Google Glass Forensics

By Julie Desautels

Champlain College

April 2014

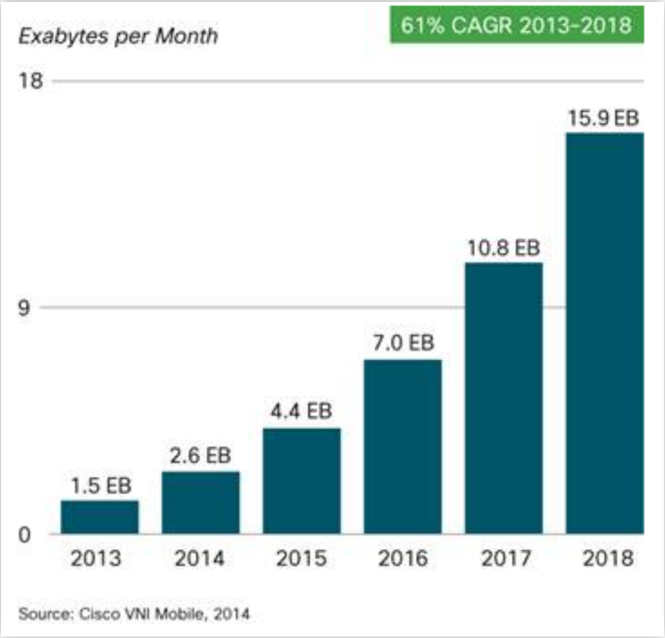
Abstract

There has been an interesting new trend in mobile technology shifting towards wearable tech. Wearable tech includes incorporating digital components into the things we wear, to further integrate technology into our every day lives. Google Glass is a wearable device that has been popular in the media lately. While interest in this new device grows, there is still little research available regarding forensically analyzing this wearable technology. The ability to understand and analyze mobile devices in today's digital world proves to be important, as mobile devices can provide evidence in forensic cases. This paper focuses on understanding the forensic artifacts on Google Glass relating to timestamps. Timestamps are important because it can help recreate a picture of what the user was doing on a device over a particular time. Creating a timeline can turn a collection of technical terms into a "story" of what a user was doing over a period of time. This paper will explain how to determine if Glass was on or off at a specific time, analyze battery levels to determine user activity, extract user voice command files from the device, understand the time artifacts on the Google Glass timeline, receive information about what mobile device Google Glass was connected to and when, and much more. While this is just the start to a huge field of research, it should aid in future Google Glass research, and help an examiner begin their Google Glass analysis.

Table of Contents

Introduction

Mobile technology has changed the way people live, and there is no denying it is growing in popularity. Cisco has done research on mobile device trends, showing, “global mobile data traffic grew 81 percent in 2013,” and “reached 1.5 exabytes per month at the end of 2013”. Their research indicates the use of mobile devices will continue to grow over the next five years. Figure 1 from a Cisco white paper shows the predicted trends in mobile device data traffic (cisco.com).



Source: Cisco VNI Mobile, 2014

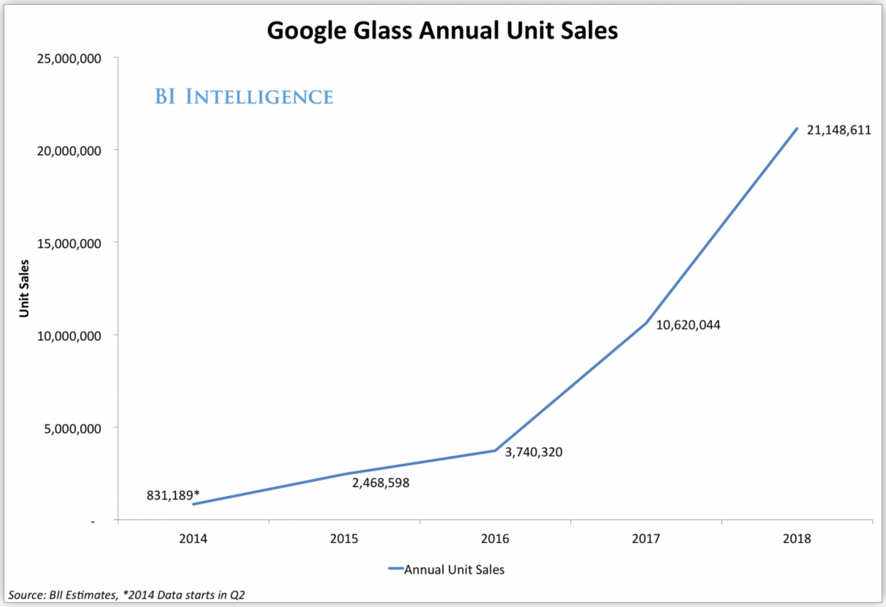
Figure 1: Cisco's Mobile Data Traffic Predictions

Cells phones account for a large amount of current mobile device usage. However, with these growing trends in mobile traffic, a large part of this increase may be brought on by a new kind of mobile device, wearable technology.

Wearable technology is a way of further integrating the mobile devices that have become such a huge part of people’s everyday lives. With technology like smart watches, smart activity wristbands, and digital glasses, people can further integrate technology seamlessly into the things they do daily. One particular wearable technology that stands out from the others is Google Glass.

Google Glass is a wearable device which allows the user to take pictures, record videos, send messages, get directions, make phone calls, and Google search from a pair of “glasses” they wear. Downloadable apps give the wearable device more functions. Using voice commands, the user can choose to control the device audibly, which provides a unique hands free experience.

According to Business Insider, in the next five years Google Glass sales are estimated to reach 21,148,611 as shown in Figure 2. As these devices become more relevant in people's everyday lives, it is more likely to be brought up as a new form of evidence in forensic cases. During digital forensic investigations, wearable technology such as Google Glass will be a new source of evidence to help build a stronger case. Being able to properly examine these devices can help forensic examiners understand the data they are given.



Source: BII Estimates, *2014 Data starts in Q2

Figure 2: Business Insider's Google Glass Sales Predictions

Because it is so important to understand how devices such as Google Glass work for forensic investigations, timestamp information on Google Glass will be analyzed in this project. The goal is to identify artifacts that have time information associated with them, and fully understand exactly what each of these timestamps are indicating. By doing so, the examiner can understand what a user was doing on the device over a period of time.

About Google Glass

Google Glass was first announced April 4, 2012 through a concept video uploaded on YouTube (youtube.com). Google Glass is a pair of "glasses" frames with a digital device component that allows user interaction through voice commands. Google Glass has the capability to take pictures, record videos, Google search, get GPS directions, take phone calls and Google Hangout. More device capabilities are available through downloading Google Glass applications.

Glass can also be paired with a user's cell phone, for further capabilities by using the phone's Bluetooth data.

Google Glass is an android device, currently running 4.0.3 Ice Cream Sandwich (cryptlife.com). The current version of Google Glass software out right now is XE 12, or explorer edition 12. This is the version that will be used for this research project.

Rooting Glass

In XE 12 there are permissions which prevent the investigator from accessing all directories on Google Glass. By rooting the device, these previously blocked directories can be accessed. For this research, it was important to examine a pair of rooted Google Glass so all data that can be extracted from the device is taken into account.

Android Developer Tools or ADT is a suite of tools which offer features to aid in Android development (developer.android.com). Using the tutorial found on the Google Developer website, a rooted image of the latest version of Google Glass XE 12, was downloaded and pushed to the device (developers.google.com/glass). By pushing a rooted version of XE 12 to the device, Glass runs on an image that has root permissions. The steps taken to root Glass are shown below.

Rooting Google Glass

1. After installing ADT tools on a Linux machine, run "adb reboot bootloader" to reboot into fastboot mode

2. Run "fastboot oem unlock"

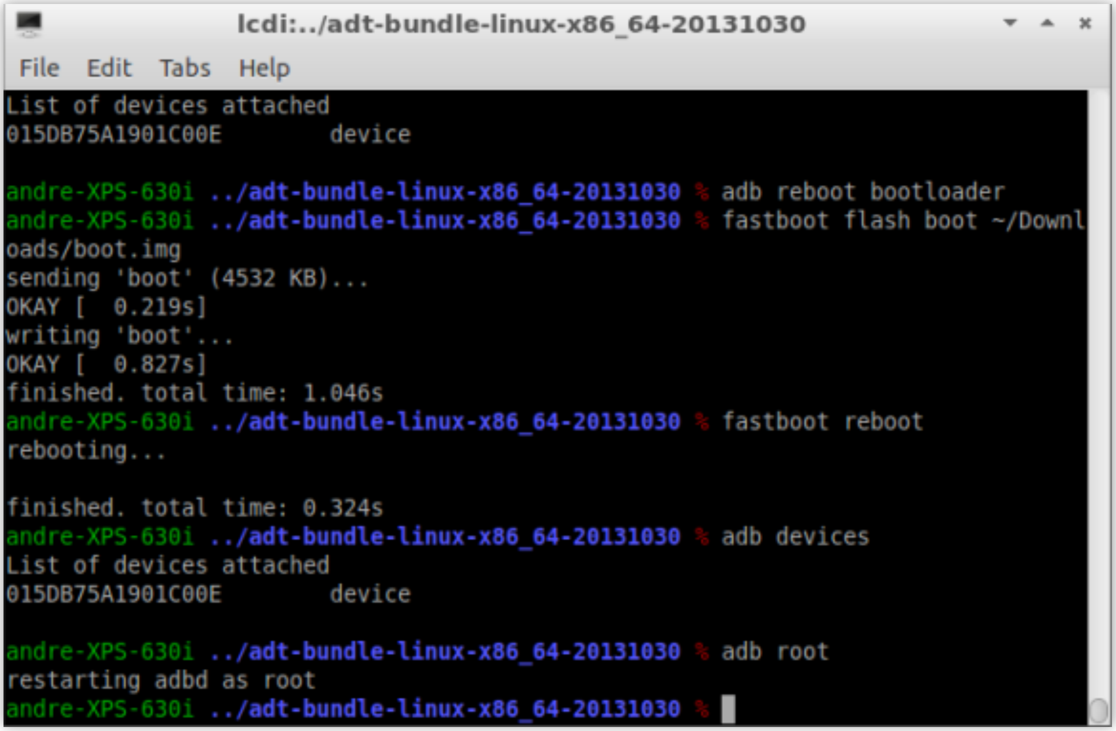
3. A warning message is given alerting Google Glass's warranty will be voided and all data would be erased. "Fastboot oem unlock" must be run again to confirm.

4. Run "fastboot flash boot 'boot.img'" which pushes the image onto the device.

5. Reboot the device with "fastboot reboot"

6. Run "adb root" restart into root.

A screenshot going through some of these steps is shown in Figure 3.



```
List of devices attached
015DB75A1901C00E        device

andre-XPS-630i ../adt-bundle-linux-x86_64-20131030 % adb reboot bootloader
andre-XPS-630i ../adt-bundle-linux-x86_64-20131030 % fastboot flash boot ~/Downl
oads/boot.img
sending 'boot' (4532 KB)...
OKAY [  0.219s]
writing 'boot'...
OKAY [  0.827s]
finished. total time: 1.046s
andre-XPS-630i ../adt-bundle-linux-x86_64-20131030 % fastboot reboot
rebooting...

finished. total time: 0.324s
andre-XPS-630i ../adt-bundle-linux-x86_64-20131030 % adb devices
List of devices attached
015DB75A1901C00E        device

andre-XPS-630i ../adt-bundle-linux-x86_64-20131030 % adb root
restarting adbd as root
andre-XPS-630i ../adt-bundle-linux-x86_64-20131030 %
```

Figure 3: Rooting Google Glass in Linux

Because rooting Google Glass wipes all the contents, by pushing a clean image with root permissions onto it, an investigator should not root Google Glass if it was not already rooted because they would lose all of the data. However, if a user has already followed the steps above and rooted their device, the examiner can get root access to the device.With a rooted pair of Google Glass, directories can be accessed that contain more information about the device, and therefore more forensically relevant artifacts can be acquired.

The Shattered Script

After Glass has been rooted, more data can be extracted from the device. The Shattered script is an acquisition tool that can extract data from Google Glass. By extracting the data from the device, the investigators can analyze the contents.

The main feature of Shattered is the automation of the ADT tool, ADB pull, which extracts files from the device (developer.android.com). Other features are available with Shattered, such as dumpsys services. This grabs data about current system service information running (source.android.com). The script was updated from v1.2 to v1.3 for this project, with the added ability to extract additional information from rooted pairs of Google Glass. The "adb root" function was added into the script so Glass would reboot as root before imaging. The root paths were also hardcoded into Glass, so it would manually pull each directory from root.

Time Related Artifacts on Google Glass

During analysis, forensic artifacts on Google Glass that can be correlated with a specific time will be examined. For this research, an image was made on a rooted Google Glass running XE 12and acquired with Shattered v1.3. All file paths referred to in this paper are based off the output structure from Shattered v1.3.

Google Glass Battery Information

Some of the most basic time related forensic artifacts found on Google Glass are located in *image/fs/system/dropbox*. This folder contains an assortment of text files, which provides information about the state of the battery at a particular time. Figure 4 shows the location of this directory.

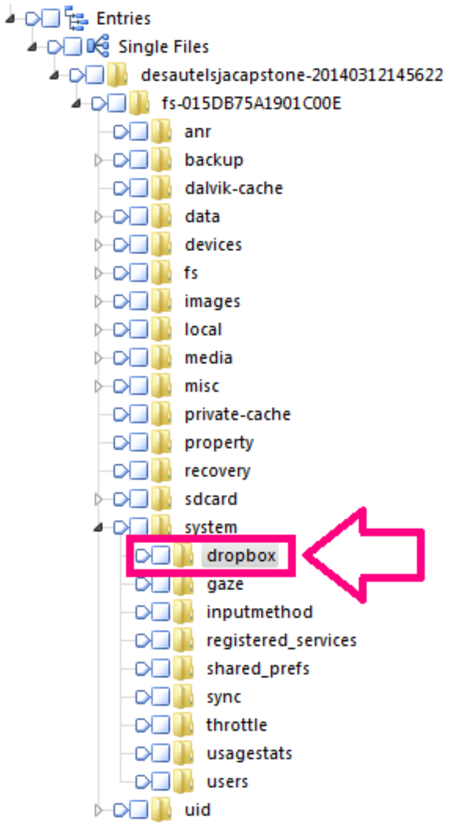


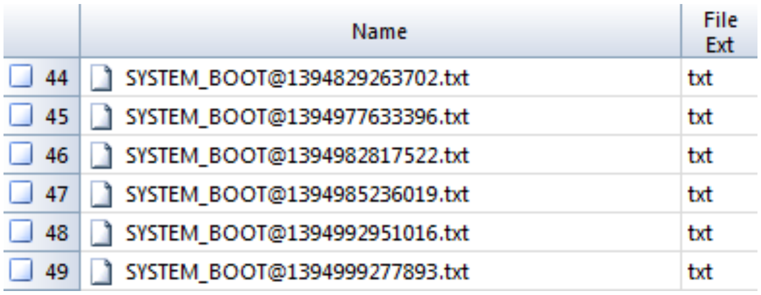
Figure 4: Image/fs/system/dropbox

Was Glass on or off at a specific time?

The “SYSTEM_BOOT@UNIXTIME.txt” files within this *image/fs/system/dropbox* directory is a text file which gives a Unix millisecond timestamp for each time Glass was powered on. Google Glass uses millisecond Unix times, which means the number of milliseconds since

January 1, 1970. Using a converter that has the capability to convert millisecond times to readable times can give the investigator a proper readable time.

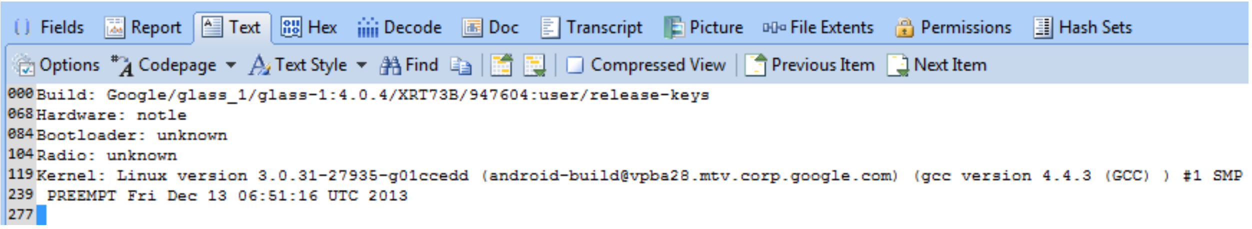
The Unix millisecond time given in the title after the "@" symbol is the time Glass was powered on. The most important part of this file is the title itself, which provides a timestamp for each power on. For example, looking at Figure 5, it is evident Google Glass was powered on six times because there are six "SYSTEM_BOOT@UNIXTIME.txt" files. In addition, after translating the millisecond Unix time after the "@" sign, it is clear the first time the device was powered on was March 14, 2014 at 4:34:23 PM in Eastern Time. Each of these times can be translated from Unix millisecond timestamps to readable times to see how many times Glass was powered on, and exactly what time each power on occurred.



	Name	File Ext
☐ 44	SYSTEM_BOOT@1394829263702.txt	txt
☐ 45	SYSTEM_BOOT@1394977633396.txt	txt
☐ 46	SYSTEM_BOOT@1394982817522.txt	txt
☐ 47	SYSTEM_BOOT@1394985236019.txt	txt
☐ 48	SYSTEM_BOOT@1394992951016.txt	txt
☐ 49	SYSTEM_BOOT@1394999277893.txt	txt

Figure 5: SYSTEM_BOOT@UNIXTIME.txt Files

While the title of these files are distinct with each time Google Glass was powered on, the contents of these files are not unique. All of the file contents appear to have the same information. As shown in Figure 6, each of the files list basic information such as the build and kernel of the device. After running all the "SYSTEM_BOOT@UNIXTIME.txt" files in MD5Summer, they all appeared to have the same MD5 hash value.



```
000 Build: Google/glass_1/glass-1:4.0.4/XRT73B/947604:user/release-keys
068 Hardware: notle
084 Bootloader: unknown
104 Radio: unknown
119 Kernel: Linux version 3.0.31-27935-g01ccedd (android-build@vpba28.mtv.corp.google.com) (gcc version 4.4.3 (GCC) ) #1 SMP
239  PREEMPT Fri Dec 13 06:51:16 UTC 2013
277
```

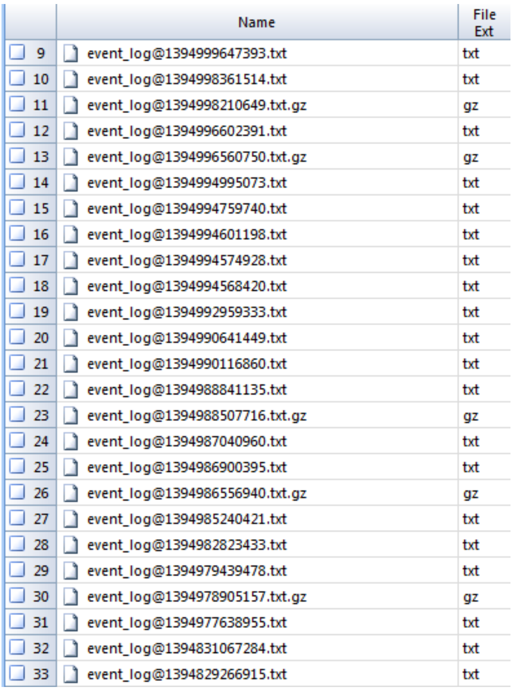
Figure 6: Content of every "SYSTEM_BOOT@UNIXTIME.txt" file

MD5 hashing uses an algorithm to create a 128 bit sum for each file. It creates a sum based on the content of the files, and can be used as a "fingerprint" to differentiate each file (cse.sc.edu). Because the MD5 sum is the same for each of the files, it confirms that although the titles of each file are different, the contents are the same.

Therefore, when examining "SYSTEM_BOOT@UNIXTIME.txt" files for timestamps, the investigator can get the relevant information about when the device was powered on right from the file name.

The *Image/fs/system/dropbox* directory also contains "event_log@UNIXTIME.txt" and "event_log@UNIXTIME.txt.gz" files as shown in Figure 7. The contents of these files contain information about the battery levels on Google Glass at a particular time.

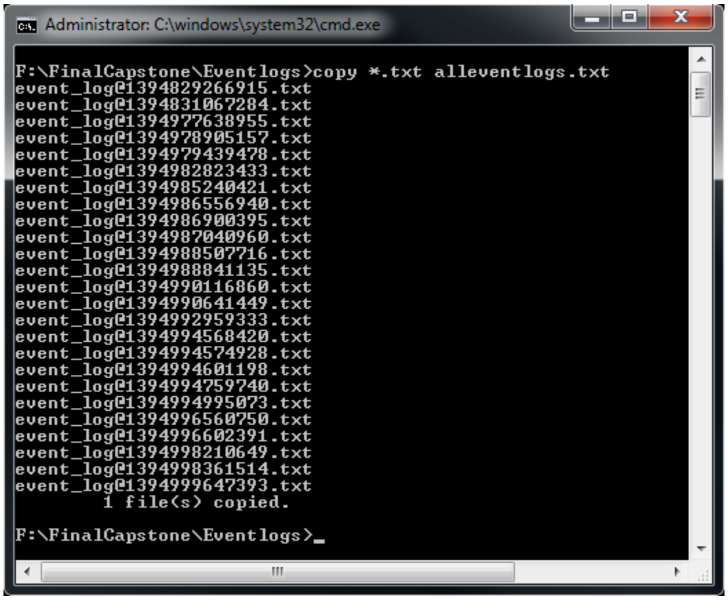
When these files were exported, there were no "event_log@UNIXTIME.txt" files larger than 4KB. After extracting the compressed "event_log@UNIXTIME.txt.gz" file, their contents were 5KB and higher. It appears when these files get larger then 5KB, they are compressed into an "event_log@UNIXTIME.txt.gz" file.



	Name	File Ext
☐ 9	event_log@1394999647393.txt	txt
☐ 10	event_log@1394998361514.txt	txt
☐ 11	event_log@1394998210649.txt.gz	gz
☐ 12	event_log@1394996602391.txt	txt
☐ 13	event_log@1394996560750.txt.gz	gz
☐ 14	event_log@1394994995073.txt	txt
☐ 15	event_log@1394994759740.txt	txt
☐ 16	event_log@1394994601198.txt	txt
☐ 17	event_log@1394994574928.txt	txt
☐ 18	event_log@1394994568420.txt	txt
☐ 19	event_log@1394992959333.txt	txt
☐ 20	event_log@1394990641449.txt	txt
☐ 21	event_log@1394990116860.txt	txt
☐ 22	event_log@1394988841135.txt	txt
☐ 23	event_log@1394988507716.txt.gz	gz
☐ 24	event_log@1394987040960.txt	txt
☐ 25	event_log@1394986900395.txt	txt
☐ 26	event_log@1394986556940.txt.gz	gz
☐ 27	event_log@1394985240421.txt	txt
☐ 28	event_log@1394982823433.txt	txt
☐ 29	event_log@1394979439478.txt	txt
☐ 30	event_log@1394978905157.txt.gz	gz
☐ 31	event_log@1394977638955.txt	txt
☐ 32	event_log@1394831067284.txt	txt
☐ 33	event_log@1394829266915.txt	txt

Figure 7: event_log@UNIXTIME.txt and event_log@UNIXTIME.txt.gz files

When examining these files for time information, it is best to combine them into a single file to make analysis simpler. After extracting the compressed "event_log@UNIXTIME.txt.gz" files, all the event logs were merged into a single file as shown in Figure 8.



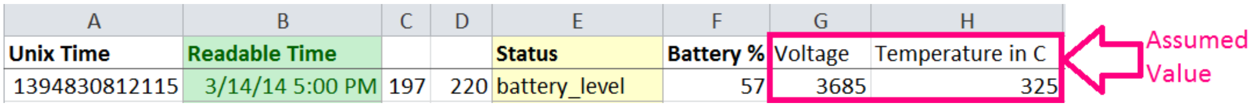
```
F:\FinalCapstone\Eventlogs>copy *.txt alleventlogs.txt
event_log@1394829266915.txt
event_log@1394831067284.txt
event_log@1394977638955.txt
event_log@1394978905157.txt
event_log@1394979439478.txt
event_log@1394982823433.txt
event_log@1394985240421.txt
event_log@1394986556940.txt
event_log@1394986900395.txt
event_log@1394987040960.txt
event_log@1394988507716.txt
event_log@1394988841135.txt
event_log@1394990116860.txt
event_log@1394990641449.txt
event_log@1394992959333.txt
event_log@1394994568420.txt
event_log@1394994574928.txt
event_log@1394994601198.txt
event_log@1394994759740.txt
event_log@1394994995073.txt
event_log@1394996560750.txt
event_log@1394996602391.txt
event_log@1394998210649.txt
event_log@1394998361514.txt
event_log@1394999647393.txt
        1 file(s) copied.

F:\FinalCapstone\Eventlogs>_
```

Figure 8: Merging the event_log@UNIXTIME.txt files

This single file of all the "event_log@UNIXTIME.txt" files was then imported into Microsoft Excel for further analysis. Note in this analysis, a column was inserted to convert the Unix timestamps to readable times. The formula, **=((((A1/1000)/60)/60)/24)+DATE(1970,1,1)+(-4/24)** takes into account Glass is recording times in Unix time in milliseconds, and allows for dates to be converted to the eastern time zone. The formula may have to be adjusted depending on the time zone. Also, that column must be set to a date and time format in Excel.

In addition to a column with the readable time being added, headers were added onto the Excel sheet to make the content easier to read. The last two columns, G and H, give numbers which could indicate the battery voltage and battery temperature. Google Glass uses a Li-ion battery. These batteries have temperature sensors in them, which could indicate how Glass is logging the temperature (Brain). Column H most likely indicates the temperature in Celsius, with the decimal point excluded. The temperature in of the battery at 5:00PM on 3/14/14 would be 32.5° Celsius according to Figure 9. Row G most likely shows the Voltage. Just like the temperature column, there is no decimal included. Therefore, it is likely the first entry in row G the voltage could indicate 3.685V. However, these are just assumed values.

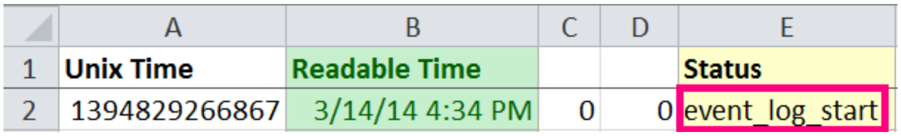


A	B	C	D	E	F	G	H
Unix Time	Readable Time			Status	Battery %	Voltage	Temperature in C
1394830812115	3/14/14 5:00 PM	197	220	battery_level	57	3685	325

Assumed Value

Figure 9: Voltage and Temperature Headers

Some information that can be obtained from this file is the time Glass was turned on. The “event_log_start” indicator shows every time Glass starts logging battery information, which happens when it is powered on. As shown in Figure 10, the file indicates Glass was powered on 3/14/14 at 4:34 PM. This can be cross referenced with the “SYSTEMBOOT@UNIXTIME.txt” files.



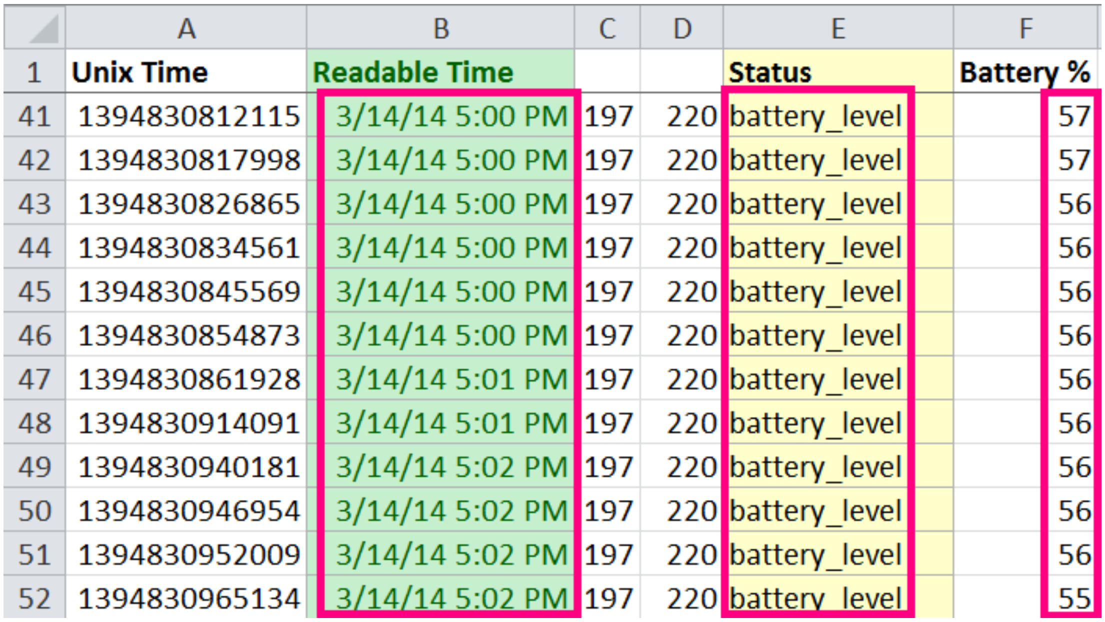
	A	B	C	D	E
1	Unix Time	Readable Time			Status
2	1394829266867	3/14/14 4:34 PM	0	0	event_log_start

Figure 10: event_log_start

This “event_log_start” timestamp can be cross referenced with the “SYSTEM_BOOT@UNIXTIME.txt” files to see what time Google Glass was powered on.

Examining Battery Drain Levels on Google Glass

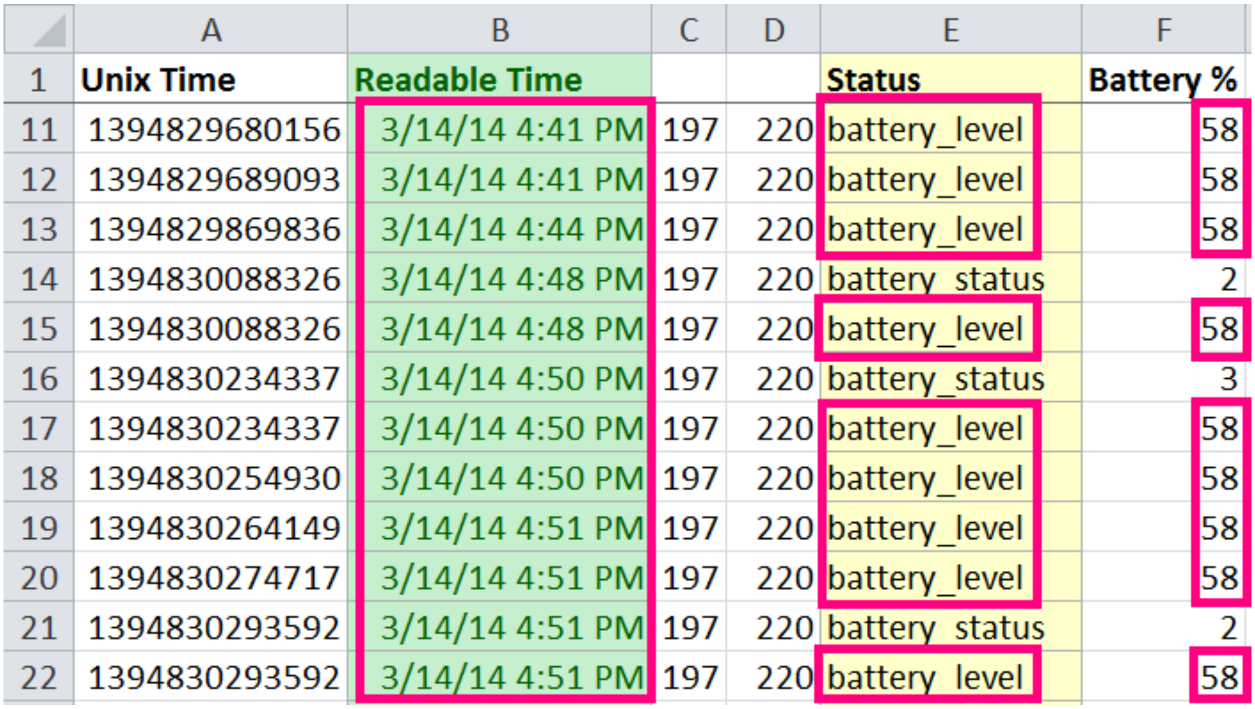
Not only can the “event_log@UNIXTIME.txt” files indicate when Google Glass was on or off, it can also give information about the battery level on the device at a specific time. Figure 11 shows the “battery_level” indicator which correlates with a specific Unix millisecond timestamp and battery percentage.



	A	B	C	D	E	F
1	Unix Time	Readable Time			Status	Battery %
41	1394830812115	3/14/14 5:00 PM	197	220	battery_level	57
42	1394830817998	3/14/14 5:00 PM	197	220	battery_level	57
43	1394830826865	3/14/14 5:00 PM	197	220	battery_level	56
44	1394830834561	3/14/14 5:00 PM	197	220	battery_level	56
45	1394830845569	3/14/14 5:00 PM	197	220	battery_level	56
46	1394830854873	3/14/14 5:00 PM	197	220	battery_level	56
47	1394830861928	3/14/14 5:01 PM	197	220	battery_level	56
48	1394830914091	3/14/14 5:01 PM	197	220	battery_level	56
49	1394830940181	3/14/14 5:02 PM	197	220	battery_level	56
50	1394830946954	3/14/14 5:02 PM	197	220	battery_level	56
51	1394830952009	3/14/14 5:02 PM	197	220	battery_level	56
52	1394830965134	3/14/14 5:02 PM	197	220	battery_level	55

Figure 11: "Battery_level" with timestamp and percentage

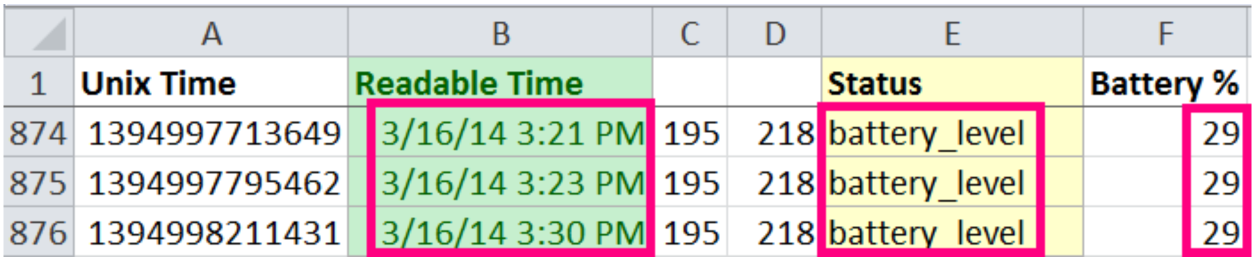
The "event_log@UNIXTIME" file is forensically relevant due to battery percentage information it can provide. It can indicate usage based on how fast or slow the battery is draining. Figure 12 shows a ten minute period of battery drain while Glass was sitting idle and connected to WiFi. This shows over a ten minute period of Google Glass being on but not in use, the battery does not even drop 1%. This can vary slightly depending on if the user looked up, which could have activated the, "head wake up" feature, which wakes Glass up and lights up the screen if the user tilts their head back (support.google.com) .



	A	B	C	D	E	F
1	**Unix Time**	**Readable Time**			**Status**	**Battery %**
11	1394829680156	3/14/14 4:41 PM	197	220	battery_level	58
12	1394829689093	3/14/14 4:41 PM	197	220	battery_level	58
13	1394829869836	3/14/14 4:44 PM	197	220	battery_level	58
14	1394830088326	3/14/14 4:48 PM	197	220	battery_status	2
15	1394830088326	3/14/14 4:48 PM	197	220	battery_level	58
16	1394830234337	3/14/14 4:50 PM	197	220	battery_status	3
17	1394830234337	3/14/14 4:50 PM	197	220	battery_level	58
18	1394830254930	3/14/14 4:50 PM	197	220	battery_level	58
19	1394830264149	3/14/14 4:51 PM	197	220	battery_level	58
20	1394830274717	3/14/14 4:51 PM	197	220	battery_level	58
21	1394830293592	3/14/14 4:51 PM	197	220	battery_status	2
22	1394830293592	3/14/14 4:51 PM	197	220	battery_level	58

Figure 12: Ten minute period of battery drain while Glass sat idle connected to WiFi

The battery drain levels on Google Glass did not change much over a ten minute period of time sitting idle when connected to a Nexus 5 using Bluetooth data. Similar to when it was connected to Wifi, the battery did not even drain 1%. Figure 13 shows the battery did not drain over this period of ten minutes. This shows the battery drain levels on Glass are not affected by its connection type, to either Wifi data or Bluetooth data.

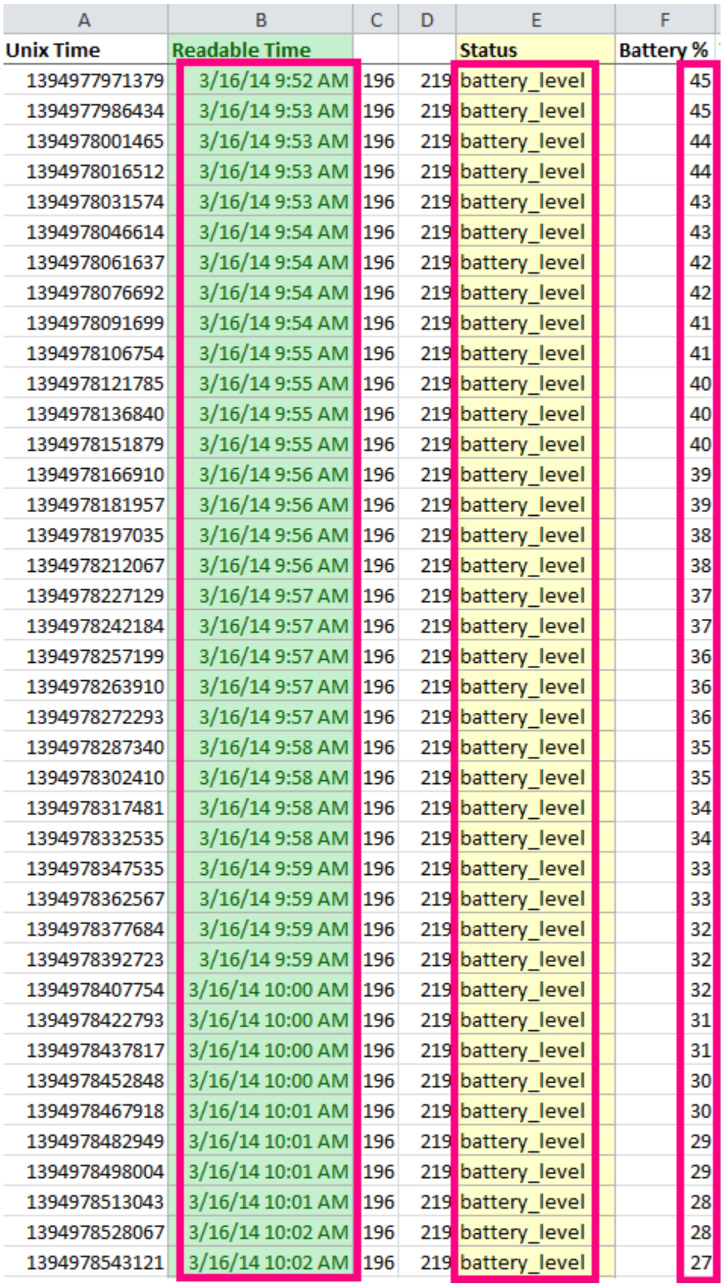


	A	B	C	D	E	F
1	Unix Time	Readable Time			Status	Battery %
874	1394997713649	3/16/14 3:21 PM	195	218	battery_level	29
875	1394997795462	3/16/14 3:23 PM	195	218	battery_level	29
876	1394998211431	3/16/14 3:30 PM	195	218	battery_level	29

Figure 13: Ten minute period of battery drain while Glass sat idle connected to Bluetooth

Although sitting idle does not have a huge impact on Google Glass battery drain, with more intensive activities on the device, the battery drains faster. When a user on Google Glass watches videos or receives GPS directions, the battery drains more quickly because the screen is on and lit up for a longer time.

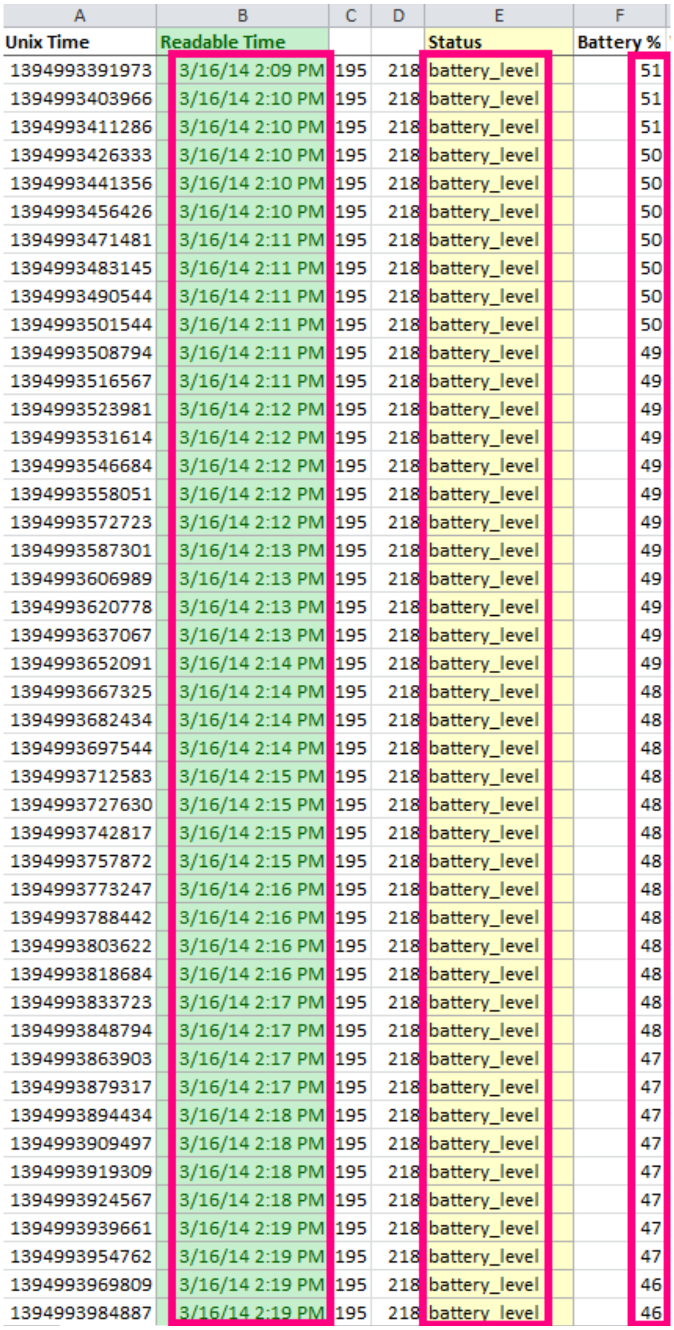
When watching a video on Google Glass, the battery drained at a much higher rate. This is because the screen is constantly on when the user watches a video. Figure 14 shows the battery draining over a ten minute period of time while a user was watching a video on Google Glass. Over ten minutes, the video drained 19%, which is a much more dramatic decrease then the idle battery drain rate.



A	B	C	D	E	F
Unix Time	Readable Time			Status	Battery %
1394977971379	3/16/14 9:52 AM	196	219	battery_level	45
1394977986434	3/16/14 9:53 AM	196	219	battery_level	45
1394978001465	3/16/14 9:53 AM	196	219	battery_level	44
1394978016512	3/16/14 9:53 AM	196	219	battery_level	44
1394978031574	3/16/14 9:53 AM	196	219	battery_level	43
1394978046614	3/16/14 9:54 AM	196	219	battery_level	43
1394978061637	3/16/14 9:54 AM	196	219	battery_level	42
1394978076692	3/16/14 9:54 AM	196	219	battery_level	42
1394978091699	3/16/14 9:54 AM	196	219	battery_level	41
1394978106754	3/16/14 9:55 AM	196	219	battery_level	41
1394978121785	3/16/14 9:55 AM	196	219	battery_level	40
1394978136840	3/16/14 9:55 AM	196	219	battery_level	40
1394978151879	3/16/14 9:55 AM	196	219	battery_level	40
1394978166910	3/16/14 9:56 AM	196	219	battery_level	39
1394978181957	3/16/14 9:56 AM	196	219	battery_level	39
1394978197035	3/16/14 9:56 AM	196	219	battery_level	38
1394978212067	3/16/14 9:56 AM	196	219	battery_level	38
1394978227129	3/16/14 9:57 AM	196	219	battery_level	37
1394978242184	3/16/14 9:57 AM	196	219	battery_level	37
1394978257199	3/16/14 9:57 AM	196	219	battery_level	36
1394978263910	3/16/14 9:57 AM	196	219	battery_level	36
1394978272293	3/16/14 9:57 AM	196	219	battery_level	36
1394978287340	3/16/14 9:58 AM	196	219	battery_level	35
1394978302410	3/16/14 9:58 AM	196	219	battery_level	35
1394978317481	3/16/14 9:58 AM	196	219	battery_level	34
1394978332535	3/16/14 9:58 AM	196	219	battery_level	34
1394978347535	3/16/14 9:59 AM	196	219	battery_level	33
1394978362567	3/16/14 9:59 AM	196	219	battery_level	33
1394978377684	3/16/14 9:59 AM	196	219	battery_level	32
1394978392723	3/16/14 9:59 AM	196	219	battery_level	32
1394978407754	3/16/14 10:00 AM	196	219	battery_level	32
1394978422793	3/16/14 10:00 AM	196	219	battery_level	31
1394978437817	3/16/14 10:00 AM	196	219	battery_level	31
1394978452848	3/16/14 10:00 AM	196	219	battery_level	30
1394978467918	3/16/14 10:01 AM	196	219	battery_level	30
1394978482949	3/16/14 10:01 AM	196	219	battery_level	29
1394978498004	3/16/14 10:01 AM	196	219	battery_level	29
1394978513043	3/16/14 10:01 AM	196	219	battery_level	28
1394978528067	3/16/14 10:02 AM	196	219	battery_level	28
1394978543121	3/16/14 10:02 AM	196	219	battery_level	27

Figure 14: Ten minute period of battery drain while user watched a video on Google Glass

The rate of battery drain also decreases faster when a user is receiving GPS directions using Google Glass. Figure 15 shows the user following GPS directions using Google Glass over a ten minute period of time. The battery level went down 5% over the course of ten minutes.



A	B	C	D	E	F
Unix Time	Readable Time			Status	Battery %
1394993391973	3/16/14 2:09 PM	195	218	battery_level	51
1394993403966	3/16/14 2:10 PM	195	218	battery_level	51
1394993411286	3/16/14 2:10 PM	195	218	battery_level	51
1394993426333	3/16/14 2:10 PM	195	218	battery_level	50
1394993441356	3/16/14 2:10 PM	195	218	battery_level	50
1394993456426	3/16/14 2:10 PM	195	218	battery_level	50
1394993471481	3/16/14 2:11 PM	195	218	battery_level	50
1394993483145	3/16/14 2:11 PM	195	218	battery_level	50
1394993490544	3/16/14 2:11 PM	195	218	battery_level	50
1394993501544	3/16/14 2:11 PM	195	218	battery_level	50
1394993508794	3/16/14 2:11 PM	195	218	battery_level	49
1394993516567	3/16/14 2:11 PM	195	218	battery_level	49
1394993523981	3/16/14 2:12 PM	195	218	battery_level	49
1394993531614	3/16/14 2:12 PM	195	218	battery_level	49
1394993546684	3/16/14 2:12 PM	195	218	battery_level	49
1394993558051	3/16/14 2:12 PM	195	218	battery_level	49
1394993572723	3/16/14 2:12 PM	195	218	battery_level	49
1394993587301	3/16/14 2:13 PM	195	218	battery_level	49
1394993606989	3/16/14 2:13 PM	195	218	battery_level	49
1394993620778	3/16/14 2:13 PM	195	218	battery_level	49
1394993637067	3/16/14 2:13 PM	195	218	battery_level	49
1394993652091	3/16/14 2:14 PM	195	218	battery_level	49
1394993667325	3/16/14 2:14 PM	195	218	battery_level	48
1394993682434	3/16/14 2:14 PM	195	218	battery_level	48
1394993697544	3/16/14 2:14 PM	195	218	battery_level	48
1394993712583	3/16/14 2:15 PM	195	218	battery_level	48
1394993727630	3/16/14 2:15 PM	195	218	battery_level	48
1394993742817	3/16/14 2:15 PM	195	218	battery_level	48
1394993757872	3/16/14 2:15 PM	195	218	battery_level	48
1394993773247	3/16/14 2:16 PM	195	218	battery_level	48
1394993788442	3/16/14 2:16 PM	195	218	battery_level	48
1394993803622	3/16/14 2:16 PM	195	218	battery_level	48
1394993818684	3/16/14 2:16 PM	195	218	battery_level	48
1394993833723	3/16/14 2:17 PM	195	218	battery_level	48
1394993848794	3/16/14 2:17 PM	195	218	battery_level	48
1394993863903	3/16/14 2:17 PM	195	218	battery_level	47
1394993879317	3/16/14 2:17 PM	195	218	battery_level	47
1394993894434	3/16/14 2:18 PM	195	218	battery_level	47
1394993909497	3/16/14 2:18 PM	195	218	battery_level	47
1394993919309	3/16/14 2:18 PM	195	218	battery_level	47
1394993924567	3/16/14 2:18 PM	195	218	battery_level	47
1394993939661	3/16/14 2:19 PM	195	218	battery_level	47
1394993954762	3/16/14 2:19 PM	195	218	battery_level	47
1394993969809	3/16/14 2:19 PM	195	218	battery_level	46
1394993984887	3/16/14 2:19 PM	195	218	battery_level	46

Figure 15: Ten minute period of battery drain while user received GPS directions on Google Glass

While battery levels cannot provide a definite indicator to what a user was doing on Google Glass at a particular time, it can support other time artifacts on the device. Battery levels can be cross referenced with other timestamps on Google Glass to further confirm or dispute a particular user activity.

Saved Voice Command Information

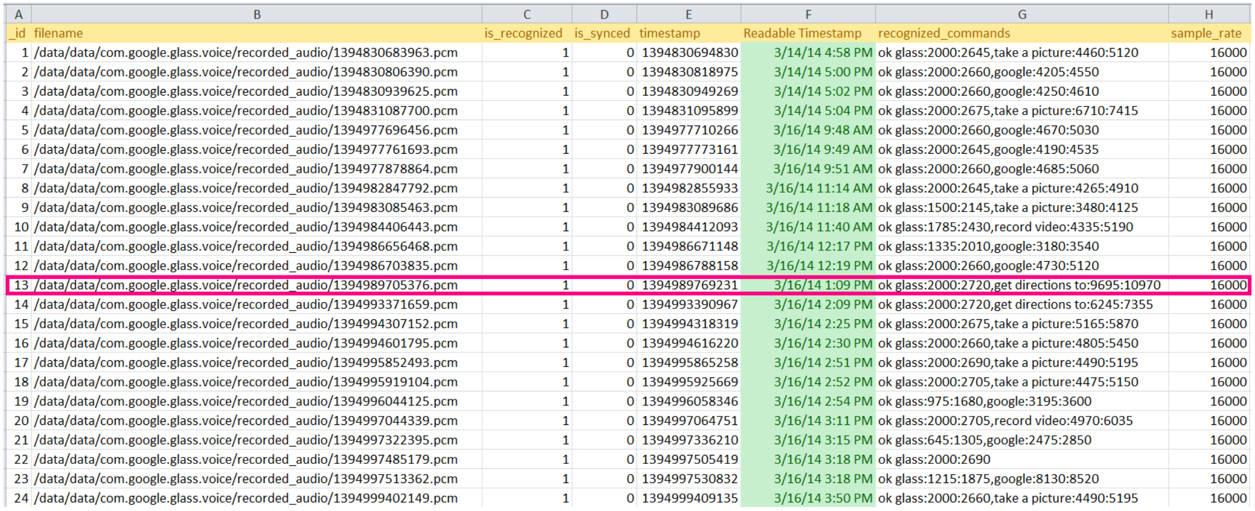
Within the *Image/fs/data/com.google.glass.voice* directory is information about the hands free voice commands on Google Glass. On Glass, users can speak and use voice commands to control the device. When the device displays the home menu screen, the user can say, “Ok Glass,” to request different commands. There are many options the user can choose from, such as taking a picture, recording a video, Google searching something, messaging someone, making a phone call, or asking for GPS directions.

Logged Voice Commands

The “saved_audio” file is a SQLite 3 database which holds information about Google Glass voice commands. This file is stored in *Image/fs/data/com.google.glass.voice/databases/saved_audio*. For analysis, the “saved_audio” table of this SQLite database was imported into Excel. Excel was used so a column could be added to convert the Unix millisecond timestamps from this database into readable timestamps. The formula used is, **=((((E2/1000)/60)/60)/24)+DATE(1970,1,1)+(-4/24)**

Figure 16 shows the “saved_audio” database uploaded into Excel. Looking at entry 13 which has been marked in the picture, the database gives the original file path where the voice file is stored on Google Glass in column B. This “filename” shows the original path of the file, which is slightly different then the output results of the Shattered script. This is because of the way the script extracts and organizes data output. It also shows the Unix millisecond timestamp in when the voice command was spoken, in column E. Column F was added in, using the formula explained above. This provides the readable version of the Unix millisecond time. In this example, the date and time of the spoken command was 3/16/14 1:09PM. The actual commands the user requested can be seen in column G. For this example, it is clear the user said, “Ok Glass, Get directions to”.

Note the entire command is not shown under, “recognized commands”. Although it is clear the user asked for directions to somewhere, it is not clear where they asked to go. For voice commands such as Google searching and asking for directions, which requires Glass to use WiFi or Bluetooth data, the voice command is cut off after the initial request is made. However, voice commands which do not require connectivity on Glass, such as take a picture or record a video, store the entire voice command.



A	B	C	D	E	F	G	H
_id	filename	is_recognized	is_synced	timestamp	Readable Timestamp	recognized_commands	sample_rate
1	/data/data/com.google.glass.voice/recorded_audio/1394830683963.pcm	1	0	1394830694830	3/14/14 4:58 PM	ok glass:2000:2645,take a picture:4460:5120	16000
2	/data/data/com.google.glass.voice/recorded_audio/1394830806390.pcm	1	0	1394830818975	3/14/14 5:00 PM	ok glass:2000:2660,google:4205:4550	16000
3	/data/data/com.google.glass.voice/recorded_audio/1394830939625.pcm	1	0	1394830949269	3/14/14 5:02 PM	ok glass:2000:2660,google:4250:4610	16000
4	/data/data/com.google.glass.voice/recorded_audio/1394831087700.pcm	1	0	1394831095899	3/14/14 5:04 PM	ok glass:2000:2675,take a picture:6710:7415	16000
5	/data/data/com.google.glass.voice/recorded_audio/1394977696456.pcm	1	0	1394977710266	3/16/14 9:48 AM	ok glass:2000:2660,google:4670:5030	16000
6	/data/data/com.google.glass.voice/recorded_audio/1394977761693.pcm	1	0	1394977773161	3/16/14 9:49 AM	ok glass:2000:2645,google:4190:4535	16000
7	/data/data/com.google.glass.voice/recorded_audio/1394977878864.pcm	1	0	1394977900144	3/16/14 9:51 AM	ok glass:2000:2660,google:4685:5060	16000
8	/data/data/com.google.glass.voice/recorded_audio/1394982847792.pcm	1	0	1394982855933	3/16/14 11:14 AM	ok glass:2000:2645,take a picture:4265:4910	16000
9	/data/data/com.google.glass.voice/recorded_audio/1394983085463.pcm	1	0	1394983089686	3/16/14 11:18 AM	ok glass:1500:2145,take a picture:3480:4125	16000
10	/data/data/com.google.glass.voice/recorded_audio/1394984406443.pcm	1	0	1394984412093	3/16/14 11:40 AM	ok glass:1785:2430,record video:4335:5190	16000
11	/data/data/com.google.glass.voice/recorded_audio/1394986656468.pcm	1	0	1394986671148	3/16/14 12:17 PM	ok glass:1335:2010,google:3180:3540	16000
12	/data/data/com.google.glass.voice/recorded_audio/1394986703835.pcm	1	0	1394986788158	3/16/14 12:19 PM	ok glass:2000:2660,google:4730:5120	16000
13	/data/data/com.google.glass.voice/recorded_audio/1394989705376.pcm	1	0	1394989769231	3/16/14 1:09 PM	ok glass:2000:2720,get directions to:9695:10970	16000
14	/data/data/com.google.glass.voice/recorded_audio/1394993371659.pcm	1	0	1394993390967	3/16/14 2:09 PM	ok glass:2000:2720,get directions to:6245:7355	16000
15	/data/data/com.google.glass.voice/recorded_audio/1394994307152.pcm	1	0	1394994318319	3/16/14 2:25 PM	ok glass:2000:2675,take a picture:5165:5870	16000
16	/data/data/com.google.glass.voice/recorded_audio/1394994601795.pcm	1	0	1394994616220	3/16/14 2:30 PM	ok glass:2000:2660,take a picture:4805:5450	16000
17	/data/data/com.google.glass.voice/recorded_audio/1394995852493.pcm	1	0	1394995865258	3/16/14 2:51 PM	ok glass:2000:2690,take a picture:4490:5195	16000
18	/data/data/com.google.glass.voice/recorded_audio/1394995919104.pcm	1	0	1394995925669	3/16/14 2:52 PM	ok glass:2000:2705,take a picture:4475:5150	16000
19	/data/data/com.google.glass.voice/recorded_audio/1394996044125.pcm	1	0	1394996058346	3/16/14 2:54 PM	ok glass:975:1680,google:3195:3600	16000
20	/data/data/com.google.glass.voice/recorded_audio/1394997044339.pcm	1	0	1394997064751	3/16/14 3:11 PM	ok glass:2000:2705,record video:4970:6035	16000
21	/data/data/com.google.glass.voice/recorded_audio/1394997322395.pcm	1	0	1394997336210	3/16/14 3:15 PM	ok glass:645:1305,google:2475:2850	16000
22	/data/data/com.google.glass.voice/recorded_audio/1394997485179.pcm	1	0	1394997505419	3/16/14 3:18 PM	ok glass:2000:2690	16000
23	/data/data/com.google.glass.voice/recorded_audio/1394997513362.pcm	1	0	1394997530832	3/16/14 3:18 PM	ok glass:1215:1875,google:8130:8520	16000
24	/data/data/com.google.glass.voice/recorded_audio/1394999402149.pcm	1	0	1394999409135	3/16/14 3:50 PM	ok glass:2000:2660,take a picture:4490:5195	16000

Figure 16: "saved_audio" database in Excel

Playing Back Raw Voice Files

Within *Image/fs/data/com.google.glass.voice/recorded_audio* are "UNIXTIME.pcm" files.

The "UNIXTIME.pcm" files are raw audio files of the user voice commands made on Google Glass. ".Pcm" stands for pulse code modulation, which is a way analog data is represented in a digital form (Rouse). These files can be examined by importing them into Audacity as raw data. When importing in Audacity, there is an option to change the sample rate. Going back to the "saved_audio" SQLite database, the sample rate is listed as 16,000.

Once this file is imported, it can be played back to hear the actual user's voice command at a specific time. This can be cross referenced back to the, "saved_audio" database, so the examiner can confirm the time for that specific request is the same, as well as hear the actual audio of the request. Voices are unique identifiers, as unique as a fingerprint or an eye scan (Authentify). This can put the user at the device at a specific time, and also tell what they were doing.

Figure 17 shows a "UNIXTIME.pcm" file imported into Audacity for playback. The, "Ok Glass," request was made two seconds into the file. This means if there was any background noise occurring prior to the, "Ok Glass," request, it would be heard in the playback of this file. The actual voice command, in this case, "get directions to," began 9.5 seconds into the file. Any background noise in between saying, "Ok Glass," and, "get directions to" can also be played back and analyzed as well.

Just like the, "saved_audio" file, voice commands that require Glass to reach out for information using WiFi or Bluetooth such as Google searches and GPS directions are cut off after the initial command. The user asked to get directions at second 9.7 in the audio clip below, but it cuts off before it is apparent where they asked for directions to.

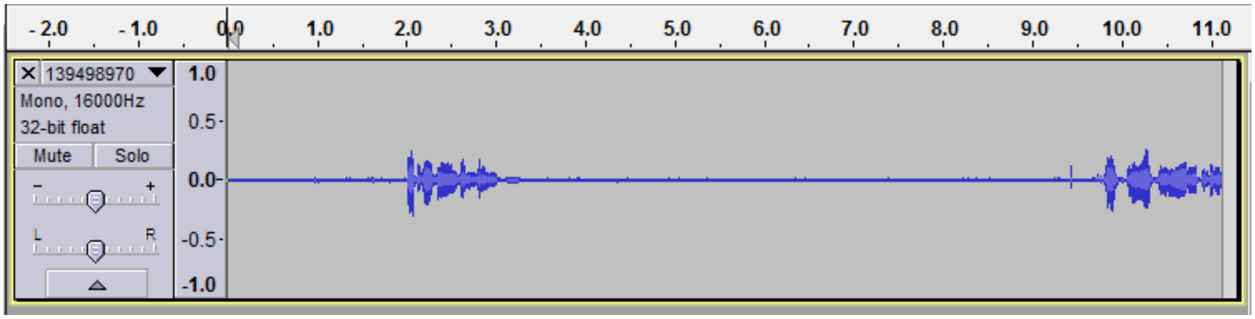


Figure 17: “UNIXTIME.pcm” file in Audacity

Google Glass can only interpret recognized commands, such as, “Ok Glass”. These interpreted commands are what get logged into the, “saved_audio” database. However, after looking into the, “saved_audio” database further, it is clear it accounts to an extent for this unrecognized audio. Under the, “recognized_commands” column, the numbers given separated by a, “:” shows the millisecond the voice command starts and ends. For an examiner, they could quickly look at the “recognized_commands” column and determine how long it took for a user to make a voice command, and therefore how likely it is for there to be useful background noise. Longer voice command files usually have more clear background noise in between the initial “Ok Glass” and the command.

The Google Glass Timeline

Google Glass uses a display feature called the, “timeline”. On this, “timeline” are various, “cards” with information on them. Applications, such as the New York Times, can use these, “cards” to show recent new stories. “Cards” can also be to things such as website results from Google searches, recent pictures taken, and other user activity.

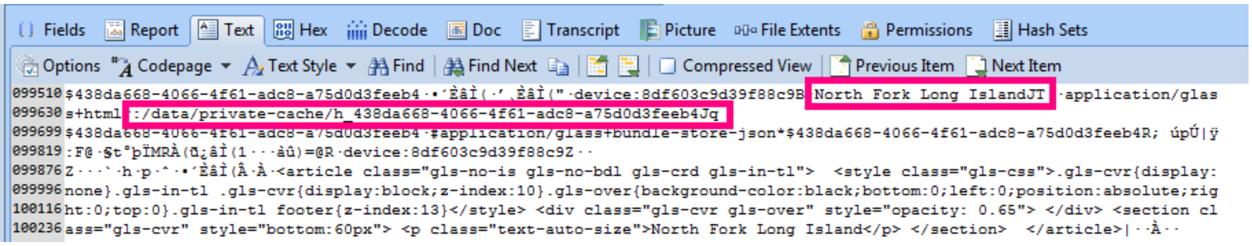
The *Image/fs/data/com.google.glass.home/databases/timeline.db* is a database of the “cards” on the Google Glass “timeline”. The way this artifact should be examined changes based on what the examiner is looking for.

Google Searches

If an examiner is looking for Google searches the timeline.db is a good place to check. Google searches can be found in the “timeline.db” when it is opened in EnCase using the “text view”. When looking through the document text, the examiner can search for, “JT” to find what the user Google searched for. “JT” may stand for JSON transformation, which is used to reformat data (ibm.com). Because Google Glass reformats web data and puts them on “cards” in the “timeline”, it is likely “JT” stands for JSON transformation. Investigators can use the “find” feature in EnCase text view to search through the file and find all the entries that end with, “JT” to see what the user Google searched for.

Figure 18 shows the “JT” search indicator in the “timeline.db” text view. After the search term is a path to a file in a directory called “private-cache” which will be explained later. The file shown, in this example is “h_438da668-4066-4f61-adc8-a75d0d3feeb4Jq”. This string of text is a GUID or globally unique identifier, which can be used in the database view later (msdn.microsoft.com)

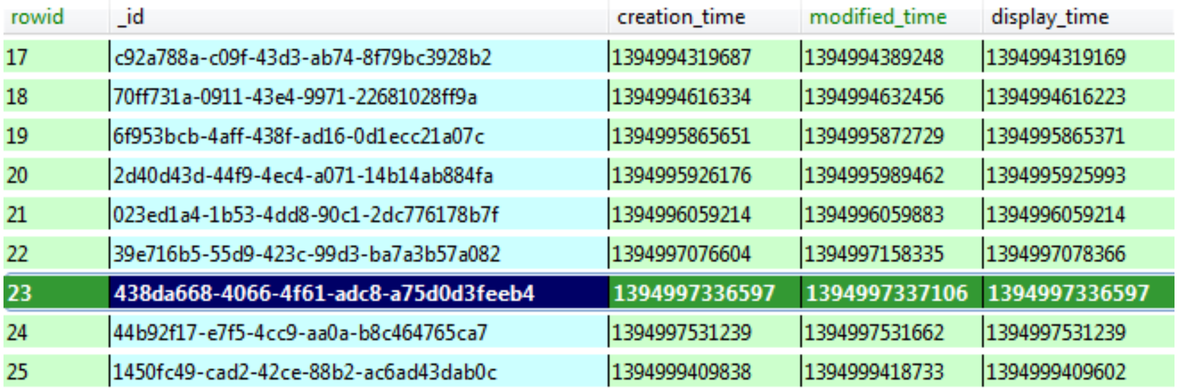
.



```
099510 $438da668-4066-4f61-adc8-a75d0d3feeb4·•´ÈâÌ(·´¸ÈâÌ("·device:8df603c9d39f88c9BNorth Fork Long IslandJT ·application/glas
099630 s+html :/data/private-cache/h_438da668-4066-4f61-adc8-a75d0d3feeb4Jq
099699 $438da668-4066-4f61-adc8-a75d0d3feeb4·#application/glass+bundle-store-json*$438da668-4066-4f61-adc8-a75d0d3feeb4R; úpÚ|ÿ
099819 :F@·$t°þÏMRÀ(ũ¿âÌ(1···âû)=@R·device:8df603c9d39f88c9Z··
099876 Z···`·h·p·^·•´ÈâÌ(À·À·<article class="gls-no-is gls-no-bdl gls-crd gls-in-tl">  <style class="gls-css">.gls-cvr{display:
099996 none}.gls-in-tl .gls-cvr{display:block;z-index:10}.gls-over{background-color:black;bottom:0;left:0;position:absolute;rig
100116 ht:0;top:0}.gls-in-tl footer{z-index:13}</style> <div class="gls-cvr gls-over" style="opacity: 0.65"> </div> <section cl
100236 ass="gls-cvr" style="bottom:60px"> <p class="text-auto-size">North Fork Long Island</p> </section>  </article>|··À··
```

Figure 18: Google Search in the "timeline.db" in EnCase text view

After importing, “timeline.db” into Firefox SQLite Manager and looking at the “timeline” table, there is a column called, “_id”. Going back to the file path after the Google search in the text view of the database, the GUID from, “h_438da668-4066-4f61-adc8-a75d0d3feeb4Jq” correlates with the “_id” column and is used to identify that particular search term. Figure 19 shows the GUID of the Google search in Firefox SQLite Manager.



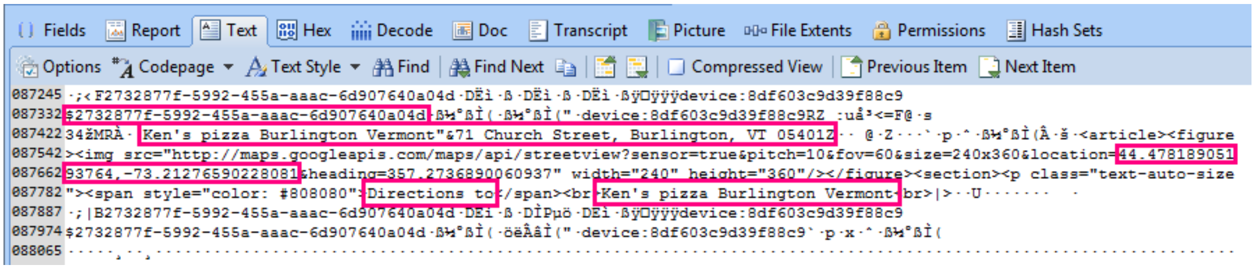
rowid	_id	creation_time	modified_time	display_time
17	c92a788a-c09f-43d3-ab74-8f79bc3928b2	1394994319687	1394994389248	1394994319169
18	70ff731a-0911-43e4-9971-22681028ff9a	1394994616334	1394994632456	1394994616223
19	6f953bcb-4aff-438f-ad16-0d1ecc21a07c	1394995865651	1394995872729	1394995865371
20	2d40d43d-44f9-4ec4-a071-14b14ab884fa	1394995926176	1394995989462	1394995925993
21	023ed1a4-1b53-4dd8-90c1-2dc776178b7f	1394996059214	1394996059883	1394996059214
22	39e716b5-55d9-423c-99d3-ba7a3b57a082	1394997076604	1394997158335	1394997078366
23	438da668-4066-4f61-adc8-a75d0d3feeb4	1394997336597	1394997337106	1394997336597
24	44b92f17-e7f5-4cc9-aa0a-b8c464765ca7	1394997531239	1394997531662	1394997531239
25	1450fc49-cad2-42ce-88b2-ac6ad43dab0c	1394999409838	1394999418733	1394999409602

Figure 19: "_id" of Google Search in SQLite Manager

With the GUID to identify a particular artifact in the timeline.db, timestamps can be identified. This database gives a creation, modified, and display time in Unix millisecond format. When translated into readable time, the creation and display times were shown at 3/16/2014 at 3:15:36 PM GMT -4. The modified time was listed as a millisecond later, at 3/16/2014 at 3:15:37 PM GMT -4. In this case, from a basic Google search, the creation, modified, and display times are generally the same. However, if something is deleted from “timeline.db”, these times are different, as explained in the next section.

Getting Directions

The “timeline.db” also stores information about GPS directions. When a user asks for directions on Glass, the history of this search can be found on a card on the timeline. If a user asks for GPS directions and cancelled the directions before arrival, the request can still be found in the text view of the “timeline.db” by searching for “Directions to”. This “Directions to” indicator is shown in Figure 20 can help and examiner quickly search through the “timeline.db” in the EnCase text view to find all the places the user requested directions to but did not arrive.

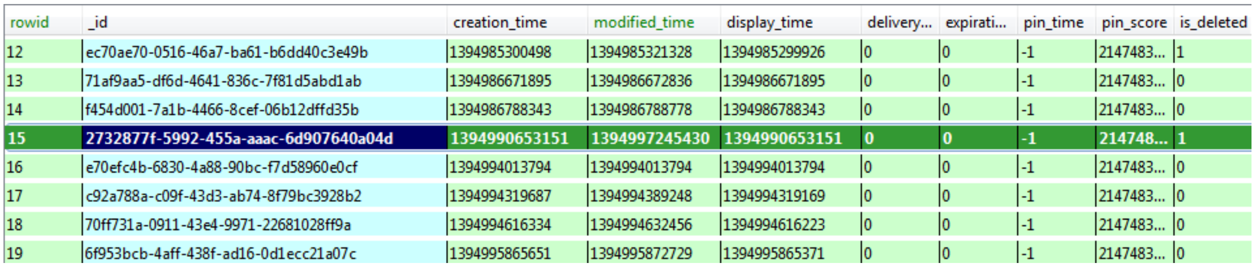


```
087245 ·;< F2732877f-5992-455a-aaac-6d907640a04d·DËì·ß·DËì·ß·DËì·ßÿ□ÿÿÿdevice:8df603c9d39f88c9
087332 $2732877f-5992-455a-aaac-6d907640a04d·ß¼°ßÌ(·ß¼°ßÌ("·device:8df603c9d39f88c9RZ :uå³<=F@·s
087422 34žMRÀ· Ken's pizza Burlington Vermont"&71 Church Street, Burlington, VT 05401Z·· @·Z···`·p·^·ß¼°ßÌ(Â·š·<article><figure
087542 ><img src="http://maps.googleapis.com/maps/api/streetview?sensor=true&pitch=10&fov=60&size=240x360&location=44.478189051
087662 93764,-73.21276590228081&heading=357.2736890060937" width="240" height="360"/></figure><section><p class="text-auto-size
087782 "><span style="color: #808080">Directions to</span><br>Ken's pizza Burlington Vermont<br>|>··U·······  ·
087887 ·;|B2732877f-5992-455a-aaac-6d907640a04d·DËì·ß·DÌPµö·DËì·ßÿ□ÿÿÿdevice:8df603c9d39f88c9
087974 $2732877f-5992-455a-aaac-6d907640a04d·ß¼°ßÌ(·öëÂâÌ("·device:8df603c9d39f88c9`·p·x·^·ß¼°ßÌ(
088065 ·····¸··¸·······················································································
```

Figure 20: Directions in the "timeline.db" in EnCase text view if the user requested directions but did not arrive

This view gives the name of the place Google Glass was getting directions to, the address, and the GPS coordinates of that location. Exactly what the user searched for indicated by the <br>SEARCH TERM<br> after the “Directions to” indicator. The “$” symbol displays the id that can be used to identify the artifact in Firefox SQLite Manager.

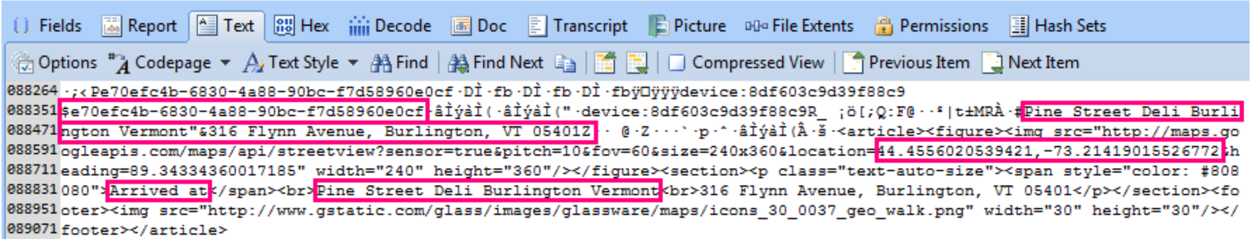
The time the user requested these directions can be found by opening up the “timeline.db” in Firefox SQLite Manager. As shown in Figure 21, the GUID that was identified by the “$” in the text view relates that specific artifact with the timestamp. The creation and display time confirm these directions were requested to be stopped, at 3/16/2014 1:24:13 PM GMT -4. However in this case, the modified time is different, 3/16/2014 3:14:05 PM GMT -4. This is because the Ken’s pizza GPS directions “card” was deleted from the timeline at the modified time. This is confirmed by the “1” in the “is_deleted” column rather than a “0”. “1” indicates a deleted file in the timeline.db. This does not just apply to GPS direction “cards” on the timeline being deleted. Other Google Glass timeline cards are indicated by a “1” in the “is_deleted” column as well as the “modified_time” changing to the time of deletion.



rowid	_id	creation_time	modified_time	display_time	delivery...	expirati...	pin_time	pin_score	is_deleted
12	ec70ae70-0516-46a7-ba61-b6dd40c3e49b	1394985300498	1394985321328	1394985299926	0	0	-1	2147483...	1
13	71af9aa5-df6d-4641-836c-7f81d5abd1ab	1394986671895	1394986672836	1394986671895	0	0	-1	2147483...	0
14	f454d001-7a1b-4466-8cef-06b12dffd35b	1394986788343	1394986788778	1394986788343	0	0	-1	2147483...	0
15	2732877f-5992-455a-aaac-6d907640a04d	1394990653151	1394997245430	1394990653151	0	0	-1	214748...	1
16	e70efc4b-6830-4a88-90bc-f7d58960e0cf	1394994013794	1394994013794	1394994013794	0	0	-1	2147483...	0
17	c92a788a-c09f-43d3-ab74-8f79bc3928b2	1394994319687	1394994389248	1394994319169	0	0	-1	2147483...	0
18	70ff731a-0911-43e4-9971-22681028ff9a	1394994616334	1394994632456	1394994616223	0	0	-1	2147483...	0
19	6f953bcb-4aff-438f-ad16-0d1ecc21a07c	1394995865651	1394995872729	1394995865371	0	0	-1	2147483...	0

Figure 21:"Timeline.db" in SQLite Manager displaying time directions were stopped and when the card was deleted

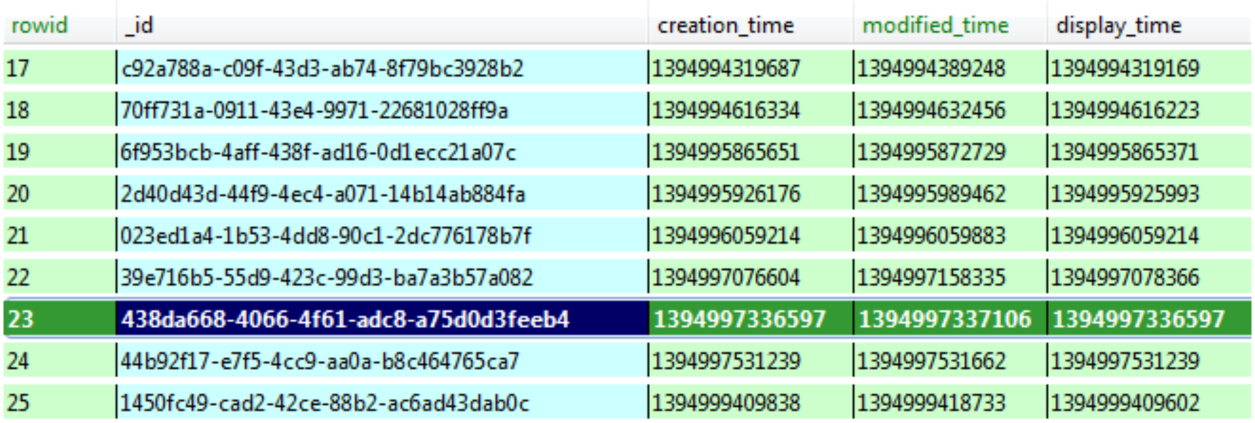
When a user asks for directions and arrives at their location, it appears slightly different in the text view of the timeline.db in EnCase. The same information is available for locations a user has arrived at, such as the address, GPS coordinates, and what the user searched for when asking for GPS directions. Also, the GUID of this artifact is still represented by the "$" symbol. However, instead of using the "Find" feature in the text view of EnCase to search for "Directions to", the investigator must such for the phrase, "Arrived At". This can confirm a person requested directions and arrived at a specific location. Figure 22 shows this "Arrived At" indicator in the EnCase text view.



```
088264 ·;‹ Pe70efc4b-6830-4a88-90bc-f7d58960e0cf·DÌ·fb·DÌ·fb·DÌ·fbÿ□ÿÿÿdevice:8df603c9d39f88c9
088351 $e70efc4b-6830-4a88-90bc-f7d58960e0cf·âÌýàÌ(·âÌýàÌ("·device:8df603c9d39f88c9R_ ¡ö[;Q:F@··ª|t±MRÀ·#Pine Street Deli Burli
088471 ngton Vermont"&316 Flynn Avenue, Burlington, VT 05401Z · @·Z···`·p·^·âÌýàÌ(Â·š·<article><figure><img src="http://maps.go
088591 ogleapis.com/maps/api/streetview?sensor=true&pitch=10&fov=60&size=240x360&location=44.4556020539421,-73.21419015526772&h
088711 eading=89.34334360017185" width="240" height="360"/></figure><section><p class="text-auto-size"><span style="color: #808
088831 080">Arrived at</span><br>Pine Street Deli Burlington Vermont<br>316 Flynn Avenue, Burlington, VT 05401</p></section><fo
088951 oter><img src="http://www.gstatic.com/glass/images/glassware/maps/icons_30_0037_geo_walk.png" width="30" height="30"/></
089071 footer></article>
```

Figure 22: Directions in the "timeline.db" in EnCase text view if the user requested directions and arrived

Opening up the "timeline.db" in Firefox SQLite Manager, this artifact can be looked up by the GUID, which is indicated by the "$" symbol in the text view. As shown in Figure 23, the created, modified, and display times indicate the user arrived at the location at 3/16/2014 at 2:20:13 PM GMT-4. This can put a user at a specific place at a specific time.

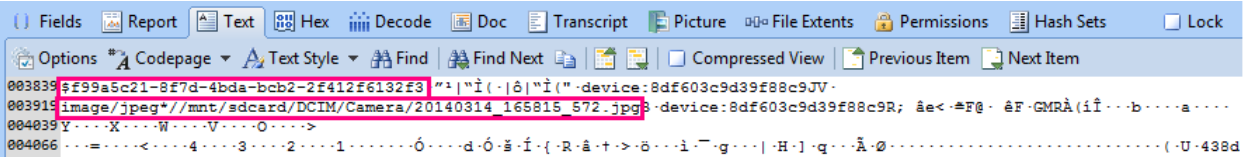


rowid	_id	creation_time	modified_time	display_time
17	c92a788a-c09f-43d3-ab74-8f79bc3928b2	1394994319687	1394994389248	1394994319169
18	70ff731a-0911-43e4-9971-22681028ff9a	1394994616334	1394994632456	1394994616223
19	6f953bcb-4aff-438f-ad16-0d1ecc21a07c	1394995865651	1394995872729	1394995865371
20	2d40d43d-44f9-4ec4-a071-14b14ab884fa	1394995926176	1394995989462	1394995925993
21	023ed1a4-1b53-4dd8-90c1-2dc776178b7f	1394996059214	1394996059883	1394996059214
22	39e716b5-55d9-423c-99d3-ba7a3b57a082	1394997076604	1394997158335	1394997078366
23	438da668-4066-4f61-adc8-a75d0d3feeb4	1394997336597	1394997337106	1394997336597
24	44b92f17-e7f5-4cc9-aa0a-b8c464765ca7	1394997531239	1394997531662	1394997531239
25	1450fc49-cad2-42ce-88b2-ac6ad43dab0c	1394999409838	1394999418733	1394999409602

Figure 23:"Timeline.db" in SQLite Manager displaying time user arrived at destination

Taking Pictures and Recording Videos

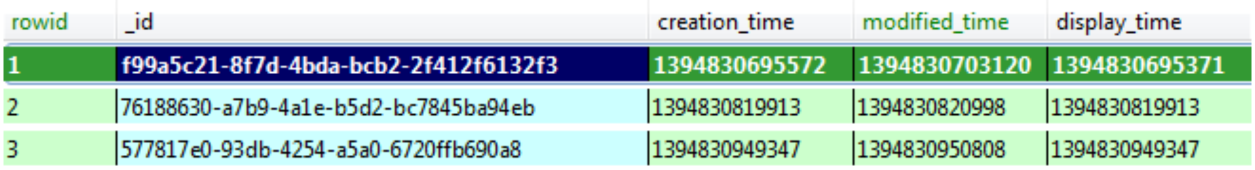
When a user takes a picture or records a video on Google Glass, it shows up in the “timeline” on a “card”. When looking at the text view of “timeline.db”, by searching for, “mnt/sdcard/DCIM/Camera”, the investigator can find pictures and videos which were on the Google Glass timeline. The screen shot below shows evidence of a picture on the Google Glass timeline in the text view of EnCase. This is indicated as a picture by the “.jpg” file format. The id for this file is indicated by the “$” symbol, in this case it is “f99a5c21-8f7d-4bda-bcb2-2f412f6132f3”, as displayed in Figure 24.



```
003839 $f99a5c21-8f7d-4bda-bcb2-2f412f6132f3 ·"¹|"Ì(·|ô|"Ì(" ·device:8df603c9d39f88c9JV·
003919 image/jpeg*//mnt/sdcard/DCIM/Camera/20140314_165815_572.jpg3 ·device:8df603c9d39f88c9R; âe<·≐F@· êF·GMRÀ(íÎ···b····a····
004039 Y····X····W····V····O····>
004066 ···=····<····4····3····2····1·······Ó····d·Ó·š·Í·{·R·â·†·>·ö···ì·¯·g···|·H·]·q···Ã·Ø···························(·U·438d
```

Figure 24: Picture taken on Glass as seen in the EnCase text view

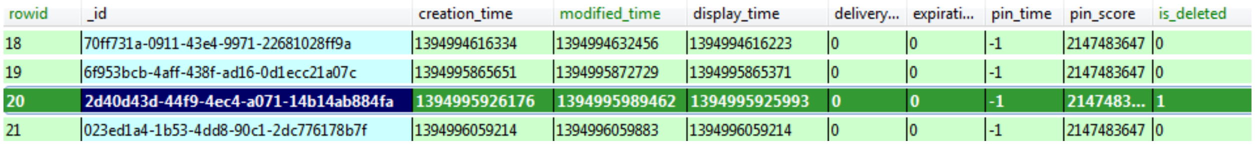
When the “timeline.db” is opened in Firefox SQLite manager, the GUID given indicated by the “$” symbol in the text view can be located in the “_id” column. This can be used to view the creation, modified, and display time for the picture. For this example, the picture was taken and not deleted, so all these times are within a minute of each other, as shown in Figure 25.



rowid	_id	creation_time	modified_time	display_time
1	f99a5c21-8f7d-4bda-bcb2-2f412f6132f3	1394830695572	1394830703120	1394830695371
2	76188630-a7b9-4a1e-b5d2-bc7845ba94eb	1394830819913	1394830820998	1394830819913
3	577817e0-93db-4254-a5a0-6720ffb690a8	1394830949347	1394830950808	1394830949347

Figure 25: A picture taken on Google Glass that is not deleted shown in SQLite Manager

Deleted pictures are not as easy to detect in the Google Glass “timeline.db” text view. The GUID of the deleted picture can be found in the text view, however there are no identifiers which establish it was a picture, or deleted. Looking at the “timeline.db” in Firefox SQLite Manager, the “_id” shows the picture GUID with the time it was taken under creation and display time, and deleted under its modified time. The “is_deleted” column indicates the picture is deleted by a “1”, as shown in Figure 26.

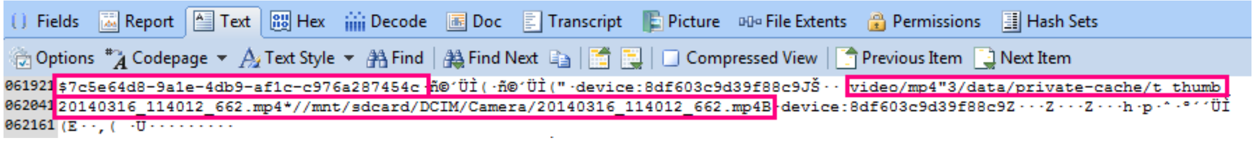


rowid	_id	creation_time	modified_time	display_time	delivery...	expirati...	pin_time	pin_score	is_deleted
18	70ff731a-0911-43e4-9971-22681028ff9a	1394994616334	1394994632456	1394994616223	0	0	-1	2147483647	0
19	6f953bcb-4aff-438f-ad16-0d1ecc21a07c	1394995865651	1394995872729	1394995865371	0	0	-1	2147483647	0
20	2d40d43d-44f9-4ec4-a071-14b14ab884fa	1394995926176	1394995989462	1394995925993	0	0	-1	2147483...	1
21	023ed1a4-1b53-4dd8-90c1-2dc776178b7f	1394996059214	1394996059883	1394996059214	0	0	-1	2147483647	0

Figure 26: A picture taken on Google Glass that was deleted shown in SQLite Manager

Right now, this GUID cannot be linked with an actual artifact, so it is not of much use. Later linking this GUID to entries in the private-cache will be explained. Cross referencing these times to other artifacts on Google Glass could contribute to a better idea of what was happening on the device at a specific time.

When a user records a video using Google Glass, it gets saved on a card on the “timeline.db” as well. As mentioned before, videos can be found by searching for “mnt/sdcard/DCIM/Camera”. Videos will appear with an “.mp4” file extension on Google Glass. Note in Figure 27, “timeline.db” also gives a path to a thumbnail of the video in a “private-cache” folder, which will be explained later.



```
061921 $7c5e64d8-9a1e-4db9-af1c-c976a287454c ñ®´ÜÌ(·ñ®´ÜÌ("·device:8df603c9d39f88c9JŠ·· video/mp4"3/data/private-cache/t_thumb
062041 20140316_114012_662.mp4*//mnt/sdcard/DCIM/Camera/20140316_114012_662.mp4B device:8df603c9d39f88c9Z···Z···Z···h·p·^·ª´´ÜÌ
062161 (E··,(  ·U·········
```

Figure 27: "private-cache" file path in "timeline.db"

If a user deletes a video from the Google Glass timeline, it reacts in a similar way to when a picture is deleted. The GUID can still be found in the text view of “timeline.db”, as well in Firefox SQLite Manager. However, there is no indication as to what this GUID is referring to.

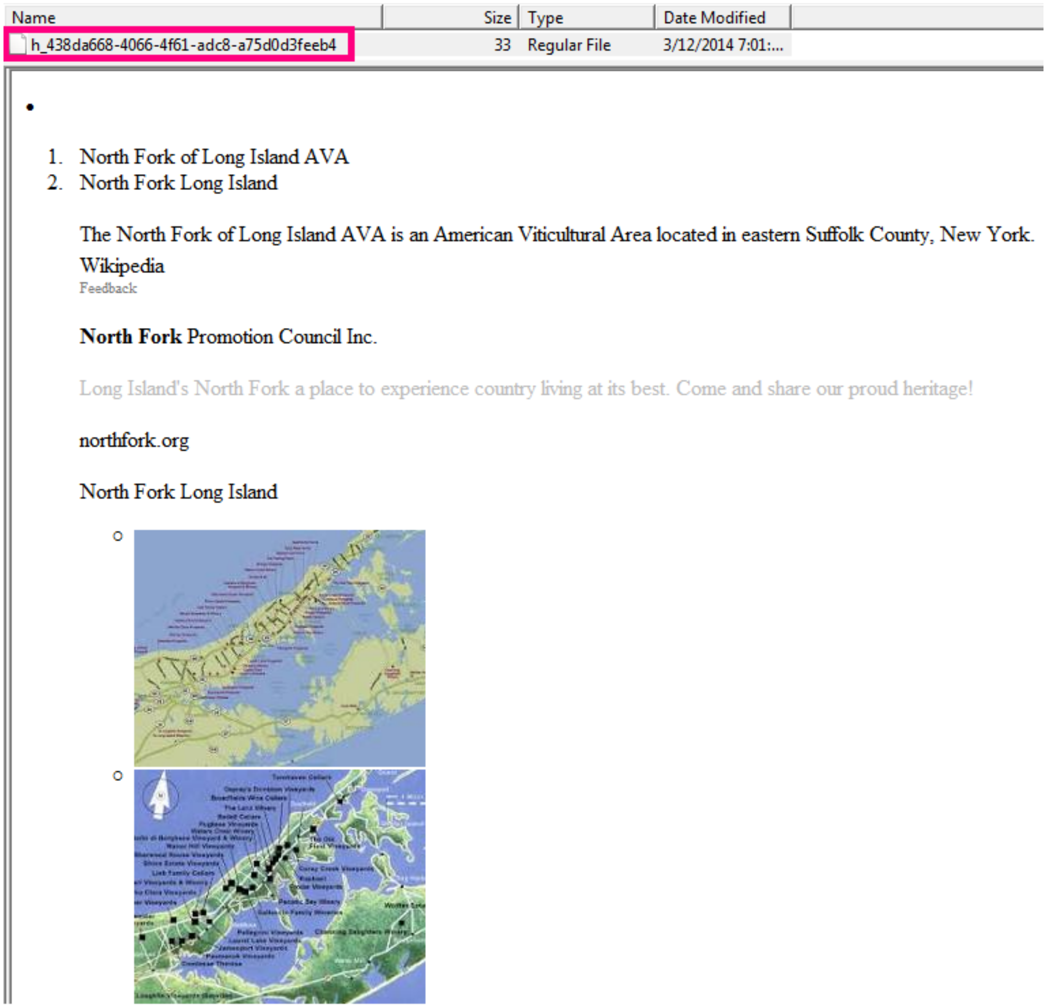
Because both pictures and videos are deleted from the timeline in a way which does not clarify what the “_id” is representing, examiners can see an item was deleted from the timeline, when it was originally created and at what time it was deleted. Cross referencing can give some context to these deleted files.

Cross Referencing with the Private Cache to Identify Timeline Artifacts

The *Image/fs/private-cache* directory caches information so the device can access it quickly. It is a feature made to create a smoother, faster user experience. Although there are not many timestamps in this particular directory, it can give some context to other artifacts on Google Glass with timestamps. The contents of the private-cache folder is structured with files made up of “tags” and “ids” The “tag” is the first part of the file name, which identifies the contents of the file. An underscore separates the “tags” and the “id”. The “id” refers back to the artifact GUIDs in the timeline.db. The “private-cache” directory can be cross referenced to provide more information about data in the “timeline.db”.

Internet History files

The “h_” tag in the “private-cache” file represents internet history. The GUID numbers in the private-cache correlate back to the “timeline.db”. Going back to a previous GUID in the “timeline.db” the file path can be followed to show the related file within the “private-cache”. These files show the contents of what was loaded during a Google search. As shown in Figure 28, the search for, “North Fork Long Island” displays the search results in the order they appeared on Google Glass. It can show both the websites that were loaded along with the pictures that appeared on Google Glass as a result of the search. This can be clearly viewed using FTK Imager. Along with the web history as it appeared on Google Glass, at the very bottom of this file it lists the original search term, which can be cross referenced back to the “timeline.db”. Cross referencing back to the “timeline.db” can also provide timestamps for this particular Google search.



Name	Size	Type	Date Modified
h_438da668-4066-4f61-adc8-a75d0d3feeb4	33	Regular File	3/12/2014 7:01:...

-

1. North Fork of Long Island AVA
2. North Fork Long Island

The North Fork of Long Island AVA is an American Viticultural Area located in eastern Suffolk County, New York. Wikipedia
Feedback

North Fork Promotion Council Inc.

Long Island's North Fork a place to experience country living at its best. Come and share our proud heritage!

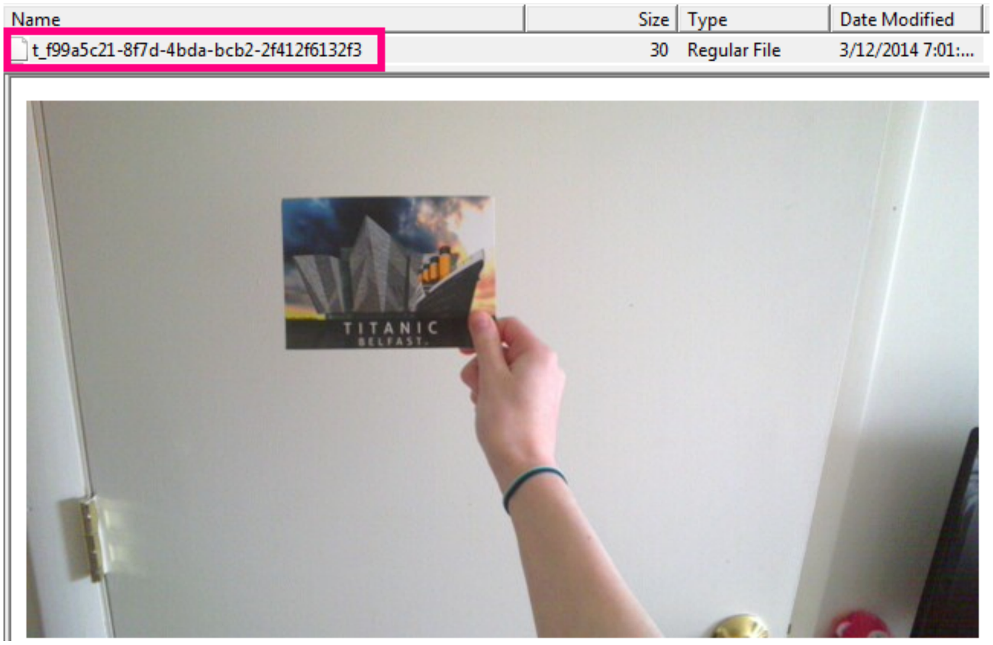
northfork.org

North Fork Long Island

Figure 28: Web History Results in the Private-Cache

Thumbnail files

The “t_” tag in the “private-cache” shows picture thumbnails. These thumbnail pictures link back with the artifacts on the “timeline.db”, and can relate a GUID number and timestamp to an actual picture. From the example on the timeline before, GUID “f99a5c21-8f7d-4bda-bcb2-2f412f6132f3” was taken 3/14/2014 at 4:58:15PM GMT -4. By viewing the related “t_GUID” file in FTK Imager, the examiner can see the actual picture as shown in Figure 29.



Name	Size	Type	Date Modified
t_f99a5c21-8f7d-4bda-bcb2-2f412f6132f3	30	Regular File	3/12/2014 7:01:...

Figure 29: Picture Thumbnail shown in private-cache

This thumbnail picture does not hold any Exif data other than dimensions, however it can be cross referenced back to the “timeline.db” for a timestamp.

The “t_GUID” files are also important because they show the thumbnails of deleted pictures. By looking at the deleted files in the “timeline.db”, and examiner can refer back to the “private-cache” to see if any are “t_GUID” files, which would indicate it was a deleted picture.

The “t_GUID” files also show thumbnails of user recorded videos. The recorded video thumbnails in the “private-cache” are different than the other files because although it has the “t_” tag to identify the file as a thumbnail, it does not have an GUID to relate the file back to “timeline.db”. These video files are also unusual because there are two thumbnails that are stored in the “private-cache” for each video. These thumbnails for a particular video hold the same hash values, however have different naming conventions. One uses the filename, “t_thumb_YYYYMMDD_HHMMSS_Millisecond.mp4”, while the other uses the filename “t_mnt_sdcard_DCIM_Camera_YYYYMMDD_HHMMSS_Millisecond.mp4”. This may be a thumbnail that has been generated from the same video that is saved in two different places on Glass. Because of these naming conventions, there are no GUID numbers which can be referenced back to the timeline.db, however, there are times in the title of these thumbnails. These times can be compared to the creation times on the timeline.db to link it to an entry for more information. For video thumbnails in the private-cache, times in the title of the filename must be used for cross referencing instead of GUIDs.

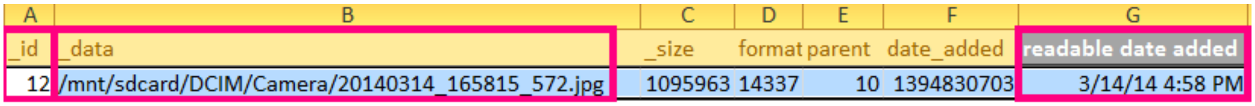
When looking at these “t_” files for videos, it should be noted, in the nature of a thumbnail, it is only a picture. Because the way these files are automatically named, “.mp4” is put at the end of each thumbnail. However after looking into the actual hex of each file, it is clear this is a JFIF picture file. Therefore, it cannot be played back, but a small frame from the start of the video can be seen.

Evidence in the Android Media Database

In the *Image/fs/data/com.android.providers.media/databases/* is a database android devices use to organize media on the device. Within the "external.db" file in the files table of the database, there is information about media files on the device (developer.android.com). This SQLite 3 database was exported into Excel for clearer analysis.

Pictures in external.db

The "external.db" in *Image/fs/data/com.android.providers.media/database* provides information about pictures taken on Google Glass. Non-deleted pictures are listed in the database, as shown in Figure 30.



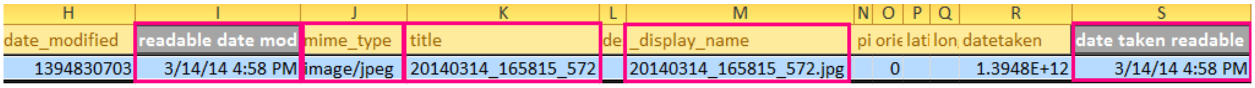
A	B	C	D	E	F	G
_id	_data	_size	format	parent	date_added	readable date added
12	/mnt/sdcard/DCIM/Camera/20140314_165815_572.jpg	1095963	14337	10	1394830703	3/14/14 4:58 PM

Figure 30: Picture link in "external.db"

Each entry in this database has an "_id" number. The "_data" column shows the path of the entry that is being described by this database. This database uses Unix time to store its "date_added" value, however unlike other Unix time values previously seen on Glass, this one does not use millisecond value. Therefore, the formula for transferring this into readable time should be edited to:

=(((F12/60)/60)/24)+DATE(1970,1,1)+(-4/24)

The "date_added" shows the time the picture was taken. There is also a "date_modified" column as shown in the screenshot below. Once again, this uses Unix time, but with no millisecond values. Both of these times can be used to confirm when the picture was taken. The mime type is also listed, with "image" as the type, and "jpeg" as the subtype for the picture. The title and display name columns are similar, using the standard YYMMDD_HHMMSS_Millisecond value Glass stores its picture in. However the "display_name" includes the file extension, in this case, ".jpg". There is also a "date_taken" time listed. This time, unlike the "date_added" and "date_modified" times in this database, does use Unix millisecond time. To get the readable time, the investigator must use the previously defined Unix time millisecond conversion formula from before. This can be seen in Figure 31.

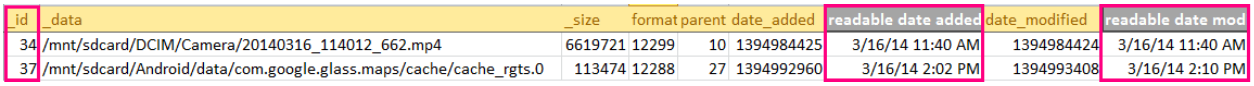


H	I	J	K	L	M	N	O	P	Q	R	S
date_modified	readable date mod	mime_type	title	de	_display_name	pi	orie	lati	lon	datetaken	date taken readable
1394830703	3/14/14 4:58 PM	image/jpeg	20140314_165815_572		20140314_165815_572.jpg		0			1.3948E+12	3/14/14 4:58 PM

Figure 31: Date taken time in "external.db"

From this database, three picture times can be extracted which can be used for cross referencing and further backing an investigation.

Although deleted pictures cannot be seen in this database, there is some evidence that suggests data could have been deleted. When looking at the "_id" numbers in the database, in this example the "_id" jump from 34 to 37 as shown in Figure 32. In between the two times of date added for those entries, two picture were taken and deleted. These entries were deleted from the database, which why it does not include "_id" number 35 and 36 in the external.db. Although this is not an absolute way to tell if picture was deleted, by cross referencing other artifacts on Glass, this could be reinforcement to the idea a picture taken at a specific time was deleted.

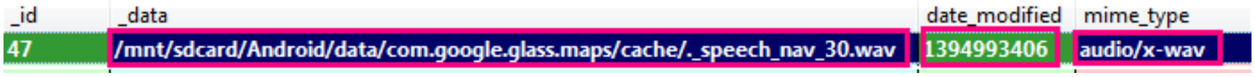


_id	_data	_size	format	parent	date_added	readable date added	date_modified	readable date mod
34	/mnt/sdcard/DCIM/Camera/20140316_114012_662.mp4	6619721	12299	10	1394984425	3/16/14 11:40 AM	1394984424	3/16/14 11:40 AM
37	/mnt/sdcard/Android/data/com.google.glass.maps/cache/cache_rgts.0	113474	12288	27	1394992960	3/16/14 2:02 PM	1394993408	3/16/14 2:10 PM

Figure 32: Missing "_id" numbers in the "external.db"

Directions in external.db

The "external.db" also contains information about the voice directions given on Google Glass. The "_data" column which lists the path of the file shows "/mnt/sdcard/Android/data/com.google.glass.maps/cache/._speech_nav_##.wav". This file relates back to a path on Google Glass which saves the audio files of spoken directions Google Glass gives when a user asks for directions. For these audio directions, the "date_modified" column gives the time each voice command was spoken. This can help understand where a person was in relation to their destination at a specific time. These times are in Unix time, however do not have millisecond value, see Figure 33.



_id	_data	date_modified	mime_type
47	/mnt/sdcard/Android/data/com.google.glass.maps/cache/._speech_nav_30.wav	1394993406	audio/x-wav

Figure 33: GPS direction audio file

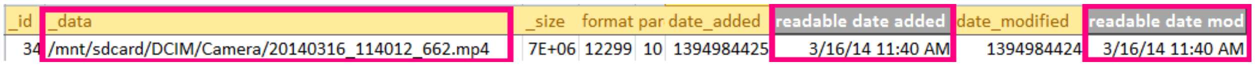
Cross Referencing GPS directions from the external.db

The *Image/fs/sdcard/Android/data/com.google.glass.maps/cache/._speech_nav_##.wav* holds information about the GPS directions given on Google Glass. This directory holds ".wav" audio files with the voice directions Google Glass spoke while giving directions. Although these files themselves do not hold timestamps, they can be referred back to in the, "Image/fs/

com.android.providers.media/databases/external.db" file. This way the investigator can play back each audio file, and hear exactly what direction was spoken at what time.

Videos in external.db

Videos that are not deleted also appear in "external.db". Similar to pictures, under the "_data" column in the database, a file path to the video as shown in Figure 34. It also includes a "date_added" and "date_modified" time. These are in Unix time but not millisecond value.



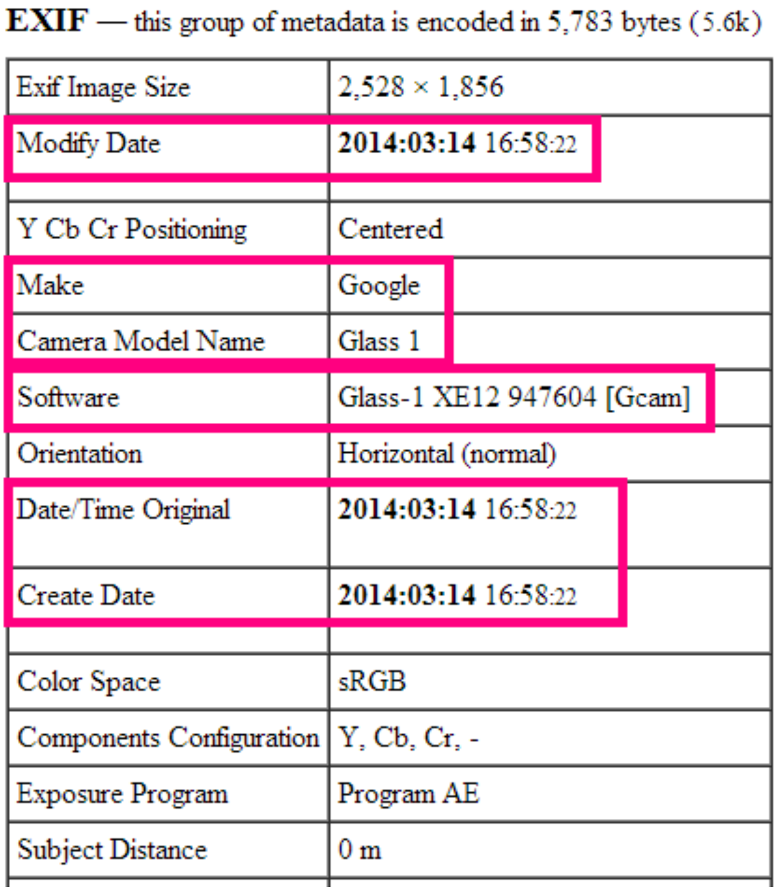
_id	_data	_size	format	par	date_added	readable date added	date_modified	readable date mod
34	/mnt/sdcard/DCIM/Camera/20140316_114012_662.mp4	7E+06	12299	10	1394984425	3/16/14 11:40 AM	1394984424	3/16/14 11:40 AM

Figure 34: Video in the "external.db"

Google Glass Pictures

Google Glass allows the user to take pictures. These pictures are stored in several different places, which can help when cross referencing in a forensic case.

Within the, *Image/fs/media/DCIM/Camera* directory are files that are named by their date and time, see Figure 35. These files are pictures or videos the user took using Google Glass. These files are in ".jpg" or ".mp4" format. These pictures hold the date and time information right in the file name, in YYYYMMDD_HHMMSS_Millisecond format, which can be helpful for an examiner. In addition to the date and time in the title of the picture, they contain Exif data which tells the time the picture was taken. For example, for "20140314_165815_572.jpg", the Exif data confirms this is the date and time the picture was taken. This Exif data also shows it was taken on Google Glass running XE12. Pictures that were deleted do not appear in this directory.



EXIF — this group of metadata is encoded in 5,783 bytes (5.6k)

Exif Image Size	2,528 × 1,856
Modify Date	**2014:03:14** 16:58:22
Y Cb Cr Positioning	Centered
Make	Google
Camera Model Name	Glass 1
Software	Glass-1 XE12 947604 [Gcam]
Orientation	Horizontal (normal)
Date/Time Original	**2014:03:14** 16:58:22
Create Date	**2014:03:14** 16:58:22
Color Space	sRGB
Components Configuration	Y, Cb, Cr, -
Exposure Program	Program AE
Subject Distance	0 m

Figure 35: Picture Exif

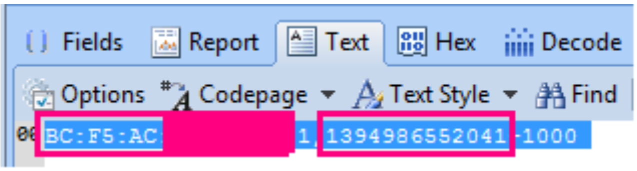
Image/fs/sdcard/DCIM/Camera/YYYYMMDD_HHMMSS_Millisecond.xxx is another place Google Glass stores pictures, on the SD card. The file name of the pictures and videos give information about the date and time they were taken. These pictures, just like the pictures stored in*, Image/fs/media/DCIM/Camera* show user taken pictures and videos. Deleted pictures and videos do not appear in here. These files have the same hash values as the pictures stored in, *Image/fs/media/DCIM/Camera*. Therefore, the Exif data is also the same. The SD card is another location to find pictures on Google Glass and check dates and times.

Bluetooth Pairing on Glass

The *Image/fs/misc* folder gives information about the Bluetooth and Wifi connections on Google Glass. This is useful to see when and what devices were paired with Glass at a particular time and other connectivity activity.

MAC address paired with Google Glass

This file gives information about the time a Bluetooth device was paired with Google Glass. The device's MAC address is given along with the Unix millisecond time. Figure 36 shows that the particular MAC address was paired with Google Glass on 3/16/2014 at 12:15:52 PM GMT -4.

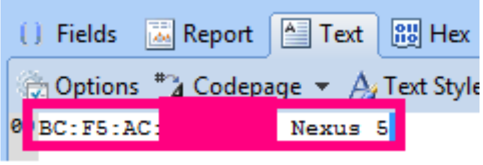


```
BC:F5:AC[redacted] 1 1394986552041-1000
```

Figure 36: Bluetooth MAC address and Unix Millisecond Time

Name of Bluetooth Device Paired with Glass

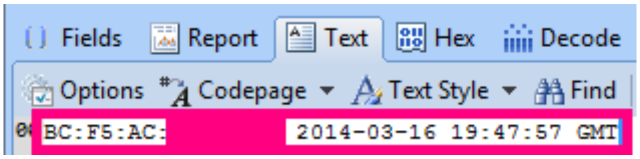
The *Image/fs/misc/bluetooth/bluetoothd/* directory provides information about the Bluetooth devices that were paired with Google Glass. Under the "names" file, the MAC address of the device Glass was paired with is listed, as well as the devices name, as shown in Figure 37. Although this doesn't have a timestamp, it can be cross referenced back to *Image/fs/misc/Bluetooth/incoming_connection.conf* to get more information about the device it was paired with.



```
BC:F5:AC:[redacted] Nexus 5
```

Figure 37: MAC Address and Bluetooth Device Name

Another file in this directory is the "last_used" file, shown in Figure 38. It shows the last time the device was used with MAC address given. In this case, when Google Glass was powered on, it automatically paired with the mobile phone, so therefore the last used time is shown at its most recent power on.



```
BC:F5:AC:[redacted] 2014-03-16 19:47:57 GMT
```

Figure 38: Last Used Bluetooth Connection

Last time used timestamps

The *Image/fs/system/usagestats/usage-history.xml* file holds information about some of the last times a user did something on Google Glass. Figure 39 shows this .xml file.

When Google Glass is setup on WiFi, a computer is used to generate a QR code that Glass uses to connect to the network. The "com.google.glass.setup.BarcodeScanActivity" gives the Unix millisecond time of the last time Google Glass scanned the QR code on a computer for setup.

The "com.google.glass.maps.NavigationLauncherActivity" gives the Unix millisecond time of the last time the user requested Google Glass get GPS directions.

When a user says “Ok Glass, get directions to” for the first time to get GPS directions, a warning disclaimer appears on the screen. The “com.google.glass.maps.DisclaimerActivity” gives the Unix millisecond time of the first time the user on Google Glass requested GPS directions and therefore saw the disclaimer and accepted it.

The “com.google.glass.maps.NavigationActivity” gives the Unix millisecond time of the last time the user arrived at a location they asked for GPS directions to.

“com.google.glass.browser.WebBrowserActivity” gives the Unix millisecond time of the last web activity the user participated in. This does not necessarily mean a Google search was done at this time. This could also mean the user opened up previously searched for web results from a card on the timeline.

The “com.google.glass.bluetooth.pairing.BluetoothPairingConfirmationActivity” gives the Unix millisecond time of the last time the user accepted a Bluetooth device be paired to Google Glass.

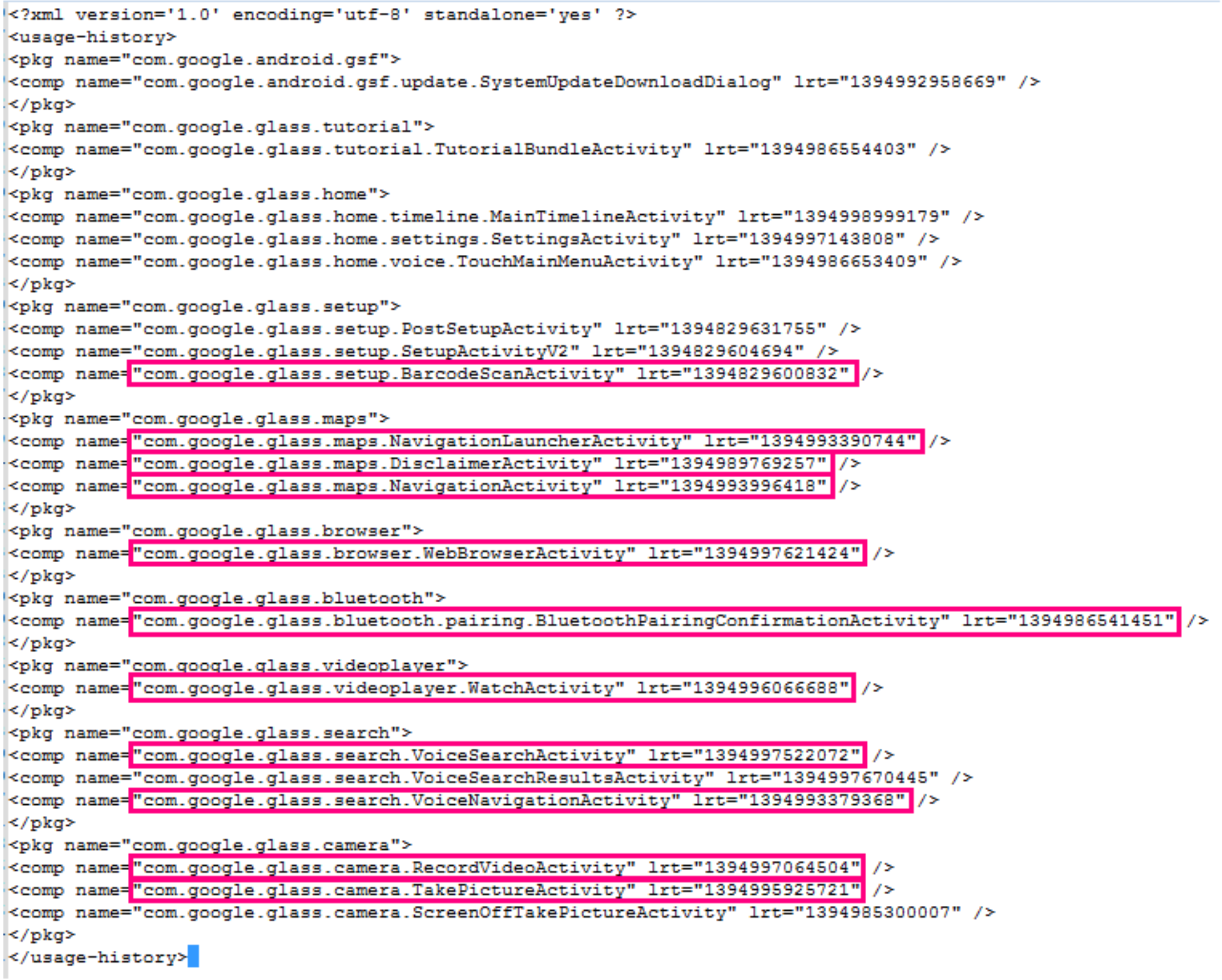
The last time a user watched a video on Google Glass is represented in Unix millisecond time by “com.google.glass.videoplayer.WatchActivity”.

The “com.google.glass.search.VoiceSearchActivity” indicates the Unix time of the last time the user made a voice search on Google Glass.

The last time the user requested voice directions is shown in Unix millisecond time by “com.google.glass.search.VoiceNavigationActivity”.

The “com.google.glass.camera.RecordVideoActivity” shows the Unix millisecond time of the last time a user recorded a video on Google Glass. This includes deleted videos.

The “com.google.glass.camera.TakePictureActivity” shows the Unix millisecond time of the last time the user took a picture on Google Glass. This includes deleted pictures.



```xml
<?xml version='1.0' encoding='utf-8' standalone='yes' ?>
<usage-history>
<pkg name="com.google.android.gsf">
<comp name="com.google.android.gsf.update.SystemUpdateDownloadDialog" lrt="1394992958669" />
</pkg>
<pkg name="com.google.glass.tutorial">
<comp name="com.google.glass.tutorial.TutorialBundleActivity" lrt="1394986554403" />
</pkg>
<pkg name="com.google.glass.home">
<comp name="com.google.glass.home.timeline.MainTimelineActivity" lrt="1394998999179" />
<comp name="com.google.glass.home.settings.SettingsActivity" lrt="1394997143808" />
<comp name="com.google.glass.home.voice.TouchMainMenuActivity" lrt="1394986653409" />
</pkg>
<pkg name="com.google.glass.setup">
<comp name="com.google.glass.setup.PostSetupActivity" lrt="1394829631755" />
<comp name="com.google.glass.setup.SetupActivityV2" lrt="1394829604694" />
<comp name="com.google.glass.setup.BarcodeScanActivity" lrt="1394829600832" />
</pkg>
<pkg name="com.google.glass.maps">
<comp name="com.google.glass.maps.NavigationLauncherActivity" lrt="1394993390744" />
<comp name="com.google.glass.maps.DisclaimerActivity" lrt="1394989769257" />
<comp name="com.google.glass.maps.NavigationActivity" lrt="1394993996418" />
</pkg>
<pkg name="com.google.glass.browser">
<comp name="com.google.glass.browser.WebBrowserActivity" lrt="1394997621424" />
</pkg>
<pkg name="com.google.glass.bluetooth">
<comp name="com.google.glass.bluetooth.pairing.BluetoothPairingConfirmationActivity" lrt="1394986541451" />
</pkg>
<pkg name="com.google.glass.videoplayer">
<comp name="com.google.glass.videoplayer.WatchActivity" lrt="1394996066688" />
</pkg>
<pkg name="com.google.glass.search">
<comp name="com.google.glass.search.VoiceSearchActivity" lrt="1394997522072" />
<comp name="com.google.glass.search.VoiceSearchResultsActivity" lrt="1394997670445" />
<comp name="com.google.glass.search.VoiceNavigationActivity" lrt="1394993379368" />
</pkg>
<pkg name="com.google.glass.camera">
<comp name="com.google.glass.camera.RecordVideoActivity" lrt="1394997064504" />
<comp name="com.google.glass.camera.TakePictureActivity" lrt="1394995925721" />
<comp name="com.google.glass.camera.ScreenOffTakePictureActivity" lrt="1394985300007" />
</pkg>
</usage-history>
```

Figure 39: Usage History File

Conclusion

Google Glass is new device worth examining, with very little research done on it. How time is stored by artifacts on Glass is not uniformed, due to half the system using Unix millisecond time and the other half using Unix second time. In addition to Unix times, pictures are also stored in date and time format which provides time information. By understanding each of these timestamps on Glass, investigators can piece together a bigger picture and understand what a user was doing on the device at a particular time. Cross referencing these time stamps can back up a piece of evidence, and build a stronger case. The ability to understand timestamps on Google Glass allows examiners to provide better evidence.

With Google Glass being released shortly, there is a need for these devices to be forensically analyzed so digital forensic examiners can be prepared when they come across one. “A Digital Forensic Examiner’s Guide to Google Glass,” found on desautelsja.blogspot.com is a series of posters aimed to help investigators tackle their first Google Glass analysis and have a smoother investigation. This research should help save forensic examiners time, and help the entire digital forensics community.

Works Cited

"Android Developer Tools." Android Developers. N.p., n.d. Web. <http://developer.android.com/tools/help/adt.html>.

Brain, Marshall. "How Lithium-ion Batteries Work" 14 November 2006. HowStuffWorks.com. <http://electronics.howstuffworks.com/everyday-tech/lithium-ion-battery.htm> 16 March 2014.

"Dumpsys." Android Developers. N.p., n.d. Web. <http://source.android.com/devices/tech/input/dumpsys.html>.

"Google Glass Specifications Revealed – Runs on Android 4.0.3 ICS or Later." Crypt Life. N.p., 16 Apr. 2013. Web. <http://www.cryptlife.com/gadgets/google-glass-specifications>.

"Guid Structure." Microsoft Developer Network. Microsoft, n.d. Web. <http://msdn.microsoft.com/en-us/library/system.guid.aspx>.

"Head Wake Up." Google Glass Help. N.p., n.d. Web. <https://support.google.com/glass/answer/3079855?hl=en>.

"JSON Transformation Examples: Transform." IBM. IBM, n.d. Web. <http://pic.dhe.ibm.com/infocenter/wsdatap/v6r0m0/index.jsp?topic=%2Fcom.ibm.dp.xg.doc%2Fjson_jsontransformationexample3.html>.

"MediaStore.Files.FileColumns." Android Developers. N.p., n.d. Web. <http://developer.android.com/reference/android/provider/MediaStore.Files.FileColumns.html>.

"Project Glass: One Day..." YouTube. YouTube, 04 Apr. 2012. Web.

<https://www.youtube.com/watch?v=9c6W4CCU9M4>.

Rouse, Margaret. "Pulse Code Modulation (PCM)." Search Networking. N.p., July 2005. Web. <http://searchnetworking.techtarget.com/definition/pulse-code-modulation-PCM>.

"System and Kernel Downloads." Google Developers. Google, n.d. Web. <https://developers.google.com/glass/tools-downloads/system>.

"Using MD5 to Verify the Integrity of File Contents." Using MD5 to Verify the Integrity of File Contents. University of South Carolina, n.d. Web. <http://www.cse.sc.edu/~okeefe/tutorials/cert/i002.01.html>.

"Visual Networking Index: Global Mobile Data Traffic Forecast Update, 2013–2018." Cisco. Cisco, 5 Feb. 2014. Web. <http://www.cisco.com/c/en/us/solutions/collateral/service-provider/visual-networking-index-vni/white_paper_c11-520862.html>.

"Voice Biometric Authentication." Authentify. N.p., n.d. Web. <http://www.authentify.com/solutions/voice_biometrics.html>.