google / **battery-historian**

Battery Historian is a tool to analyze battery consumers using Android "bugreport" files.

15 commits **1** branch **0** releases **5** contributors

Branch: **master** New pull request New file Find file HTTPS `https://github.com/googl` Download ZIP

jocelyndang Download specific closure compiler version to avoid flakiness. … Latest commit `fa8b40f` on 1 Feb

analyzer	External release of Battery Historian 2.0.	10 months ago
build	External release of Battery Historian 2.0.	10 months ago
checkinparse	External release of Battery Historian 2.0.	10 months ago
checkinutil	External release of Battery Historian 2.0.	10 months ago
cmd	Change setup.sh compile regex to only consider JS files when compiling	2 months ago
csv	External release of Battery Historian 2.0.	10 months ago
js	External release of Battery Historian 2.0.	10 months ago
packageutils	External release of Battery Historian 2.0.	10 months ago
parseutils	External release of Battery Historian 2.0.	10 months ago
pb	External release of Battery Historian 2.0.	10 months ago
presenter	External release of Battery Historian 2.0.	10 months ago
screenshots	External release of Battery Historian 2.0.	10 months ago
sliceparse	External release of Battery Historian 2.0.	10 months ago
static	External release of Battery Historian 2.0.	10 months ago
templates	External release of Battery Historian 2.0.	10 months ago
LICENSE	Add proper license text	2 years ago
README.md	Change setup.sh compile regex to only consider JS files when compiling	2 months ago
historian.py	External release of Battery Historian 2.0.	10 months ago
regen_proto.sh	External release of Battery Historian 2.0.	10 months ago
setup.sh	Download specific closure compiler version to avoid flakiness.	a month ago

README.md

Battery Historian 2.0

Battery Historian is a tool to inspect battery related information and events on an Android device (Android 5.0 Lollipop and later: API Level 21+) while the device was on battery. It allows application developers to visualize system and application level events on a timeline and easily see various aggregated statistics since the device was last fully charged.

Introduction

Battery Historian 2.0 is a complete rewrite in Go and uses some JavaScript visualization libraries to display battery related events on a timeline with panning and zooming functionality. In addition, v2.0 allows developers to pick an application and inspect the metrics that impact battery specific to the chosen application.

Getting Started

If you are new to the Go programming language:

- Follow the instructions available at http://golang.org/doc/install for downloading and installing the Go compilers, tools, and libraries.
- Create a workspace directory according to the instructions at http://golang.org/doc/code.html#Organization and ensure that `GOPATH` and `GOBIN` environment variables are appropriately set and added to your `$PATH` environment variable. `$GOBIN should be set to $GOPATH/bin` .

Next, install Go support for Protocol Buffers by running go get.

```
# Grab the code from the repository and install the proto package.
$ go get -u github.com/golang/protobuf/proto
$ go get -u github.com/golang/protobuf/protoc-gen-go
```

The compiler plugin, protoc-gen-go, will be installed in $GOBIN, which must be in your $PATH for the protocol compiler, protoc, to find it.

Next, download the Battery Historian 2.0 code:

```
# Download Battery Historian 2.0
$ go get -u github.com/google/battery-historian/...

$ cd $GOPATH/src/github.com/google/battery-historian

# Compile Javascript files using the Closure compiler
$ bash setup.sh

# Run Historian on your machine (make sure $PATH contains $GOBIN)
$ go run cmd/battery-historian/battery-historian.go [--port <default:9999>]
```

Remember, you must always run battery-historian from inside the `$GOPATH/src/github.com/google/battery-historian` directory:

```
cd $GOPATH/src/github.com/google/battery-historian
go run cmd/battery-historian/battery-historian.go [--port <default:9999>]
```

How to take a bug report

To take a bug report from your Android device, you will need to enable USB debugging under `Settings > System > Developer Options` . On Android 4.2 and higher, the Developer options screen is hidden by default. You can enable this by following the instructions here.

Next, to obtain a bug report from your development device

```
$ adb bugreport > bugreport.txt
```

Start analyzing!

You are all set now. Run `historian` and visit http://localhost:9999 and upload the `bugreport.txt` file to start analyzing.

By default, Android does not record timestamps for application-specific userspace wakelock transitions even though aggregate statistics are maintained on a running basis. If you want Historian to display detailed information about each individual wakelock on the timeline, you should enable full wakelock reporting using the following command before starting your experiment:

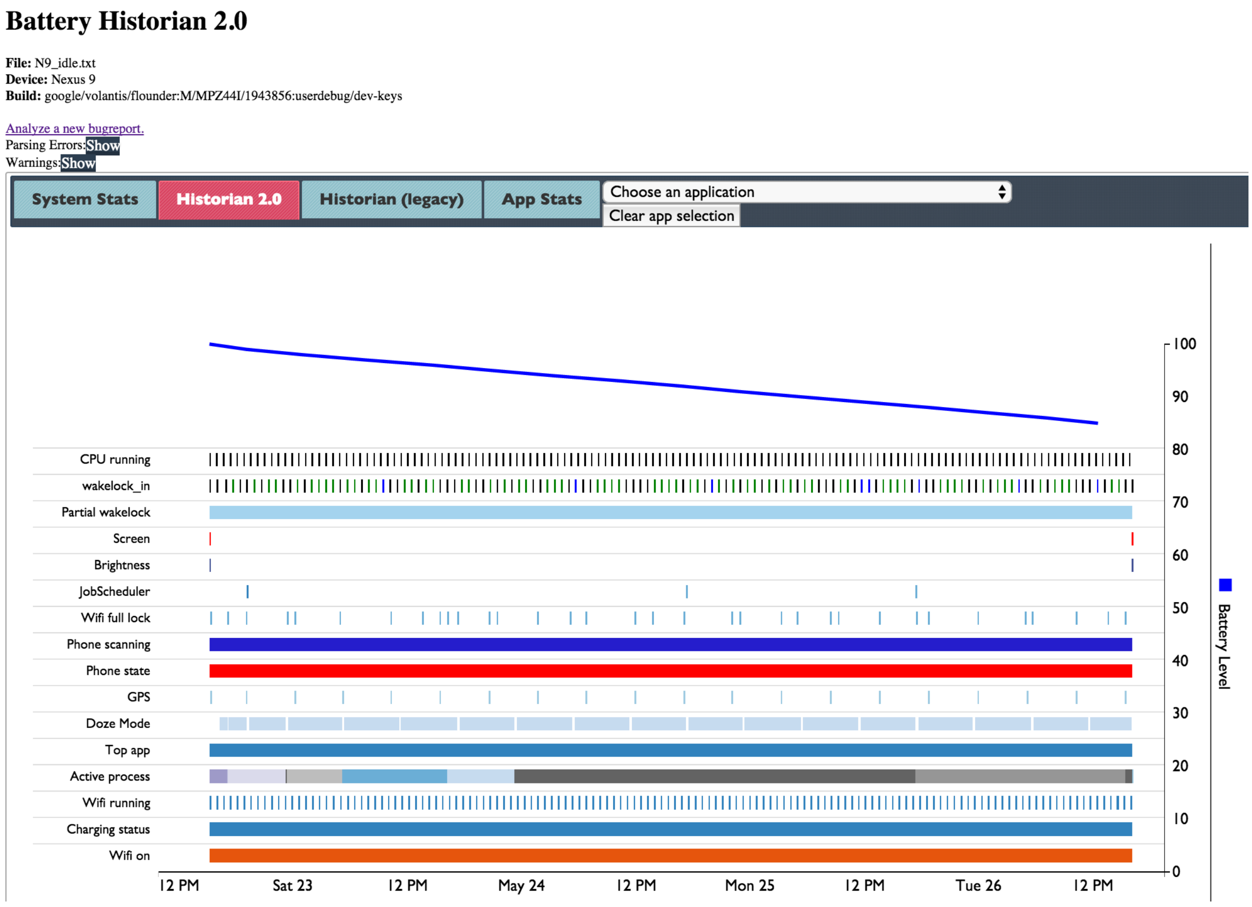
```
adb shell dumpsys batterystats --enable full-wake-history
```

Note that by enabling full wakelock reporting the battery history log overflows in a few hours. Use this option for short test runs (3-4 hrs).

To reset aggregated battery stats and timeline at the beginning of a measurement:

```
adb shell dumpsys batterystats --reset
```

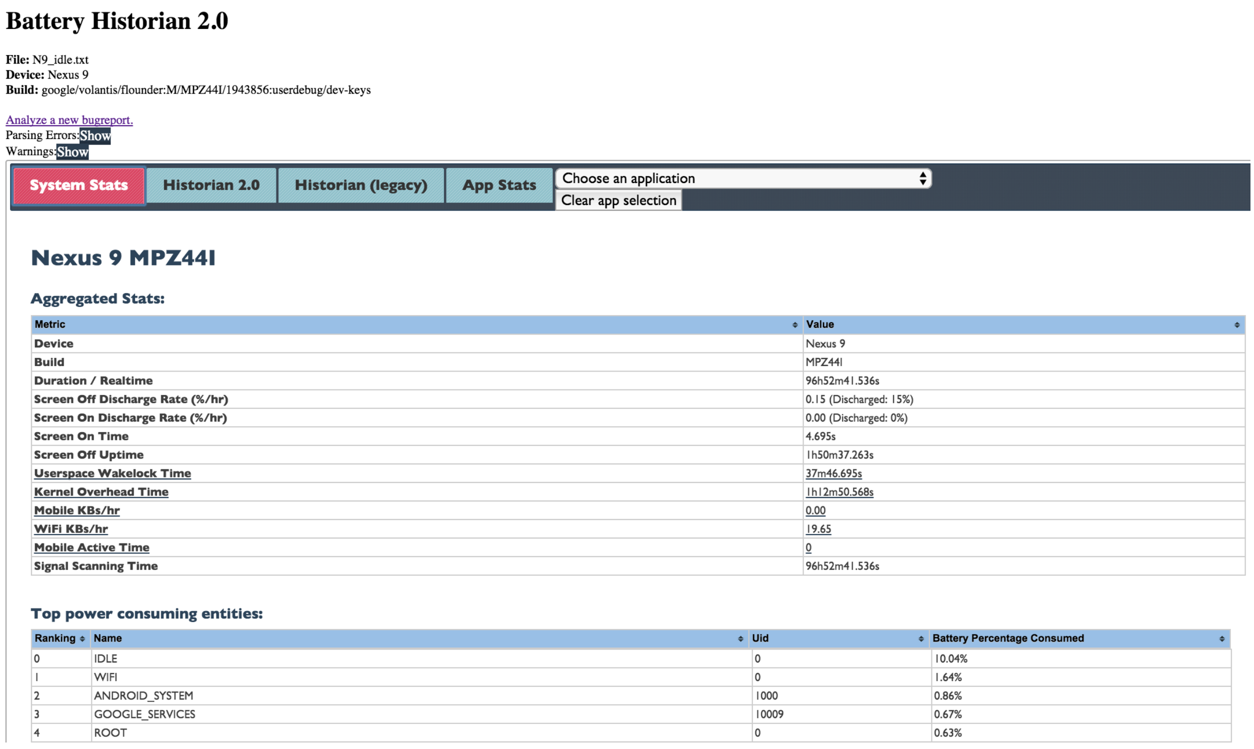
Screenshots



Battery Historian 2.0

File: N9_idle.txt
Device: Nexus 9
Build: google/volantis/flounder:M/MPZ44I/1943856:userdebug/dev-keys

Analyze a new bugreport.
Parsing Errors:Show
Warnings:Show



Battery Historian 2.0

File: N9_idle.txt
Device: Nexus 9
Build: google/volantis/flounder:M/MPZ44I/1943856:userdebug/dev-keys

Analyze a new bugreport.
Parsing Errors:Show
Warnings:Show

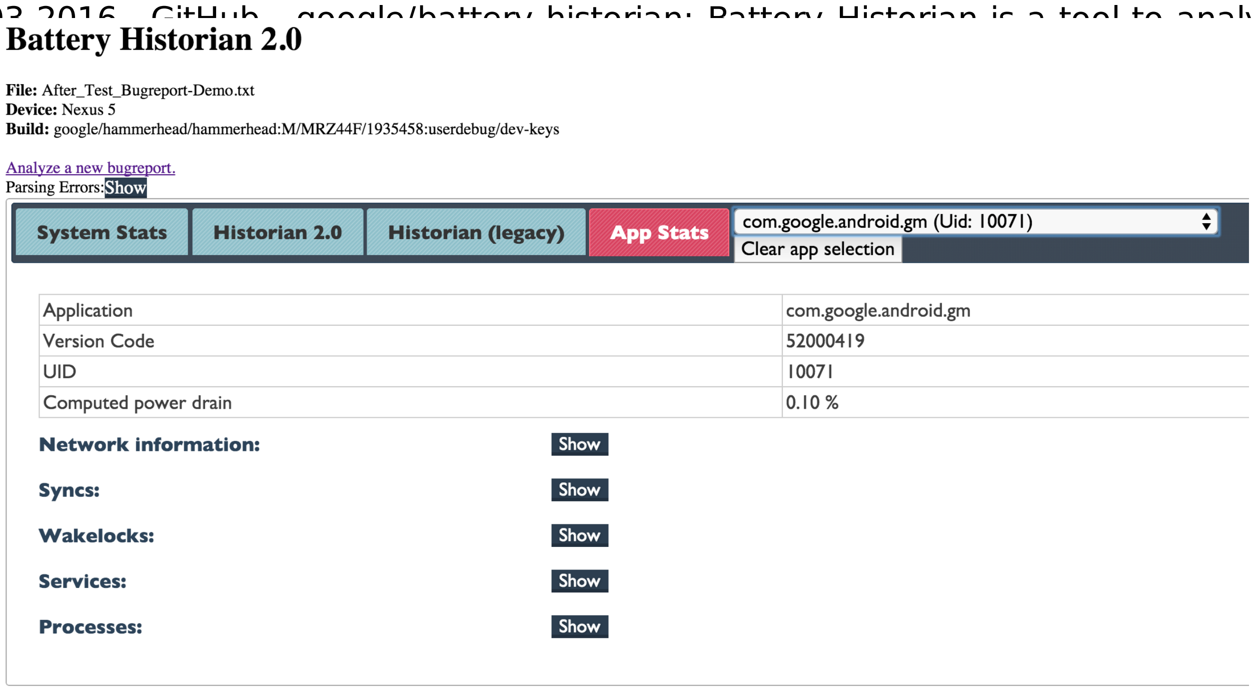
Nexus 9 MPZ44I

Aggregated Stats:

Metric	Value
Device	Nexus 9
Build	MPZ44I
Duration / Realtime	96h52m41.536s
Screen Off Discharge Rate (%/hr)	0.15 (Discharged: 15%)
Screen On Discharge Rate (%/hr)	0.00 (Discharged: 0%)
Screen On Time	4.695s
Screen Off Uptime	1h50m37.263s
Userspace Wakelock Time	37m46.695s
Kernel Overhead Time	1h12m50.568s
Mobile KBs/hr	0.00
WiFi KBs/hr	19.65
Mobile Active Time	0
Signal Scanning Time	96h52m41.536s

Top power consuming entities:

Ranking	Name	Uid	Battery Percentage Consumed
0	IDLE	0	10.04%
1	WIFI	0	1.64%
2	ANDROID_SYSTEM	1000	0.86%
3	GOOGLE_SERVICES	10009	0.67%
4	ROOT	0	0.63%



Battery Historian 2.0

File: After_Test_Bugreport-Demo.txt
Device: Nexus 5
Build: google/hammerhead/hammerhead:M/MRZ44F/1935458:userdebug/dev-keys

Analyze a new bugreport.
Parsing Errors: Show

Application	com.google.android.gm
Version Code	52000419
UID	10071
Computed power drain	0.10 %

Network information: Show

Syncs: Show

Wakelocks: Show

Services: Show

Processes: Show

Advanced

The following information is for advanced users only who are interested in modifying the code.

Modifying the proto files

If you modify the proto files (pb/*/*.proto), you will need to regenerate the compiled Go output files using `regen_proto.sh` .

Other command line tools

```
# System stats
$ go run exec/local_checkin_parse.go --input=bugreport.txt

# Timeline analysis
$ go run exec/local_history_parse.go --summary=totalTime --input=bugreport.txt
```

Support

- G+ Community (Discussion Thread: Battery Historian): https://plus.google.com/b/108967384991768947849/communities/114791428968349268860

If you've found an error in this sample, please file an issue: https://github.com/google/battery-historian/issues

License

Copyright 2016 Google, Inc.

Licensed under the Apache License, Version 2.0 (the "License"); you may not use this file except in compliance with the License. You may obtain a copy of the License at

http://www.apache.org/licenses/LICENSE-2.0

Unless required by applicable law or agreed to in writing, software distributed under the License is distributed on an "AS IS" BASIS, WITHOUT WARRANTIES OR CONDITIONS OF ANY KIND, either express or implied. See the License for the specific language governing permissions and limitations under the License.

© 2016 GitHub, Inc. Terms Privacy Security Contact Help

Status API Training Shop Blog About Pricing