. PSS peak
   c. Number of GC runs
   d. Time impact of GC runs
   e. CPU f
   f. Average CPU % cycles consumed

All of these sections are available to be customized in the generated report via the "Customize..." option in the report view.

Http Stats

Little Eye allows the collection of HTTP stats via instrumentation. You can now launch an app after instrumenting it to view detailed HTTP requests that the App is making. The HTTP stats view will show each server that the App is connecting to, along with bytes sent/received and latency for each server.
To enable the collection of HTTP stats, the App needs to be “instrumented” first.

Insights

Along with the report, several insights are automatically generated based on the runtime performance. The list of supported insights are:

1. Holding a wakelock for longer than necessary
2. Not testing orientation changes
3. Not testing all activities
4. Not batching Wifi/Data transfer
5. Likelihood of App crashes on lower memory devices
6. Coverage of Activities by the test case

Saving (Little Eye Data Format)

All data collected during a data run is saved in the .lel file. It includes all performance metrics, events and the video. Reports can also be re-generated by loading the .lel file.

All runs in Little Eye are also automatically saved in the "autosave" directory.

Exporting Reports

The generated reports can be exported to a PDF format or an HTML format (with embedded images)

Exporting CSV

All the data collected by Little Eye can also be exported into a CSV format, so you can analyze it outside the application. A CSV file can be exported both from the UI (by clicking on the export icon) or from the command line.

Little Eye Headless Mode

Since v1.1, Little Eye can run in headless mode from the command line, and connect directly to an android device, run a test case, and generate the .lel file. Command line options that can be passed to Little Eye include

```
    monitor -- to monitor any given application
    export csv -- to export a given data file to a csv file.
    help [monitor|export] -- to print help for the given command

The monitor subcommand can be used to monitor any give application.
Usage: littleeye-nowin.bat monitor --package=<app-package> [options] [command-to-run]
   --package=<app-package-name>    Package name of the app to be monitored
   --maxtime=<time-in-seconds>     Little Eye monitoring for this specified time

   [command-to-run]                Little Eye starts this command after the start
                                    of the monitoring and waits until the command
                                    finishes or until the specified time finishes.
                                    This and the arguments that the command takes
                                    should be specified at the end. At least one of -
-maxtime or the command-to-run must be specified

 Additionally, the following options could be specified.

   --device=<device-identifier>   <device-identifier> is the device string shown
                                     with the command 'adb devices'.
                                     Little Eye monitors the app on this device.
                                     If not specified the first device is chosen
```

```
   --output=<output-file-path>   Little Eye will save the monitoring data in
                                   the file. If this is not specified, it is
                                   saved to
                                   <user-home>/littleeye/autosaved/<appname>-<date>-
<time>.lel

   --screenrecord=<yes/no>       Whether to record the screen (of device)
                                   while monitoring. Default is 'yes'

   --startapp=<yes/no>           If this option is set, Little Eye will start
                                   the app if it is not in foreground.
                                   Default is 'yes'

   --silent=<yes/no>             If no, Little Eye prints verbose information.
                                   Default is 'no'

The export subcommand can be used to export any Little Eye data file into csv format.
Usage: littleeye-nowin.bat export [--format=csv] --output=<output-file-path> <Little
Eye Data File(.lel)>
   [--format=csv]  Little Eye will export into this given format. Only csv is
currently supported.
                    This is optional and default is csv.
   --output=<output-file-path>   Little Eye will export the data in the specified
file.
```

Little Eye Performance Impact

As with any performance monitoring tool, Little Eye will add some overhead to the app being profiled. The overhead is dependent on several factors, including power of the CPU, number of cores and the size of the display.

Little Eye's overhead on App performance can be between 5% - 15%. If your App's performance is affected severely, you may try disabling the video capture, which is the feature with the most overhead.

Since Little Eye runs in an external process on the phone, the performance overhead does not affect the performance and power numbers of the App under test.

Licensing

Little Eye has a built in licensing service that checks for a license from the Little Eye license server at littleeye.co. The licensing checks are done over HTTPS, so port 443 to the licensing server should be reachable from the machine running Little Eye. If you have a proxy server that is used to connect to the internet, you can configure it under "*Window -> Preferences ->Little Eye -> Internet Proxy*"

Little Eye uses a floating license model. This means, Little Eye can be installed on any number of computers, but only one instance of Little Eye can be active at a time.

If there was no license or if the license was invalid or expired, Little Eye reverts to a read-only mode. In this mode, .lel files can be loaded and browsed, but a new app monitoring session cannot be launched.

Further Questions

For any further questions not included in this data sheet, please email support@littleeye.co