

**518.4**

# Application Data Analysis

**SANS**

# 518.4

# Application Data Analysis





Copyright © 2020, Sarah Edwards. All rights reserved to Sarah Edwards and/or SANS Institute.

PLEASE READ THE TERMS AND CONDITIONS OF THIS COURSEWARE LICENSE AGREEMENT ("CLA") CAREFULLY BEFORE USING ANY OF THE COURSEWARE ASSOCIATED WITH THE SANS COURSE. THIS IS A LEGAL AND ENFORCEABLE CONTRACT BETWEEN YOU (THE "USER") AND SANS INSTITUTE FOR THE COURSEWARE. YOU AGREE THAT THIS AGREEMENT IS ENFORCEABLE LIKE ANY WRITTEN NEGOTIATED AGREEMENT SIGNED BY YOU.

With the CLA, SANS Institute hereby grants User a personal, non-exclusive license to use the Courseware subject to the terms of this agreement. Courseware includes all printed materials, including course books and lab workbooks, as well as any digital or other media, virtual machines, and/or data sets distributed by SANS Institute to User for use in the SANS class associated with the Courseware. User agrees that the CLA is the complete and exclusive statement of agreement between SANS Institute and you and that this CLA supersedes any oral or written proposal, agreement or other communication relating to the subject matter of this CLA.

BY ACCEPTING THIS COURSEWARE, YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. BY ACCEPTING THIS SOFTWARE, YOU AGREE THAT ANY BREACH OF THE TERMS OF THIS CLA MAY CAUSE IRREPARABLE HARM AND SIGNIFICANT INJURY TO SANS INSTITUTE, AND THAT SANS INSTITUTE MAY ENFORCE THESE PROVISIONS BY INJUNCTION (WITHOUT THE NECESSITY OF POSTING BOND) SPECIFIC PERFORMANCE, OR OTHER EQUITABLE RELIEF.

If you do not agree, you may return the Courseware to SANS Institute for a full refund, if applicable.

User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of the Courseware, in any medium whether printed, electronic or otherwise, for any purpose, without the express prior written consent of SANS Institute. Additionally, User may not sell, rent, lease, trade, or otherwise transfer the Courseware in any way, shape, or form without the express written consent of SANS Institute.

If any provision of this CLA is declared unenforceable in any jurisdiction, then such provision shall be deemed to be severable from this CLA and shall not affect the remainder thereof. An amendment or addendum to this CLA may accompany this Courseware.

SANS acknowledges that any and all software and/or tools, graphics, images, tables, charts or graphs presented in this Courseware are the sole property of their respective trademark/registered/copyright owners, including:

AirDrop, AirPort, AirPort Time Capsule, Apple, Apple Remote Desktop, Apple TV, App Nap, Back to My Mac, Boot Camp, Cocoa, FaceTime, FileVault, Finder, FireWire, FireWire logo, iCal, iChat, iLife, iMac, iMessage, iPad, iPad Air, iPad Mini, iPhone, iPhoto, iPod, iPod classic, iPod shuffle, iPod nano, iPod touch, iTunes, iTunes logo, iWork, Keychain, Keynote, Mac, Mac Logo, MacBook, MacBook Air, MacBook Pro, Macintosh, Mac OS, Mac Pro, Numbers, OS X, Pages, Passbook, Retina, Safari, Siri, Spaces, Spotlight, There's an app for that, Time Capsule, Time Machine, Touch ID, Xcode, Xserve, App Store, and iCloud are registered trademarks of Apple Inc.

PMP and PMBOK are registered marks of PMI.

SOF-ELK® is a registered trademark of Lewes Technology Consulting, LLC. Used with permission.

SIFT® is a registered trademark of Harbingers, LLC. Used with permission.

Governing Law: This Agreement shall be governed by the laws of the State of Maryland, USA.

FOR518.4

Mac and iOS Forensic Analysis and Incident Response



# FOR518 Section 4: Application Data Analysis

© 2020 Sarah Edwards | All Rights Reserved | Version F01\_01

Author: Sarah Edwards  
oompa@csh.rit.edu  
mac4n6.com  
<http://twitter.com/iamevltwin>

<https://digital-forensics.sans.org/>  
<http://twitter.com/sansforensics>

## Course Agenda

Section : Mac and iOS Essentials

Section 2: File Systems and System Triage

Section 3: User Data, System Configuration, and Log Analysis

Section 4: Application Data Analysis

Section 5: Advanced Analysis Topics

Section 6: Mac Forensic Challenge

This page intentionally left blank.





**SANS DFIR**

DIGITAL FORENSICS & INCIDENT RESPONSE

---

# Application Data Analysis

---

**SANS | DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 3

This page intentionally left blank.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



---

## Section 4: Part I

# Application Fundamentals

---

This page intentionally left blank.



## App Fundamentals

File System Locations

iOS Snapshots

Permissions

App Testing and Analysis

This page intentionally left blank.

## App Preference Files

### Application Configurations

- Bundle ID/Reverse DNS Filename – net.whatsapp.WhatsApp.plist, com.apple.Maps.plist

### macOS

- ~/Library/Preferences/
- ~/Library/Containers/.../<bundle\_id>/.../Preferences/

### iOS Backup

- /mobile/Applications/<bundle\_id>/
- /mobile/Library/Preferences

### iOS Physical

- /private/var/mobile/Containers/.../<bundle\_id>/Preferences/
- /private/var/mobile/Preferences

Preference files are usually plist files that contain configuration data for a particular application. This data can include a wide variety of items.

These preference files are plist files that are named using the bundle ID, the reverse DNS format.

## Example Preference File: Hyatt Mobile Entry App

▼ Root	Dictionary	(20 items)
▶ ADMS_LifecycleData	Dictionary	(14 items)
ADMS_PAUSE	Date	2015-05-03 23:59:06
ADMS_START	Date	2015-05-03 23:51:31
ADOBEMOBILE_STOREDEDEFAULTS_AID	String	2A9D72208507A683-4000010A200860D2
CurrentProperty	Number	2
OMCK1	Date	2015-04-25 00:48:01
OMCK2	String	1
OMCK5	Date	2015-05-03 23:48:26
OMCK6	Number	3
OMCK7	Boolean	True
WebDatabaseDirectory	String	/var/mobile/Containers/Data/Application/A977
WebKitDiskImageCacheSavedCacheDirectory	String	
WebKitLocalStorageDatabasePathPreferenceKey	String	/var/mobile/Containers/Data/Ap
WebKitOfflineWebApplicationCacheEnabled	Boolean	True
WebKitShrinksStandaloneImagesToFit	Boolean	True
com.hyatt.mobile_key.goldPassportId	String	[REDACTED]
com.hyatt.mobile_key.guestName	String	SARAH EDWARDS
com.hyatt.mobile_key.rememberMe	Boolean	True
com.hyatt.mobile_key.savedGoldPassportNumberOrUsername	String	[REDACTED]
com.hyatt.mobile_key.savedGoldPassportPassword	String	[REDACTED]

Yep,  
password  
in  
plaintext  
☹

In addition to application data such as messages or contact lists, an analyst may need to look for user IDs such as email addresses and account numbers. Analysts may also find passwords in plaintext associated with various accounts.

The example above is from the Hyatt Mobile Entry application (`com.hyatt.MobileEntry.plist`). The app configuration plist files may contain some interesting information and are always worth quickly reviewing. If an analyst is trying to find passwords used to attempt to break into other encrypted containers, seemingly random applications on that user's device may contain this information in an easy-to-get way! Why brute force when you can get passwords in plaintext!



## App Databases and Other Files

SQLite Databases, Proprietary Files, Logs – Anything Goes!

### macOS

- ~/Library/
- ~/Library/Application Support/
- ~/Library/Containers/...

### iOS Backup

- /mobile/Applications/<bundle\_id>/
- /mobile/Library/

### iOS Physical

- /private/var/mobile/Containers/... /<bundle\_id>/Library/
- /private/var/mobile/Library/

Databases and other application files can be found in these locations. Each application will have its own files representing the data used for the apps themselves. Common files seen are SQLite database files, proprietary data, and maybe some log files.

## Application Caches

### Cache.db SQLite Database and/or Other Cached Files

#### macOS

- ~/Library/Caches/
- ~/Library/Containers/.../<bundle\_id>/.../Cache/

#### iOS Backup

- Probably little to none; cached items rarely get backed up

#### iOS Physical

- /private/var/mobile/Containers/.../<bundle\_id>/Library/Caches/
- /private/var/mobile/Library/Cache/<bundle\_id>

Application caches are always worth a review. Browsers are often the most common for an analyst to review; however, other apps may retain cached data that could be of importance in an investigation.

## Example Cache.db File: Uber Application

```
1 select cfurl_cache_response.entry_ID,request_key,time_stamp, receiver_data
2 from cfurl_cache_response
3 left outer join cfurl_cache_receiver_data on cfurl_cache_response.entry_ID = cfurl_cache_receiver_data.entry_ID
4 where cfurl_cache_response.Entry_ID = 613
```

entry_ID	request_key	time_stamp	receiver_data
1 613	https://cn-sjc1.uber.com/rt/geocoding/reverse?latitude=47.44236897086387&longitude=-122.300106501819&lang...	2016-03-12 22:54:02	{"latitude":4...

Mode:  Import Export as NULL

```
["latitude":47.443301,"longitude":-122.3016229,"shortAddress":"12 Departures Dr","longAddress":"12 Departures Dr, SeaTac, WA 98158, USA","addressComponents":[{"long_name":"12","short_name":"12","types":["street_number"]},{"long_name":"Departures Drive","short_name":"Departures Dr","types":["route"]},{"long_name":"SeaTac","short_name":"SeaTac","types":["locality","political"]}, {"long_name":"King County","short_name":"King County","types":["administrative_area_level_2","political"]}, {"long_name":"Washington","short_name":"WA","types":["administrative_area_level_1","political"]}, {"long_name":"United States","short_name":"US","types":["country","political"]}, {"long_name":"98158","short_name":"98158","types":["postal_code"]}]]
```

Another type of data an analyst may be searching for is location data. Depending on the acquisition type, the analyst may not have access to the location databases on iOS. An examiner can get creative and look in many types of applications for this data.

The example above shows the `Cache.db` (`com.ubercab.UberClient/Cache.db`) file for the Uber application. Any applications that use location services may be good candidates for location data. The Uber `cache.db` database contains HTTP URL queries containing location data for where the user was requesting a car. The response (shown in the “Edit Database Cell” window) contains the return information from this query, which also contains location information with more granular detail.

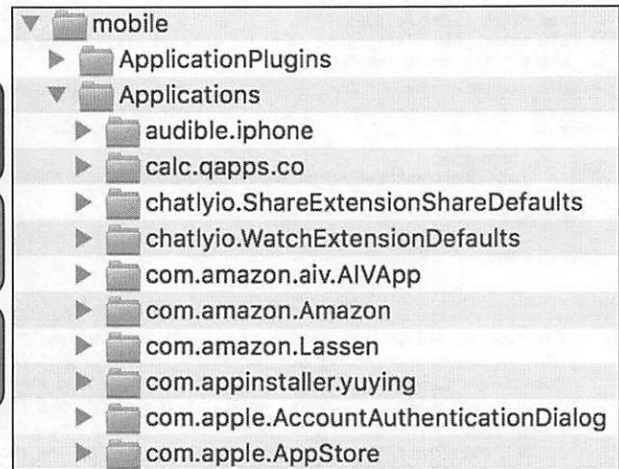


## iOS Backup Structure: Third-Party App Data

/mobile/Applications/<App\_Bundle\_ID>

All Third-Party App Data under Apple Bundle ID Directory

Normalized Structure



The iOS backup structure when shown in BlackLight (shown) and other tools will show a normalized data structure. This is not exactly how it is being stored on disk.

All application data is stored under the directory associated with that application's Bundle ID.

## iOS Containers: Disk Structure App Data [iOS 8+]

`/private/var/mobile/Containers/`

`/Bundle/Application/<GUID>/`

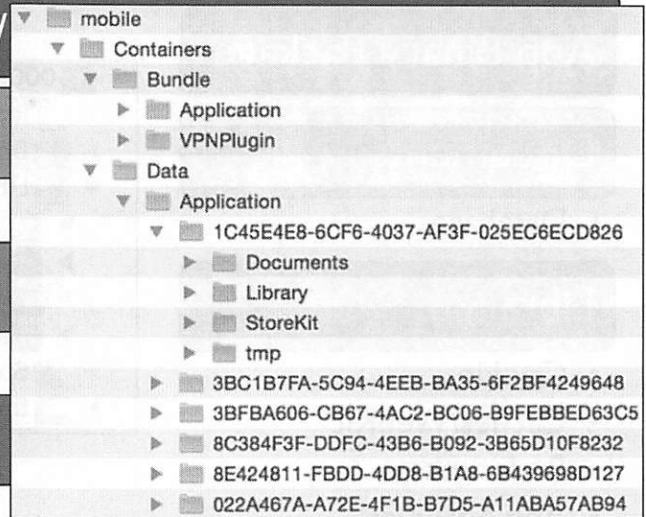
- Contains the Application Binary

`/Data/Application/<GUID>/`

- App Data

`/Shared/AppGroup/<GUID>/`

- Shared App Data



iOS 8 introduced Containers to the file system. Application data has been slightly moved around to fit this new sandboxing implementation.

Much of the same data (i.e., `/Documents`, `/Library/Caches`, `/Library`, etc.) is stored under a new file path, as shown above.

A database in iOS 7 might have had the path:

`/private/var/mobile/Applications/<GUID>/Documents/somedatabase.db`.

In iOS 8, this path may be

`/private/var/mobile/Containers/Data/Application/<GUID>/Documents/somedatabase.db`

Also new to iOS 8 is “shared” data. This can be data shared among apps by the same developer, such as the Microsoft Word, PowerPoint, and Excel applications (for iDevices of course!)

Reference:

<https://developer.apple.com/library/content/documentation/FileManagement/Conceptual/FileSystemProgrammingGuide/FileSystemOverview/FileSystemOverview.html>

## iOS Third-Party App Directory Structure [iOS 8+] [1]

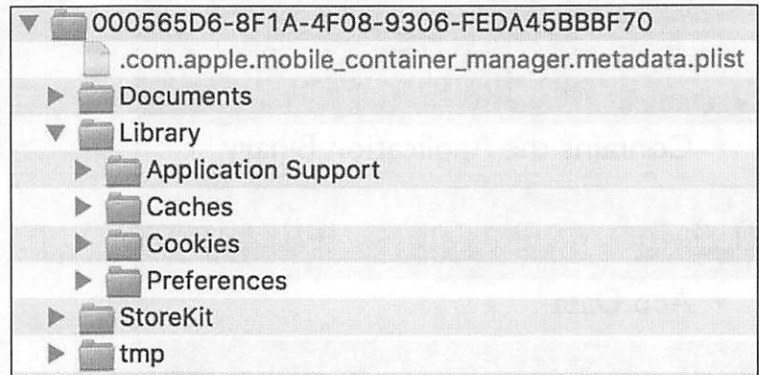
### App Binary Package

### Documents

- Databases

### Library

- Cookies
- Caches/WebKit
- Preferences
- App Analytics



Each third-party application directory contains a standard set of folders which (usually) has a basic set of data contained within them.

The Documents directory may contain documents, but also application databases. The Library directory may contain cookie data, cached data in the cache.db databases, and configuration plists, as well as application analytics information.

For iOS 8 and newer devices, the Shared directory should also be reviewed for app data. Some apps will use this directory for all their files instead of the one above.

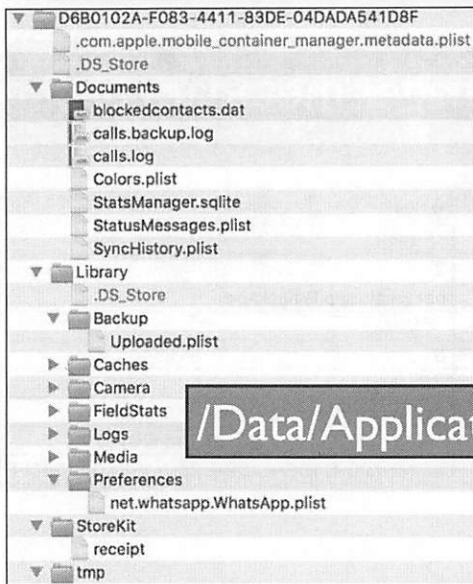
## iOS Third-Party App Directory Structure [iOS 8+] [2]

`.com.apple.mobile_container_manager.metadata.plist`

Key	Type	Value
▼ Root	Dictionary	(4 items)
MCMMetadataContentClass	Number	2
MCMMetadataIdentifier	String	net.whatsapp.WhatsApp
▼ MCMMetadataInfo	Dictionary	(2 items)
com.apple.MobileInstallation.ContentProtectionClass	Number	0
▼ com.apple.MobileInstallation.GroupContainerIDs	Array	(2 items)
Item 0	String	group.com.facebook.family
Item 1	String	group.net.whatsapp.WhatsApp.shared
MCMMetadataUUID	String	3166F790-D188-4A09-B95C-1F84D8F886FF

Each application on iOS 8 or newer will have a hidden file in its root directory containing the Bundle ID of the application as well as other metadata, such as protection classes and group bundle ID information.

## App Data Directory and Shared Data Directory [iOS 8+]



/Data/Application/



/Shared/AppGroup/

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 16

The differences between the `/Data/Application/` and the `/Shared/AppGroup/` directories in the app containers can be quite different and it's always worth looking at both to find the information you are seeking.

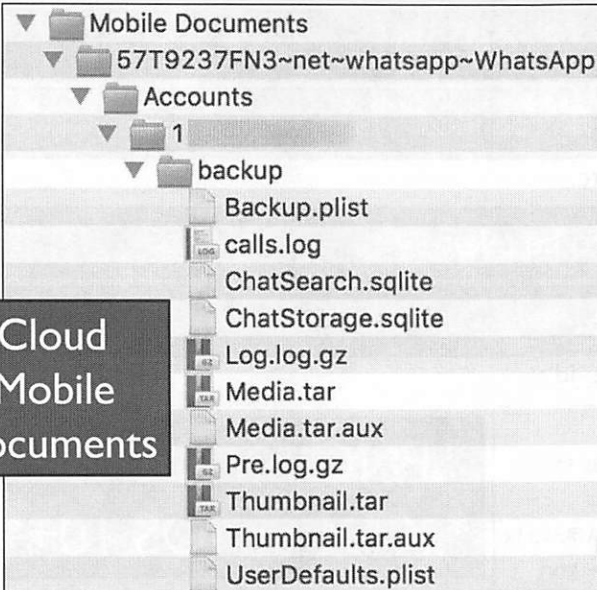
On the left is a screenshot of the `/Data/Application` directory for the WhatsApp app (`net.whatsapp.whatsapp`). This shows some information, but if you are looking for the Messages and Contacts databases for the WhatsApp app, you will have to look in the `/Shared/AppGroup/` directory instead.

Not shown in the screenshots above, the Media directory in the `/Data/Application/` directory contains the media (photos/videos) sent back and forth in conversations—this media is not in the `/Shared/AppGroup/` directory for WhatsApp.

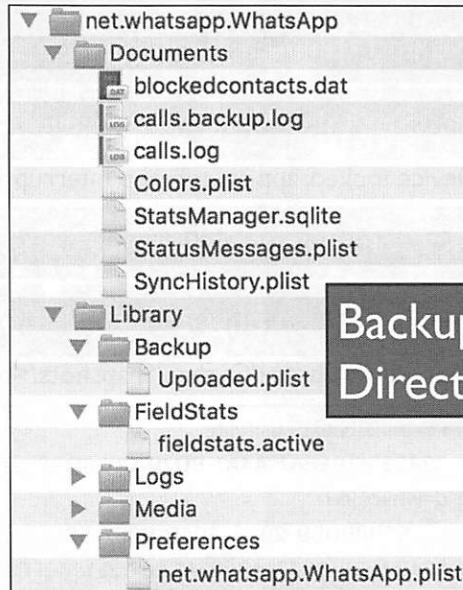
These examples come from physical-logical dumps.



## iCloud and Backup Data Directories



iCloud  
Mobile  
Documents



Backup  
Directory

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 17

Along with the `/Data/Application/` and `/Shared/AppGroup/` directories, the analyst should also look at the `iCloud Mobile Documents` directory (shown left), as well as any backups that may be found on the device.

Each directory structure will choose to store only a specific directory. If the app does not want to back up anything to iCloud, you will likely not find much in the `Mobile Documents` directory. If the app developer chooses to back up specific items in local backups, you will find these in the normalized data structure, such as the example shown on the right for WhatsApp.

The `Mobile Documents` directory example comes from a physical-logical dump.

## iOS Application Snapshots

Native and third-party applications

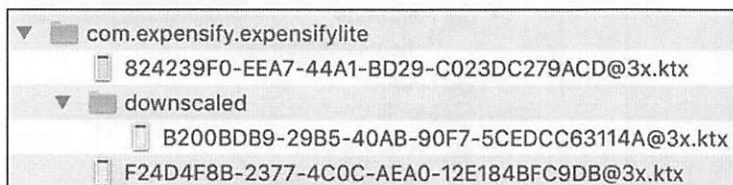
Screenshot taken when application is backgrounded

- Device locked, app switch, app interruption, etc.

Application may not allow screenshots to be taken

Only available on physical acquisitions

- `<app_dir>/Library/Caches/Snapshots/<bundle_id>/`



\*.PNG on iOS 9-  
\*.KTX on iOS 10+

Whenever an application is backgrounded, a screenshot is taken of what the user is currently viewing on the screen and is saved to the file system. This allows for smoother transitions and the appearance of faster switching between applications.

A screenshot is taken when an app is switched to another application, when the device is locked, or when an application is interrupted by another application (i.e., receiving a phone call).

These screenshots may be available for native as well as third-party applications. Depending on the application, developers may choose to limit this screen capture by explicitly telling their code to do so. This may be done for “security”, as sensitive data may be saved in these applications. Many messaging or banking applications do not use this feature, or they use a decoy image instead.

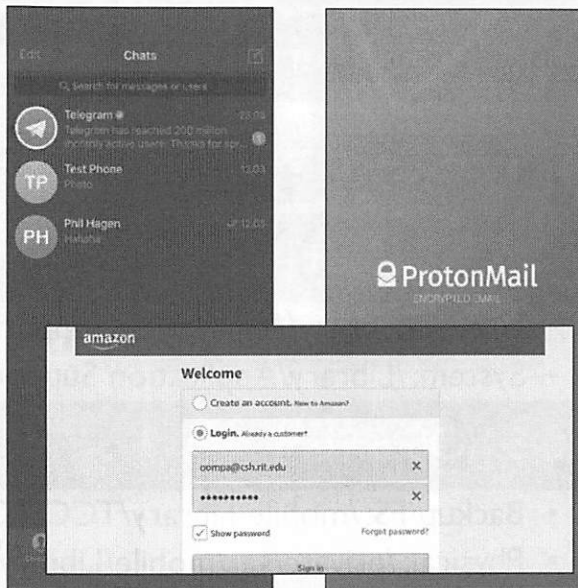
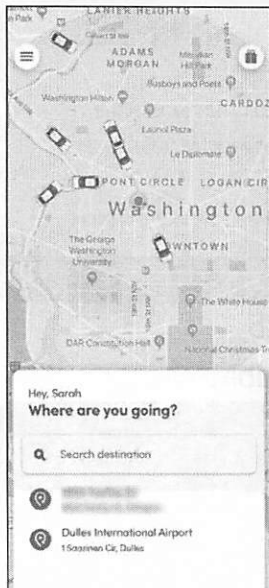
These screen captures are only available on physical acquisitions.

These screenshots are PNG files on iOS 9 and older. On iOS 10, the files use a newer format—KTX. These files can be viewed in the macOS Preview application.

Reference:

[https://www.khronos.org/opengles/sdk/tools/KTX/file\\_format\\_spec/](https://www.khronos.org/opengles/sdk/tools/KTX/file_format_spec/)

## iOS Application Snapshot Examples



SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 19

Snapshots can be used in many ways to help an investigation. You are viewing a snapshot in time of exactly what the user was seeing. These pictures have the potential to contain information that may not necessarily be stored in the applications database or plist files.

Snapshots may be different for applications depending whether or not the user was using the application in portrait or landscape modes.

Contacts / Address Book: Last contact viewed

Camera: Last "thing" viewed, perhaps without having an actual photo taken

Calendar: Schedule the last day viewed

Messages: List of messaged contacts with last message to them—could potentially be captured before message is deleted

Third-Party Messaging Apps: Messages viewed, typed

Mapping Applications: Last viewed location

Web Browsers: Last viewed webpage

The examples above from left to right are the following apps:

- Lyft
- Google Voice
- Telegram ("Secure" Messenger)
- Proton Mail – Secure Email
- Safari Browser (horizontal)

## Application Transparency, Consent, Control (TCC)

### Application Permissions

### SQLite Database

### macOS [10.8+]

- User: ~/Library/Application Support/com.apple.TCC/TCC.db
- System: /Library/Application Support/com.apple.TCC/TCC.db

### iOS [iOS 6+]

- Backup/FS: /mobile/Library/TCC/TCC.db
- Physical: /private/var/mobile/Library/TCC/TCC.db



Spectacle requires that the Accessibility API be enabled

Would you like to open the Universal Access preferences so that you can turn on "Enable access for assistive devices"?

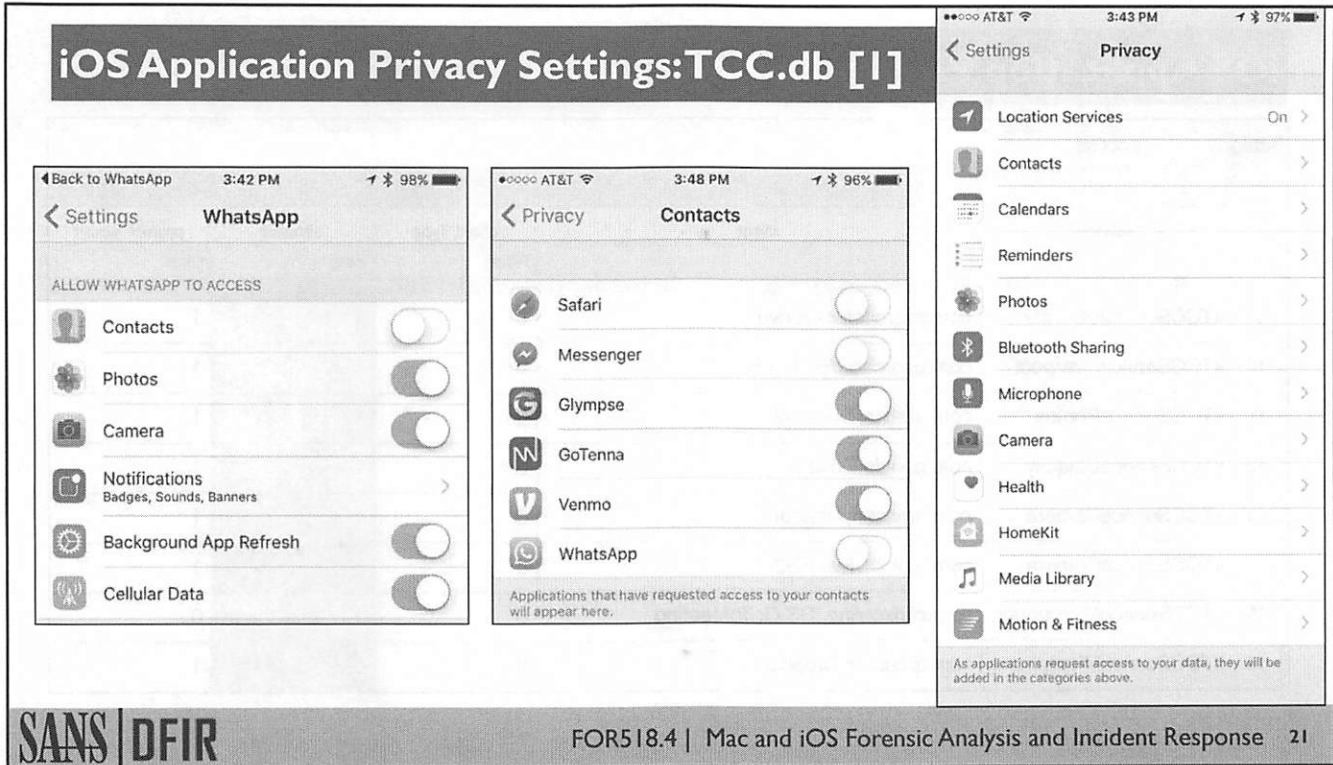
Stop Spectacle

Open Universal Access Preferences

Applications on macOS and iOS ask users which permissions they may have for different capabilities available on the system. This database is named `TCC.db` and is a SQLite database.

On macOS, they are recorded in a couple of different databases: One for the system, and one for each user.

On iOS, they are stored in one database and are available in a backup or physical acquisition.



The TCC.db database contains application privacy settings. This database gets its name from Transparency, Consent, and Control.

This database can be viewed to determine what a specific application has access to. In the screenshot to the right, an application may have access to some of the following items: Location, Contacts, Calendars, Reminders, Photos, Bluetooth, Microphone, Camera, Health/Fitness, Media, HomeKit, etc.

Each application's settings can be viewed in that particular configuration in the Settings area on the device, as shown in the screenshot to the left. The WhatsApp application on this device has access to Photos and Camera, but it does not have access to Contacts.

A certain type of permission can also be viewed to determine which applications have access. The middle screenshot shows an example of this. The Contacts database is only available to the Glympse, GoTenna, and Venmo applications but not Safari, Facebook Messenger, or WhatsApp.

**References:**

- <https://developer.apple.com/videos/play/wwdc2016/705/>
- <https://developer.apple.com/videos/play/wwdc2016/709/>



## iOS Application Privacy Settings:TCC.db [2]

Table:  access

	service	client	client_type	allowed	prompt_count
	Filter	Filter	Filter	Filter	Filter
9	kTCCServiceUbiquity	com.microsoft.skydrive	0	1	1
10	kTCCServiceLiverpool	com.ubercab.UberClient	0	1	1
11	kTCCServicePhotos	com.atebits.Tweetie2	0	1	1
12	kTCCServiceUbiquity	com.google.Drive	0	1	1
13	kTCCServiceCamera	com.amazon.Amazon	0	1	1
14	kTCCServiceCamera	com.atebits.Tweetie2	0	0	1
15	kTCCServiceMicrophone	com.citrixonline.iOS.GoToMeeting	0	1	0
16	kTCCServiceMicrophone	com.dropcam.Dropcam	0	0	1

The TCC.db SQLite database file keeps track of what app has which permissions or “services” as it is labeled in the database.

The screenshot above shows the following:

- Microsoft SkyDrive (com.microsoft.skydrive) has access to the iCloud “Ubiquity” Service
- Uber (com.ubercab.UberClient) has access to the “Liverpool” service
- Twitter (com.atebits.Tweetie2) has access to Photos, but no access to the Camera
- Google Drive (com.google.Drive) has access to iCloud
- Amazon (com.amazon.Amazon) has access to the Camera, probably to scan product codes
- GoToMeeting (com.citrixonline.iOS.GoToMeeting) has access to the microphone
- Nest DropCam (com.dropcam.Dropcam) DOES NOT have access to the microphone


Services may include the following:

- kTCCServiceAll
- kTCCServiceAccessibility
- kTCCServiceCalendar
- kTCCServiceAddressBook
- kTCCServiceLocation
- kTCCServiceReminders
- kTCCServiceFacebook
- kTCCServiceLinkedIn
- kTCCServiceTwitter
- kTCCServiceSinaWeibo
- kTCCServiceLiverpool
- kTCCServiceUbiquity
- kTCCServiceTencentWeibo

### References:

<https://developer.apple.com/videos/play/wwdc2016/705/>  
<https://developer.apple.com/videos/play/wwdc2016/709/>

## macOS Application Privacy Settings: User's TCC.db



```

1 select
2 service, client, allowed, indirect_object_identifier,
3 datetimelast_modified,'unixepoch') as last_modified
4 from access;

```

	service	client	allowed	indirect_object_identifier	last_modified
13	kTCCServiceLiverpool	com.apple.systempreferences	1	UNUSED	2018-11-30 00:29:41
14	kTCCServiceLiverpool	com.apple.Safari	1	UNUSED	2018-11-30 00:36:13
15	kTCCServiceUbiquity	com.getdropbox.dropbox	1	UNUSED	2018-11-30 00:47:55
16	kTCCServiceAppleEvents	com.egnyte.desktopsyncinst...	1	com.apple.systemevents	2018-11-30 01:06:32
17	kTCCServiceAppleEvents	com.egnyte.desktopsyncinst...	1	com.apple.finder	2018-11-30 01:06:37
18	kTCCServiceLiverpool	com.apple.stocks	1	UNUSED	2018-11-30 02:09:32
19	kTCCServiceCamera	com.objective-see.dnd	1	UNUSED	2018-11-30 02:11:54
20	kTCCServiceAppleEvents	just/bin/osascript	1	com.apple.systemevents	2018-11-30 02:38:28
21	kTCCServiceUbiquity	com.apple.reminders	1	UNUSED	2018-12-01 19:16:54
22	kTCCServiceLiverpool	com.apple.Notes	1	UNUSED	2018-12-01 19:17:51
23	kTCCServiceUbiquity	com.apple.Safari	1	UNUSED	2018-12-01 20:35:45
24	kTCCServiceUbiquity	com.apple.Photos	1	UNUSED	2018-12-03 01:08:46
25	kTCCServiceAppleEvents	com.TechSmith.Snagit2019	1	com.apple.Safari	2018-12-03 01:39:38
26	kTCCServiceAddressBook	com.evernote.Evernote	1	UNUSED	2018-12-09 15:23:36
27	kTCCServiceMicrophone	com.logmein.GoToMeeting	1	UNUSED	2018-12-17 22:56:41
28	kTCCServiceCamera	com.logmein.GoToMeeting	1	UNUSED	2018-12-17 22:57:18
29	kTCCServiceUbiquity	com.apple.iChat	1	UNUSED	2018-12-24 15:35:18
30	kTCCServiceLiverpool	com.apple.Maps	1	UNUSED	2018-12-24 18:05:45

The macOS privacy settings found in TCC.db are taken from System Preferences | Security & Privacy | Privacy.

For this macOS user example, two applications are using the Camera permission, GoToMeeting and Objective See's Do Not Disturb. In 10.14, the last modification time was added.

Other privacy settings may use the Address Book, Calendar, Reminders, Diagnostics, and/or various social media credentials such as Facebook and Twitter.

## macOS Application Privacy Settings: System TCC.db



```

1 select
2 service, client, allowed, indirect_object_identifier,
3 datetime(last_modified,'unixepoch') as last_modified
4 from access

```

	service	client	allowed	indirect_object_identifier	last_modified
1	KTCCServicePostEvent	com.divisiblebyzero.Spectacle	1	UNUSED	2018-11-30 01:19:59
2	KTCCServiceAccessibility	com.divisiblebyzero.Spectacle	1	UNUSED	2018-11-30 01:19:59
3	KTCCServicePostEvent	com.getdropbox.dropbox	1	UNUSED	2018-11-30 01:26:55
4	KTCCServiceAccessibility	com.getdropbox.dropbox	1	UNUSED	2018-11-30 02:11:10
5	KTCCServiceAccessibility	com.vmware.fusion	1	UNUSED	2018-11-30 02:38:48
6	KTCCServiceAccessibility	com.techsmith.snagit.capturehelper2019	1	UNUSED	2018-12-01 21:28:41
7	KTCCServiceAccessibility	com.TechSmith.Snagit2019	1	UNUSED	2018-12-01 21:28:41
8	KTCCServicePostEvent	com.TechSmith.Snagit2019	1	UNUSED	2018-12-01 21:27:09
9	KTCCServicePostEvent	com.techsmith.snagit.capturehelper2019	1	UNUSED	2018-12-03 01:37:13
10	KTCCServiceAccessibility	net.sourceforge.sqlitebrowser	1	UNUSED	2018-12-03 03:03:11
11	KTCCServicePostEvent	net.sourceforge.sqlitebrowser	1	UNUSED	2018-12-03 03:51:08
12	KTCCServiceSystemPolicyAllFiles	com.apple.dt.Xcode	1	UNUSED	2018-12-09 15:17:13
13	KTCCServiceSystemPolicyAllFiles	com.macropoint.IExplorer	1	UNUSED	2018-12-09 15:17:21
14	KTCCServicePostEvent	com.logmein.GoToMeeting	1	UNUSED	2018-12-17 22:57:11
15	KTCCServiceAccessibility	com.logmein.GoToMeeting	1	UNUSED	2018-12-17 22:57:11
16	KTCCServiceSystemPolicyAllFiles	com.apple.Terminal	1	UNUSED	2018-12-30 03:06:01

The macOS privacy settings found in TCC.db are taken from System Preferences | Security & Privacy | Privacy.

For this macOS System example, 10.14 added more permissions-based configurations. The one forensic analysts may run into the most often is the 'Full Disk Access' permission. This will limit access to some file system directories and files unless the application is provided access. In the example above, three apps have been given access:

- Terminal.app
- Xcode.app
- iExplorer.app

The database shows this access with the service "KTCCServiceSystemPolicyAllFiles".

## iOS Applications Using Location Services: clients.plist

Key	Type	Value
▼ Root	Dictionary	(95 items)
▶ com.apple.locationd.executable-/usr/libexec/locationd	Dictionary	(1 item)
▶ com.apple.locationd.bundle-/System/Library/LocationBundles/PassbookRelevancy.bundle	Dictionary	(6 items)
▶ com.myfitnesspal.mfp	Dictionary	(7 items)
▶ com.weather.TWC	Dictionary	(10 items)
▶ com.yourcompany.SpeedBoxLite	Dictionary	(9 items)
▶ com.marriott.iphoneprod	Dictionary	(5 items)
▶ com.apple.mobileslideshow	Dictionary	(6 items)
▶ com.ookla.speedtest	Dictionary	(8 items)
▶ com.apple.PassbookUIService	Dictionary	(1 item)
▶ com.facebook.Facebook	Dictionary	(7 items)
▶ com.zillow.ZillowMap	Dictionary	(6 items)
▶ com.apple.AppStore	Dictionary	(7 items)
▶ RunKeeperPro	Dictionary	(9 items)
▶ com.wunderground.weatherunderground	Dictionary	(8 items)
▶ com.apple.springboard	Dictionary	(1 item)
▶ com.zenlabs.c25k	Dictionary	(7 items)
▶ com.google.Maps	Dictionary	(8 items)
▶ com.redfin.redfin	Dictionary	(10 items)
▶ com.apple.mobileme.fmf1	Dictionary	(7 items)
▶ com.apple.locationd.bundle-/System/Library/PrivateFrameworks/HomeKitDaemon.framework	Dictionary	(6 items)
▶ com.glympse.iphone.glympse	Dictionary	(12 items)
▶ com.apple.locationd.bundle-/System/Library/LocationBundles/MotionCalibration.bundle	Dictionary	(7 items)

25

Backup: /root/Library/Caches/locationd/clients.plist

Physical: /private/var/root/Library/Caches/locationd/clients.plist

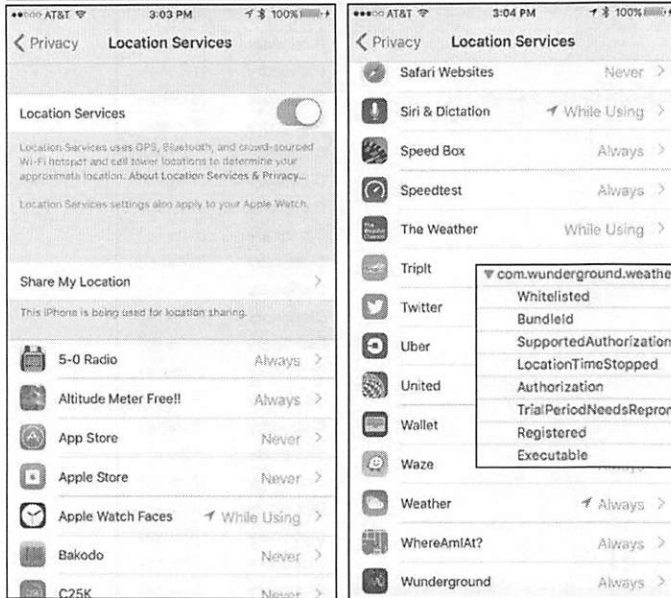
The locationd directory contains several files related to location information stored on an iOS device. The most notable may be the `consolidated.db` file, which was a cause for major concern when its contents were made public to many unsuspecting iOS users. This file contained wireless hotspots and cell tower locations within a certain radius of the device, which was said to better allow the device to calculate its location rather than relying solely on GPS satellite data. This file once contained up to a year's worth of location data, but has since been revised in subsequent iOS firmware releases.

The most valuable file in the locationd directory now is the `clients.plist` file. This file maintains a list of all of the applications that have been granted GPS permissions. This is important because it may clue the examiner to pay extra detail to those applications listed in the file, as GPS coordinates are most likely stored along with other relevant user information.

Reference:

<https://www.apple.com/newsroom/2011/04/27Apple-Q-A-on-Location-Data/>

# iOS App Location Service Authorizations: clients.plist



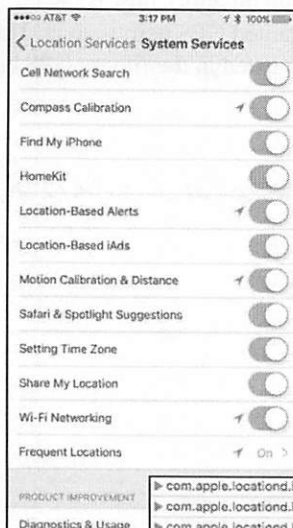
**"Authorization" Key:**  
 1: Never  
 2: While Using  
 4: Always

com.wunderground.weatherunderground	Dictionary	(8 items)
Whitelisted	Boolean	NO
BundleId	String	com.wunderground.weatherunderground
SupportedAuthorizationMask	Number	7
LocationTimeStopped	Number	484,834,287.043168
Authorization	Number	4
TrialPeriodNeedsReprompt	Boolean	NO
Registered	String	/private/var/mobile/Containers/Bundle/Application/39C7
Executable	String	/private/var/mobile/Containers/Bundle/Application/39C7

Each application's location services information is detailed in the `clients.plist` file. This plist file contains the "LocationTimeStopped" key, which keeps track of the time in Mac Epoch of when the application last used Location Services to get the location of the device. The key "Authorization" shows the level of permission for location services for that application.

- 1: Never authorized (no Location Services)
- 2: Only use Location Services when using the application
- 4: Always use Location Services (even when the application is backgrounded)

Along with applications, OS services can also use Location Services to notify the user of traffic conditions or to find their lost iPhone.



PRODUCT IMPROVEMENT	com.apple.locationd.bundle-/System/Library/PrivateFrameworks/FindMyDevice.framework
PRODUCT IMPROVEMENT	com.apple.locationd.bundle-/System/Library/LocationBundles/AppGenius.bundle
PRODUCT IMPROVEMENT	com.apple.locationd.bundle-/System/Library/LocationBundles/NavdLocationBundle!OS.bundle
PRODUCT IMPROVEMENT	com.apple.locationd.bundle-/System/Library/PrivateFrameworks/MobileWiFi.framework
PRODUCT IMPROVEMENT	com.apple.locationd.executable-/System/Library/PrivateFrameworks/Search.framework/searchd
PRODUCT IMPROVEMENT	com.apple.locationd.bundle-/System/Library/LocationBundles/CompassCalibration.bundle
PRODUCT IMPROVEMENT	com.apple.locationd.bundle-/System/Library/LocationBundles/AppleWatchFaces.bundle



## macOS App Location Service Authorizations: clients.plist

The screenshot shows the macOS Security & Privacy window with the Location Services tab selected. The 'Enable Location Services' checkbox is checked. Below it, a list of apps is shown with checkboxes for location services: Weather (checked), Calendar.app (checked), and Siri & Dictation (checked). A 'Details...' button is visible next to the Siri & Dictation entry. A note indicates that an arrow next to an app icon means it has requested location within the last 24 hours.

Overlaid on the right side of the screenshot is a detailed view of the `clients.plist` file. It shows a dictionary structure for `com.apple.weather` and `com.apple.iCal`.

Key	Type	Value
<code>com.apple.weather</code>	Dictionary	(9 items)
<code>Hide</code>	Number	0
<code>Whitelisted</code>	Boolean	NO
<code>LocationTimeStopped</code>	Number	508,209,366.563485
<code>BundleId</code>	String	com.apple.weather
<code>BundlePath</code>	String	/System/Library/CoreServices/Weather.app
<code>Registered</code>	String	/System/Library/CoreServices/Weather.app/Co
<code>Executable</code>	String	/System/Library/CoreServices/Weather.app/Cd
<code>Requirement</code>	String	identifier "com.apple.weather" and anchor app
<code>Authorized</code>	Boolean	YES
<code>com.apple.locationd.bundl...</code>	Dictionary	(6 items)
<code>com.apple.iCal</code>	Dictionary	(8 items)
<code>Hide</code>	Number	0
<code>Whitelisted</code>	Boolean	NO
<code>BundleId</code>	String	com.apple.iCal
<code>Registered</code>	String	
<code>BatchEnabled</code>	Boolean	NO
<code>Authorized</code>	Boolean	YES
<code>Requirement</code>	String	identifier "com.apple.iCal" and anchor apple
<code>LocationTimeStarted</code>	Number	508,295,306.279674

Just like with iOS, the location information is stored in the `clients.plist` file on macOS. This file is located in the `/private/var/db/locationd/` directory.

## Mac and iOS App Testing and Analysis

### On Analysis Host (Mac)

- iproxy (libimobiledevice)
- sqlite3
- SQLite Viewer
- Xcode
- Virtual Machines
- etc.

### On Jailbroken Device

- SSH/SCP
- sqlite3
- Jonathan Levin's Binpack
- fsmon
- cda
- etc.

Third-party (or native!) app analysis can be done in a variety of ways, using both command-line tools as well as GUI-based tools.

On the host analysis system (for this example, we will use a Mac) we can install the `libimobiledevice` (<http://www.libimobiledevice.org/>) suite of utilities. The easiest way to install these tools is to install Homebrew first (<http://brew.sh/>), then use the command “`brew install libimobiledevice`”. This will install lots of tools, including `iproxy` and `ifuse`. `iproxy` can be used to connect via USB to an analysis device (jailbroken w/SSH installed). Then SSH can be used to connect to the device for analysis without the need to connect over Wi-Fi.

```
iproxy 4242 22
ssh root@127.0.0.1 -p 4242
```

Other utilities on the host will include viewing applications like SQLite (GUI or CLI), and Xcode for plist files.

On the iDevice itself, you will want to use a jailbroken device to get full root access to the system. You never know where files are going to be stored! In the test device, you will want to at least install SSH server (some jailbreaks come with it) to get easy access to the device. Once you install SSH, make sure you change the default password from “`alpine`” as soon as possible, to protect from others getting on your test device.

Other utilities include basic UNIX commands contained in Jonathan Levin's binpack ([newosxbook.com](http://newosxbook.com)), `fsmon` for file system monitoring (<https://github.com/nowsecure/fsmon>), and `CDA` (<https://github.com/ay-kay/cda>) to find where app data is stored.

Reference:

<https://www.mac4n6.com/blog/2018/11/25/do-it-live-dynamic-ios-forensic-testing>

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



**SANS DFIR**

DIGITAL FORENSICS & INCIDENT RESPONSE

---

## Section 4: Part 2

# Introduction to SQLite Queries

---

**SANS | DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 30

This page intentionally left blank.

## SQLite Databases

### Database Files

- Main Database – \*.db, \*.sqlite, \*.sqlitedb, \*.storedata, or no file extension!
- Write Ahead Log – \*.wal – May contain additional database transactions
- Shared Memory – \*.shm

### Tables and Columns

### Analysis

- Forensic Viewers vs. Non-Forensic Viewers
- GUI Viewers vs. Command-Line Utilities
- Database Coalescing

SQLite databases are made up of multiple files. The main database may have a variety of file extensions or none at all. Looking for that “SQLite format 3” header can help you determine if a file is a SQLite database. The Write Ahead Log or WAL file is important as it may contain additional database transactions not yet coalesced into the main database file. The Shared Memory file (\*.shm) likely does not contain any database transaction but is used to help facilitate transactions into the WAL file.

As with other databases, SQLite databases can be comprised from multiple tables, each table containing different data in columns. Some databases may have one table, others may have hundreds. Most database schemas are quite different from each other. The data in the columns will be a variety of different datatypes.

There are many types of SQLite viewers out there to use for analysis. Some are GUI-based, others can be run from the command line. Most are not forensic-based viewers and may coalesce the transactions in the WAL file into the main database, like DB Browser for SQLite. This is fine as long as the analyst is aware that is happening or not. Forensic-based utilities may allow the analyst to choose to coalesce or not. Forensic-based browsers may also attempt to recover database entries.

#### References:

<https://www.sqlite.org/docs.html>

<http://www.sqlitetutorial.net>



# Example Database – TCC.db

Table: access New Record

service	client	client_type	allowed	prompt_count	csreq	policy_id	indirect_object_Identifier_type	indirect_object_Identifier	indirect_object_code_Identifier	flags	last_modified
1	com.apple.weather	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543535532
2	com.apple.CloudDocs.MobileDocumentsFileProvider	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543535919
3	com.apple.Automator	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535941
4	com.apple.ScriptEditor2	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535945
5	com.apple.Work.Numbers	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535945
6	com.apple.Preview	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535945
7	com.apple.Work.Keynote	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535945
8	com.apple.BooksX	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535945
9	com.apple.Work.Pages	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535945
10	com.apple.mail	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535953
11	com.apple.QuickTimePlayerX	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535955
12	com.apple.TextEdit	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543535958
13	com.apple.systempreferences	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543537781
14	com.apple.Safari	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543538173
15	com.getdropbox.dropbox	0	1	1	NEEL	NEEL	NEEL	UNUSED	NEEL	0	1543538875
16	com.egnyte.desktopsyncinstaller.Egnyte-Desktop-Sync-Installer	0	1	1	BLDP	NEEL	0	com.apple.systemevents	BLDP	NEEL	1543539992
17	com.egnyte.desktopsyncinstaller.Egnyte-Desktop-Sync-Installer	0	1	1	BLDP	NEEL	0	com.apple.finder	BLDP	NEEL	1543539997
18	com.apple.stocks	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543543772
19	com.objective-see.dnd	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543543914
20	/usr/bin/osascript	1	1	1	BLDP	NEEL	0	com.apple.systemevents	BLDP	NEEL	1543545508
21	com.apple.reminders	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543691814
22	com.apple.Notes	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543691871
23	com.apple.Safari	0	1	1	BLDP	NEEL	NEEL	UNUSED	NEEL	0	1543696545



The database above is an example that will be used for most of the following query examples. This TCC.db database is a good starter database to use because it is relatively simple. We will be looking at one table, 'access' to review the permissions for various applications on a macOS system.

## SELECT Statement – Using DB Browser for SQLite

```
1 select * from access
```

	service	client	client_type	allowed	prompt_count	csreq	policy_id	object_identi	irect_object_identif	object_code	flags	last_modified
1	kTCCServiceUbiquity	com.apple.weather	0	1	1	BLOB	NULL	NULL	UNUSED	NULL	0	1543535532
2	kTCCServiceUbiquity	com.apple.CloudDocs.Mobi...	0	1	1	BLOB	NULL	NULL	UNUSED	NULL	0	1543535919
3	kTCCServiceUbiquity	com.apple.Automator	0	1	1	NULL	NULL	NULL	UNUSED	NULL	0	1543535941
4	kTCCServiceUbiquity	com.apple.ScriptEditor2	0	1	1	NULL	NULL	NULL	UNUSED	NULL	0	1543535945

```
1 select
2 service, client, allowed, indirect_object_identifier, last_modified
3 from access
```

	service	client	allowed	indirect_object_identifier	last_modified
24	kTCCServiceUbiquity	com.apple.Photos	1	UNUSED	1543799326
25	kTCCServiceAppleEvents	com.TechSmith.Sagit2019	1	com.apple.Safari	1543801178
26	kTCCServiceAddressBook	com.evernote.Evernote	1	UNUSED	1544369016
27	kTCCServiceMicrophone	com.logmein.GoToMeeting	1	UNUSED	1545087401
28	kTCCServiceCamera	com.logmein.GoToMeeting	1	UNUSED	1545087438
29	kTCCServiceUbiquity	com.apple.iChat	1	UNUSED	1545665718

The SELECT SQLite statement is the basis of most SQL queries that will extract data from a database. SELECT is used to “select” different items from tables and columns.

In the top example, a wildcard (\*) is used to “select” everything from the “access” table. This is the most broad example, but also useful to filter down columns to what is of investigative importance while in a query building window of DB Browser for SQLite.

The second example shows data being extracted from only certain columns; service, client, allowed, indirect\_object\_identifier, and last\_modified.

It is worth mentioning that once queries start becoming more complicated with table JOINS, you will want to preface the column name with a specific table to differentiate columns that may have the same name across tables (i.e.: access.service, access.client, access.allowed and so on).

Reference:

<https://www.sqlite.org/docs.html>

## SELECT Statement – Using sqlite3

```
Sarahs-Air:com.apple.TCC oompa$ sqlite3 TCC.db
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite> select * from access;
kTCCServiceUbiquity|com.apple.weather|0|1|1|??
||UNUSED||0|1543535532
kTCCServiceUbiquity|com.apple.CloudDocs.MobileDocumentsFileProvider|0|1|1|??
||UNUSED||0|1543535919
kTCCServiceUbiquity|com.apple.Automator|0|1|1|1|UNUSED|0|1543535941
kTCCServiceUbiquity|com.apple.ScriptEditor2|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.iWork.Numbers|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.Preview|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.iWork.Keynote|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.iBooksX|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.iWork.Pages|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.mail|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.QuickTimePlayerX|0|1|1|1|UNUSED|0|1543535945
kTCCServiceUbiquity|com.apple.TextEdit|0|1|1|1|UNUSED|0|1543535945
kTCCServiceLiverpool|com.apple.systempreferences|0|1|1|1|??
||UNUSED||0|1543535945
kTCCServiceLiverpool|com.apple.Safari|0|1|1|1|??
||UNUSED||0|1543535945
kTCCServiceUbiquity|com.getdropbox.dropbox|0|1|1|1|UNUSED|0|1543535945
```

```
Sarahs-Air:com.apple.TCC oompa$ sqlite3 TCC.db
SQLite version 3.26.0 2018-12-01 12:34:55
Enter ".help" for usage hints.
sqlite> .headers on
sqlite> .mode column
sqlite> select * from access;
service          client          client_type  allowed  prompt_count  csreq
-----
kTCCServiceUbiquity  com.apple.weather  0           1         1             ??
kTCCServiceUbiquity  com.apple.CloudDo  0           1         1             ??
kTCCServiceUbiquity  com.apple.Automat  0           1         1
kTCCServiceUbiquity  com.apple.ScriptE  0           1         1
kTCCServiceUbiquity  com.apple.iWork.N  0           1         1
kTCCServiceUbiquity  com.apple.Preview  0           1         1
kTCCServiceUbiquity  com.apple.iWork.K  0           1         1
kTCCServiceUbiquity  com.apple.iBooksX  0           1         1
```

If you want to use `sqlite3` on the command line to do your analysis, that is another option. Performing queries on the command line can be done in a couple of ways.

The examples show a query being performed in the SQLite shell. This is an interactive shell specifically for SQLite databases. The default output in the top example is not exactly the easiest to visually parse. Using a few SQLite shell commands in the second example we can make the output more appealing for analysis.

- `.headers on` – This puts the column names at the top of the output.
- `.mode column` – Puts the data in columns. The column width can be configured; this example shows a default width. The data in the columns may be cut off.

Reference:

<https://www.sqlite.org/docs.html>



## Timestamp Conversion and Column Renaming

```
1 select
2 service, client, allowed, indirect_object_identifier,
3 datetime(last_modified,'unixepoch','localtime')
4 from access
```

	service	client	allowed	indirect_object_identifier	datetime(last_modified,'unixepoch','localtime')
24	KTCCServiceUbiquity	com.apple.Photos	1	UNUSED	2018-12-02 20:08:46
25	KTCCServiceAppleEvents	com.TechSmith.Snagit2019	1	com.apple.Safari	2018-12-02 20:39:38
26	KTCCServiceAddressBook	com.evernote.Evernote	1	UNUSED	2018-12-09 10:23:36
27	KTCCServiceMicrophone	com.logmein.GoToMeeting	1	UNUSED	2018-12-17 17:56:41
28	KTCCServiceCamera	com.logmein.GoToMeeting	1	UNUSED	2018-12-17 17:57:18

```
1 select
2 service, client, allowed, indirect_object_identifier,
3 datetime(last_modified,'unixepoch','localtime') as "Last Modified"
4 from access
```

	service	client	allowed	indirect_object_identifier	Last Modified
24	KTCCServiceUbiquity	com.apple.Photos	1	UNUSED	2018-12-02 20:08:46
25	KTCCServiceAppleEvents	com.TechSmith.Snagit2019	1	com.apple.Safari	2018-12-02 20:39:38

Adding to our initial `SELECT` query, we can manipulate some of the data to make it easier to read. One data type that you will often want to convert is epoch timestamps. Our example stores a Last Modified timestamp in Unix epoch, or number of seconds from 1/1/1970 00:00:00 UTC.

SQLite can convert this by using a `DATETIME` function. The example above is converting from `'unixepoch'` to `'localtime'`. The `'localtime'` modifier will show you the timestamp in local system time. If you wanted to keep it to UTC, remove that modifier, as it will be in UTC by default.

Once converted, the column name is long and unwieldy; this can be changed by using `AS`. The example at the bottom is changing the column name to "Last Modified". This can be useful to put more context to otherwise strange column names. Some database designers are better at naming columns/tables than others.

### References:

<https://www.sqlite.org/docs.html>

[https://www.sqlite.org/lang\\_datefunc.html](https://www.sqlite.org/lang_datefunc.html)



## DISTINCT Keyword and Commenting

```
1 select
2 distinct service
3 --, client, allowed, indirect_object_identifier, datetime(last_modified,'unixepoch','localtime') as "Last Modified"
4 from access
```

	service
1	KTCCServiceAddressBook
2	KTCCServiceAppleEvents
3	KTCCServiceCamera
4	KTCCServiceLiverpool
5	KTCCServiceMicrophone
6	KTCCServiceUbiquity

The `DISTINCT` keyword is useful to find unique values in a column. Using our TCC example, if I wanted to see all the different types of permissions being used in this table I would use `DISTINCT` on the `service` column. There are six distinct permissions for this example.

`DISTINCT` is here used on a single column. When doing analysis, sometimes you just want to quickly see what the values are but not delete everything in your query. This example also shows me commenting out the rest of the query by using `--`.

Reference:

<https://www.sqlite.org/docs.html>

## CASE Expression

```
1 select
2   service, client,
3   case allowed
4     when 0 then 'Not Allowed'
5     when 1 then 'Allowed'
6   end Allowed,
7   indirect_object_identifier,
8   datetime(last_modified,'unixepoch','localtime') as "Last Modified"
9   from access
```

	service	client	Allowed	indirect_object_identifier	Last Modified
30	kTCCServiceLiverpool	com.apple.Maps	Allowed	UNUSED	2018-12-24 13:05:45
31	kTCCServiceAppleEvents	com.blackbagtech.BlackLight	Allowed	com.apple.finder	2019-01-12 19:02:39
32	kTCCServiceAddressBook	com.microsoft.onenote.mac	Allowed	UNUSED	2019-01-18 12:15:28
33	kTCCServiceAddressBook	com.microsoft.Word	Allowed	UNUSED	2019-01-18 12:16:09
34	kTCCServiceAddressBook	com.microsoft.Powerpoint	Allowed	UNUSED	2019-01-18 12:17:15
35	kTCCServiceAppleEvents	com.logitech.presenter	Allowed	com.apple.systempreferences	2019-01-28 10:16:50
36	kTCCServiceAppleEvents	com.logitech.presenter	Allowed	com.apple.Safari	2019-01-28 10:21:03
37	kTCCServiceUbiquity	com.apple.Grab	Allowed	UNUSED	2019-01-31 16:38:24
38	kTCCServiceAddressBook	com.microsoft.Excel	Not Allowed	UNUSED	2019-02-06 20:07:05
39	kTCCServiceAppleEvents	com.logitech.presenter	Allowed	com.google.Chrome	2019-02-20 09:09:02

When you have certain values for different types of entries it can be useful to rename them or perform a certain action on just that value. This is where the CASE expression is handy.

The example above is changing a value of “0” in the allowed column to “Not Allowed” and a value of “1” to “Allowed”. This can help provide context to the extracted data.

Another CASE example might be to change the service to something more human friendly. (i.e.: kTCCServiceUbiquity to “iCloud”)

Reference:

<https://www.sqlite.org/docs.html>

## ORDER BY Clause

```

1 select
2 service, client,
3 case allowed
4 when 0 then 'Not Allowed'
5 when 1 then 'Allowed'
6 end Allowed,
7 indirect_object_identifier,
8 datetime(last_modified,'unixepoch','localtime') as "Last Modified"
9 from access
10 order by "Last Modified" asc

```

	service	client	Allowed	Indirect_object_Identifier	Last Modified
1	kTCCServiceUbiquity	com.apple.weather	Allowed	UNUSED	2018-11-29 18:52:12
2	kTCCServiceUbiquity	com.apple.CloudDocs.MobileDocumentsFileP...	Allowed	UNUSED	2018-11-29 18:58:39
3	kTCCServiceUbiquity	com.apple.Automator	Allowed	UNUSED	2018-11-29 18:59:01
4	kTCCServiceUbiquity	com.apple.ScriptEditor2	Allowed	UNUSED	2018-11-29 18:59:05

```

4 when 0 then 'Not Allowed'
5 when 1 then 'Allowed'
6 end Allowed,
7 indirect_object_identifier,
8 datetime(last_modified,'unixepoch','localtime') as "Last Modified"
9 from access
10 order by "Last Modified" desc

```

	service	client	Allowed	Indirect_object_Identifier	Last Modified
1	kTCCServiceAppleEvents	com.TechSmith.Snagit2019	Allowed	com.google.Chrome	2019-04-28 17:08:44
2	kTCCServiceLiverpool	com.apple.news	Allowed	UNUSED	2019-04-10 23:34:36
3	kTCCServiceUbiquity	com.apple.news	Allowed	UNUSED	2019-04-10 23:34:33
4	kTCCServiceAppleEvents	/usr/bin/osascript	Allowed	com.apple.finder	2019-03-31 09:02:14

An analysis technique could be to order certain columns, this is commonly done on timestamp columns to determine what records were performed before others.

The ORDER BY clause can be used to order temporally in the example above on the “Last Modified” column. The two examples show the same statement but different orderings.

- ASC – Ascending Order, this is the default if this keyword is not provided.
- DESC – Descending Order

Reference:

<https://www.sqlite.org/docs.html>

## WHERE Clause and LIKE Function

```
1 select
2   service, client,
3   case allowed
4     when 0 then 'Not Allowed'
5     when 1 then 'Allowed'
6   end Allowed,
7   indirect_object_identifier,
8   datetime(last_modified,'unixepoch','localtime') as "Last Modified"
9 from access
10 where "Last Modified" like "%2019-%"
11 order by "Last Modified"
```

	service	client	Allowed	indirect_object_identifier	Last Modified
1	kTCCServiceAppleEvents	com.blackbagtech.BlackLight	Allowed	com.apple.finder	2019-01-12 19:02:39
2	kTCCServiceAddressBook	com.microsoft.onenote.mac	Allowed	UNUSED	2019-01-18 12:15:28
3	kTCCServiceAddressBook	com.microsoft.Word	Allowed	UNUSED	2019-01-18 12:16:09
4	kTCCServiceAddressBook	com.microsoft.Powerpoint	Allowed	UNUSED	2019-01-18 12:17:15
5	kTCCServiceAppleEvents	com.logitech.presenter	Allowed	com.apple.systempreferences	2019-01-28 10:16:50
6	kTCCServiceAppleEvents	com.logitech.presenter	Allowed	com.apple.Safari	2019-01-28 10:21:03

Some databases may have thousands of records and filtering is required to find specific records of interest.

Using the `WHERE` clause, we can filter on different pieces of the output. The example shows a filter for “Last Modified” times that start with the string “2019-”. The ‘%’ wildcards are used as part of the string matching, zero or more characters before or after the specified string.

Reference:

<https://www.sqlite.org/docs.html>

## Table JOINS (Using Safari CloudTabs.db)

```
1 select
2   cloud_tab_devices.device_name,
3   cloud_tabs.title, cloud_tabs.url,
4   datetime(cloud_tab_devices.last_modified+978307200,'unixepoch','localtime') as last_modified
5 from cloud_tabs
6 left join cloud_tab_devices where cloud_tabs.device_uuid = cloud_tab_devices.device_uuid
```

	device_name	title	url	last_modified
1	Sarah's MacBook Air	Twitter / Notifications	<a href="https://twitter.com/i/notifications">https://twitter.com/i/notifications</a>	2019-01-01 13:45:50
2	Sarah's MacBook Air	About Secure Boot - Apple Support	<a href="https://support.apple.com/en-us/HT208330">https://support.apple.com/en-us/HT208330</a>	2019-01-01 13:45:50
3	Sarah's MacBook Air	Twitter	<a href="https://twitter.com/">https://twitter.com/</a>	2019-01-01 13:45:50
4	Sarah's MacBook Air	Objective-See's Blog	<a href="https://objective-see.com/blog/blog_0x3C.html">https://objective-see.com/blog/blog_0x3C.html</a>	2019-01-01 13:45:50
5	Sarah's MacBook Air	Jailbreak - The iPhone Wiki	<a href="https://www.theiphonewiki.com/wiki/Jailbreak">https://www.theiphonewiki.com/wiki/Jailbreak</a>	2019-01-01 13:45:50
6	miPhoneX	Koala - Wikipedia	<a href="https://en.m.wikipedia.org/wiki/Koala">https://en.m.wikipedia.org/wiki/Koala</a>	2019-01-01 13:47:51
7	miPhoneX	Mac & iOS Forensic Analysis & Incident Resp...	<a href="https://www.sans.org/course/mac-and-ios-fo...">https://www.sans.org/course/mac-and-ios-fo...</a>	2019-01-01 13:47:51
8	miPhoneX	london - Google Search	<a href="https://www.google.com/search?q=london&amp;ie...">https://www.google.com/search?q=london&amp;ie...</a>	2019-01-01 13:47:51
9	miPhoneX	Apple Park Visitor Center - Apple	<a href="https://www.apple.com/retail/appleparkvisitor...">https://www.apple.com/retail/appleparkvisitor...</a>	2019-01-01 13:47:51

Table JOINS are used to combine data between two tables. These joins are performed on a common piece of data in each table.

The example above shows the CloudTabs.db Safari database. The table cloud\_tabs contains the title and url or synced Safari tabs, while the table cloud\_tab\_devices contains the device\_name, and last\_modified timestamp.

The table join is done with a device\_uuid (not shown) column in each table.

Reference:

<https://www.sqlite.org/docs.html>



## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

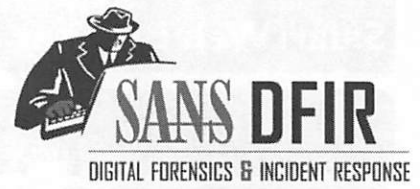
Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



---

## Section 4: Part 3

### Safari Browser

---

This page intentionally left blank.

## Safari Web Browser

Internet History

Cache

Browser Sessions

Thumbnails

Downloads

Cookies

### iOS Physical:

- iOS 8+:  
/private/var/mobile/Containers/Data/Application/<GUID>/Library/Safari/
- iOS 7-: /private/var/mobile/Applications/<GUID>/Library/Safari/
- iOS 6-: /private/var/mobile/Library/Safari/

### iOS Backup/FS:

- iOS 8+: /mobile/Applications/com.apple.mobilesafari/Library/Safari/
- iOS 7: /mobile/Applications/com.apple.mobilesafari/Library/Safari/
- iOS 6: /mobile/Library/Safari/

### iOS Backup/FS:

- ~/Library/Safari/
- ~/Library/Caches/com.apple.Safari/

## Safari and Mobile Safari



Safari data on iOS or macOS devices has changed over time; it was in a plist file and has now migrated to a SQLite database. The various directories and files will be discussed, but it is worth taking a look at the preferences files `com.apple.safarimobile.plist/com.apple.safari.plist` or `RecentSearches.plist` (iOS 7-) for Recent Web Searches (Date/Search String), the `Bookmarks.db/Bookmarks.plist` file for Bookmarks, and the binary cookies files as well.

The web browser native to OS X since 10.3 (before this, the default was Internet Explorer!) is the Safari web browser, while the Mobile Safari browser has been on iOS since the beginning.

Other browsers may also be used, such as Firefox, Google Chrome, and Opera; however, we will only cover the native browser in this class, as other web browsers tend to be platform independent.

Below is an example of recent web searches that are stored in the `com.apple.Safari.plist` file in the user's macOS preferences directory.

▼ RecentWebSearches	Array	(3 items)
▼ Item 0	Dictionary	(2 items)
SearchString	String	penguins
Date	Date	Dec 15, 2014, 3:34:11 AM
▼ Item 1	Dictionary	(2 items)
SearchString	String	mac pro
Date	Date	Dec 15, 2014, 3:33:42 AM
▼ Item 2	Dictionary	(2 items)
SearchString	String	chrome
Date	Date	Dec 14, 2014, 4:50:41 PM

## Safari History: History.db

[macOS 10.10+ | iOS 8+]

- History.db
- Was a plist file!



Clearing history will remove related cookies and other website data.

History will also be removed on other devices signed into your iCloud account.

Clear



Cancel

Clear History

macOS 10.10+ and iOS 8+: Synced iCloud history

History kept for:

- macOS: 1 year (default) ... or 1 month, 2 weeks, 1 week, 1 day, manually
- iOS: ~1 month

macOS:

- 10.10+: ~/Library/Safari/History.db
- 10.9-: ~/Library/Safari/History.plist

iOS Physical:

- iOS 8+:  
/private/var/mobile/Containers/Data/Application/<GUID>/Library/Safari/History.db
- iOS 7-:  
/private/var/mobile/Applications/<GUID>/Library/Safari/History.plist
- iOS 6-: /private/var/mobile/Library/Safari/History.plist

iOS Backup/FS:

- iOS 8+:  
/mobile/Applications/com.apple.mobilesafari/Library/Safari/History.db
- iOS 7:  
/mobile/Applications/com.apple.mobilesafari/Library/Safari/History.plist
- iOS 6: /mobile/Library/Safari/History.plist

The Safari history files changed from History.plist in iOS 7-/macOS 10.9- to the new SQLite database, History.db in iOS 8-/macOS 10.10, and newer. If the user has chosen to sync their Safari history using iCloud in iOS 8-/macOS 10.10+, the history will be synced to all devices using that iCloud login. If the user clears their history on one device, it will be cleared on all devices as shown in the screenshot above.

In iOS, the history is kept for about a month, while on macOS, it is kept for potentially a year unless otherwise configured.



# Safari: Newer History Database—History.db

Origin = 0: Visited on This Device

Origin = 1: Visited on Another iCloud Connected System

```
1 select
2 history_visits.id as "History Item ID",datetime(history_visits.visit_time+978307200,'unixepoch','localtime') as "Visit Time",
3 history_items.url,history_items.visit_count,hex(history_items.daily_visit_counts) as "Daily Visits Hex",
4 history_visits.title,history_visits.load_successful,history_visits.redirect_source,history_visits.redirect_destination,
5 history_visits.origin,history_visits.generation
6 from history_items
7 left outer join history_visits on history_items.id == history_visits.history_item
8 order by "Visit Time"
```

History Item ID	Visit Time	url	visit_count	Daily Visits Hex	title	load_successful	redirect_source	redirect_destination	origin	generation
1	2016-09-07 21:40:09	https://www.google.com/search?q=the+imitation+game&ie=UTF...	1	01000000	the imitation game - Google Search	1	NULL	NULL	0	0
2	2016-09-07 21:40:20	https://en.m.wikipedia.org/wiki/The_imitation_Game	1	01000000	The Imitation Game - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
3	2016-09-07 21:40:58	https://en.m.wikipedia.org/wiki/Benedict_Cumberbatch	1	01000000	Benedict Cumberbatch - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
4	2016-09-07 21:41:26	https://en.m.wikipedia.org/wiki/Alan_Turing	1	01000000	Alan Turing - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
5	2016-09-07 21:41:46	https://en.m.wikipedia.org/wiki/Bletchley_Park	1	01000000	Bletchley Park - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
6	2016-09-07 21:41:54	http://www.bletchleypark.org.uk/	1	01000000	Bletchley Park	1	NULL	NULL	0	0
7	2016-09-07 21:42:21	http://mac4n6.com/	1	01000000		1	NULL	8	1	0
8	2016-09-07 21:42:21	https://www.mac4n6.com/	1	01000000	mac4n6.com	1	7	NULL	1	0
9	2016-09-07 21:47:06	https://m.washingtonpost.com/	1	01000000	Washington Post: Breaking News, World, US, DC News & Analysis - The Washin...	1	10	NULL	0	0
10	2016-09-07 21:47:06	http://washingtonpost.com/	2	02000000		1	NULL	9	0	0
11	2016-09-07 21:47:07	http://washingtonpost.com/	2	02000000	Washington Post: Breaking News, World, US, DC News & Analysis - The Washin...	1	NULL	NULL	0	0

SANS DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 47

Using the SQLite query below, the following data was extracted from the Safari History.db file: The history ID, visit timestamp (Mac Epoch), URL, visit count, hex visit (daily), webpage title, if it loaded successfully, redirection source/destination, the origin, and generation.

Two tables in this database hold the most relevant data:

- history\_items: Contains the URLs, domains, and visit count.
- history\_visits: Contains the Mac Epoch timestamp of when the visits occurred, and the title of the webpage.

If iCloud Safari syncing is enabled, the “origin” column will display a “1” if that URL was viewed on another device, while a “0” means it was viewed from this device.

```
select
history_visits.id as "History Item ID",
datetime(history_visits.visit_time+978307200,'unixepoch','localtime') as
"Visit Time",
history_items.url,
history_items.visit_count,
history_items.daily_visit_counts,
history_items.weekly_visit_counts,
history_visits.title,
history_visits.load_successful,
history_visits.redirect_source,
history_visits.redirect_destination
from history_items
left outer join history_visits on history_items.id ==
history_visits.history_item
order by "Visit Time"
```

```

1 select
2 history_visits.id as "History Item ID",datetime(history_visits.visit_time+978307200,'unixepoch','localtime') as "Visit Time",
3 history_items.url,history_items.visit_count,hex(history_items.daily_visit_counts) as "Daily Visits Hex",
4 history_visits.title,history_visits.load_successful,history_visits.redirect_source,history_visits.redirect_destination,
5 history_visits.origin,history_visits.generation
6 from history_items
7 left outer join history_visits on history_items.id == history_visits.history_item
8 order by "Visit Time"

```

History Item ID	Visit Time	url	visit_count	Daily Visits Hex	title	load_successful	redirect_source	redirect_destination	origin	generation
1	2016-09-07 21:40:09	https://www.google.com/search?q=the+imitation+game&ie=UT...	1	01000000	the imitation game - Google Search	1	NULL	NULL	0	0
2	2016-09-07 21:40:20	https://en.m.wikipedia.org/wiki/The_Imitation_Game	1	01000000	The Imitation Game - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
3	2016-09-07 21:40:58	https://en.m.wikipedia.org/wiki/Benedict_Cumberbatch	1	01000000	Benedict Cumberbatch - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
4	2016-09-07 21:41:26	https://en.m.wikipedia.org/wiki/Alan_Turing	1	01000000	Alan Turing - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
5	2016-09-07 21:41:46	https://en.m.wikipedia.org/wiki/Bletchley_Park	1	01000000	Bletchley Park - Wikipedia, the free encyclopedia	1	NULL	NULL	0	0
6	2016-09-07 21:41:54	http://www.bletchleypark.org.uk/	1	01000000	Bletchley Park	1	NULL	NULL	0	0
7	2016-09-07 21:42:21	http://mac4n6.com/	1	01000000		1	NULL	8	1	0
8	2016-09-07 21:42:21	https://www.mac4n6.com/	1	01000000	mac4n6.com	1	7	NULL	1	0
9	2016-09-07 21:47:06	https://m.washingtonpost.com/	1	01000000	Washington Post: Breaking News, World, US, DC News & Analysis - The Washin...	1	10	NULL	0	0
10	2016-09-07 21:47:06	http://washingtonpost.com/	2	02000000		1	NULL	9	0	0
11	2016-09-07 21:47:07	http://washingtonpost.com/	2	02000000	Washington Post: Breaking News, World, US, DC News & Analysis - The Washin...	1	NULL	NULL	0	0

## Safari Cache: Cache.db [1]

### SQLite Database (iOS: Physical Only)

Contains downloaded cache files (or references to them)

Files with originating location and download date

### macOS 10.7+ (iOS 6+)

- Cache Metadata: cfurl\_cache\_response
- Cache Data: cfurl\_cache\_receiver\_data

macOS: ~/Library/Caches/com.apple.Safari/Cache.db

iOS Physical:

- iOS 8+:  
/private/var/mobile/Containers/Data/Application/<GUID>/Library/Caches/com.apple.mobilesafari/Cache.db
- iOS 7-:  
/private/var/mobile/Applications/<GUID>/Library/Caches/com.apple.mobilesafari/Cache.db
- iOS 6-: /private/var/mobile/Library/Caches/com.apple.mobilesafari/Cache.db

The SQLite database Cache.db contains the downloaded cache files. Each file has a corresponding location and download date.

Databases for macOS 10.6 and 10.7+ systems differ slightly, but each contains a metadata table and a data table in the database.

The Safari Cache on iOS devices appears to only be available on physical acquisitions. Very often cached data is not backed up for space considerations.

## Safari Cache: Cache.db [2]

```
1 select cfurl_cache_response.entry_ID, cfurl_cache_response.request_key, cfurl_cache_response.time_stamp, cfurl_cache_receiver_data.receiver_data
2 from cfurl_cache_response
3 left join cfurl_cache_receiver_data on cfurl_cache_response.entry_ID == cfurl_cache_receiver_data.entry_ID
```

entry_ID	request_key	time_stamp	receiver_data	
915	2046	https://clients1.google.com/complete/search?json=t&noLabels=t&client=iphonesafari&q=mo	2016-06-02 01:11:40	["mo",["movies", "money monster", "movies near me", "montgomery college", "movie theater", "m...
916	2047	https://clients1.google.com/complete/search?json=t&noLabels=t&client=iphonesafari&q=mov	2016-06-02 01:11:43	["mov",["movies", "movies near me", "movie theater", "movietone", "movie times", "movies ou...
917	2048	https://clients1.google.com/complete/search?json=t&noLabels=t&client=iphonesafari&q=movi	2016-06-02 01:11:43	["movi",["movies", "movie times", "movietube", "movie theater...
918	2049	https://clients1.google.com/complete/search?json=t&noLabels=t&client=iphonesafari&q=movie	2016-06-02 01:11:43	["movie",["movies", "movies near me", "movie theater", "movietone", "movie times", "movies ou...
919	2050	https://clients1.google.com/complete/search?json=t&noLabels=t&client=iphonesafari&q=movies	2016-06-02 01:11:43	["movies",["movies", "movies near me", "movies in theaters", "movies 2016", "movies 2015", "movie...
920	2051	https://dalk4zrp4jp3q.cloudfront.net/sprites/6ee1bbb017b910abbc9edb065806bed0_iphone_2x.json	2016-06-16 00:13:58	["iphone_AppleHelp_iOS_2x":{1240,1138,120,120}],["iphone_MLB_2x":{920,6...
921	2052	https://dalk4zrp4jp3q.cloudfront.net/sprites/6ee1bbb017b910abbc9edb065806bed0_iphone_2x.png	2016-06-16 00:13:58	6A38E58D-8D16-498D-B996-8D259D3ED7CA
922	2053	https://github.com/apple-touch-icon-180x180.png	2016-06-16 00:13:58	EA26805A-DD2B-4608-9BD7-D0F6013888CE
923	2054	https://clients1.google.com/complete/search?json=t&noLabels=t&client=iphonesafari&q=JetBlue+Airways+1055	2016-06-16 00:14:02	["JetBlue Airways 1055",[]]
924	2055	https://dalk4zrp4jp3q.cloudfront.net/sprites/767a2e84bda337de2c375557a28e6c5b_iphone_2x.json	2016-06-17 22:42:36	["iphone_AppleHelp_iOS_2x":{1240,1138,120,120}],["iphone_MLB_2x":{920,6...
925	2056	https://dalk4zrp4jp3q.cloudfront.net/sprites/767a2e84bda337de2c375557a28e6c5b_iphone_2x.png	2016-06-17 22:42:36	F2D17B7A-B05B-4924-82CE-7E0484C76B98

The SQLite query below extracts and correlates the cache from two tables: One metadata, the other with data. The `cfurl_cache_response` table contains cache file metadata, including the file cached and its corresponding timestamp. The `cfurl_cache_receiver_data` table contains the cached file. The cached file can be matched up with its metadata by using the `entry_ID` number.

Each cached entry has an associated `entry_ID`, URL (`request_key`), timestamp, and cached data (`receiver_data`). The `receiver_data` column may contain a GUID that can be associated with a file in the `/Caches` and `/fsCachedData` directory, shown next.

```
select
cfurl_cache_response.entry_ID,
cfurl_cache_response.request_key,
cfurl_cache_response.time_stamp,
cfurl_cache_receiver_data.receiver_data
from cfurl_cache_response
left join cfurl_cache_receiver_data on cfurl_cache_response.entry_ID ==
cfurl_cache_receiver_data.entry_ID
```



```

1 select cfurl_cache_response.entry_ID, cfurl_cache_response.request_key, cfurl_cache_response.time_stamp,cfurl_cache_receiver_data.receiver_data
2 from cfurl_cache_response
3 left join cfurl_cache_receiver_data on cfurl_cache_response.entry_ID == cfurl_cache_receiver_data.entry_ID

```

entry_ID	request_key	time_stamp	receiver_data	
915	2046	https://clients1.google.com/complete/search?json=t&nolabels=t&client=iphonesafari&q=mo	2016-06-02 01:11:40	["mo",["movies", "money monster", "movies near me", "montgomery college", "movie theater", "m...
916	2047	https://clients1.google.com/complete/search?json=t&nolabels=t&client=iphonesafari&q=mov	2016-06-02 01:11:43	["mov",["movies", "movies near me", "movie theater", "moviefone", "movie times", "movies ou...
917	2048	https://clients1.google.com/complete/search?json=t&nolabels=t&client=iphonesafari&q=movi	2016-06-02 01:11:43	["movi",["movies", "movie times", "moviefone", "movietube", "movie theater...
918	2049	https://clients1.google.com/complete/search?json=t&nolabels=t&client=iphonesafari&q=movie	2016-06-02 01:11:43	["movie",["movies", "movies near me", "movie theater", "moviefone", "movie times", "movies ou...
919	2050	https://clients1.google.com/complete/search?json=t&nolabels=t&client=iphonesafari&q=movies	2016-06-02 01:11:43	["movies",["movies", "movies near me", "movies in theaters", "movies 2016", "movies 2015", "movie...
920	2051	https://dalk4zrp4jp3q.cloudfront.net/sprites/6ee1bbb017b910abbc9edb065806bed0_iphone_2x.json	2016-06-16 00:13:58	{"iphone_AppleHelp_iOS_2x": [1240,1138,120,120], "iphone_MLB_2x": [920,6...
921	2052	https://dalk4zrp4jp3q.cloudfront.net/sprites/6ee1bbb017b910abbc9edb065806bed0_iphone_2x.png	2016-06-16 00:13:58	6A38E58D-8D16-49BD-B996-8D259D3ED7CA
922	2053	https://github.com/apple-touch-icon-180x180.png	2016-06-16 00:13:58	EA26805A-DD2B-4608-9BD7-D0F6013888CE
923	2054	https://clients1.google.com/complete/search?json=t&nolabels=t&client=iphonesafari&q=JetBlue+Airways+1055	2016-06-16 00:14:02	["JetBlue Airways 1055",[]]
924	2055	https://dalk4zrp4jp3q.cloudfront.net/sprites/767a2e84bda337de2c375557a28e6c5b_iphone_2x.json	2016-06-17 22:42:36	{"iphone_AppleHelp_iOS_2x": [1240,1138,120,120], "iphone_MLB_2x": [920,6...
925	2056	https://dalk4zrp4jp3q.cloudfront.net/sprites/767a2e84bda337de2c375557a28e6c5b_iphone_2x.png	2016-06-17 22:42:36	F2D17B7A-B05B-4924-82CE-7E0484C76B98



## Safari Cache: Cache.db—Referenced Cached Files

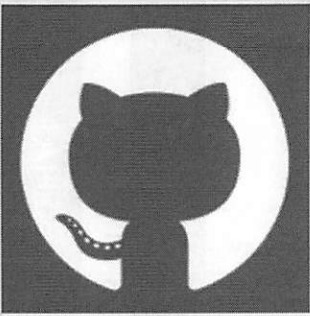
The screenshot shows a file system view of the Safari cache. The 'Library' folder contains a 'Caches' folder (257.7 MB) and an 'fsCachedData' folder. The 'Caches' folder contains 'CloudKit' (6 KB), 'com.apple.iTunesStore' (756 bytes), and 'com.apple.mobilesafari' (12.7 MB). The 'com.apple.mobilesafari' folder contains 'Cache.db' (4.6 MB), 'Cache.db-shm' (33 KB), and 'Cache.db-wal' (756 bytes). The 'fsCachedData' folder contains several files with GUID-based names, including 'EA26805A-DD2B-4608-9BD7-D0F6013888CE'. A preview window for this file is open, showing the GitHub logo and the GUID.

Folder/Item	Size
Library	279.3 MB
Caches	257.7 MB
CloudKit	6 KB
com.apple.iTunesStore	756 bytes
com.apple.mobilesafari	12.7 MB
Cache.db	4.6 MB
Cache.db-shm	33 KB
Cache.db-wal	756 bytes
com.apple.opengl	
fsCachedData	
OC4A60FB-9D90-4117-BF4A-1A55203AF7CB	
1BB8B1DD-3563-4BBB-9611-4B0D270A4AB0	
6A38E58D-8D16-49BD-B996-8D259D3ED7CA	
6E61ACC8-9A3A-4382-81A0-BD706B107505	
021B4B25-2617-4B3D-AC61-8B685455ECB7	
58CCDA35-70BA-4F3E-ADF8-CF70C5E9E425	
EA26805A-DD2B-4608-9BD7-D0F6013888CE	
F2D17B7A-B05B-4924-82CE-7E0484C76B98	
F5CEB104-4EA0-4537-A7DD-D53E11DE3474	

The externally referenced cached files will be located in the `fsCachedData` directory, as shown above. The filenames are the same as the GUID in the Safari `Cache.db` database file.

Library	279.3 MB
Caches	257.7 MB
CloudKit	6 KB
com.apple.iTunesStore	756 bytes
com.apple.mobilesafari	12.7 MB
Cache.db	4.6 MB
Cache.db-shm	33 KB
Cache.db-wal	Zero bytes
com.apple.opengl	
fsCachedData	
0C4A60FB-9D90-4117-BF4A-1A55203AF7CB	
1BB8B1DD-3563-4BBB-9611-4B0D270A4AB0	
6A38E58D-8D16-49BD-B996-8D259D3ED7CA	
6E61ACC8-9A3A-4382-81A0-BD706B107505	
021B4B25-2617-4B3D-AC61-8B685455ECB7	
58CCDA35-70BA-4F3E-ADF8-CF70C5E9E425	
EA26805A-DD2B-4608-9BD7-D0F6013888CE	
F2D17B7A-B05B-4924-82CE-7E0484C76B98	
F5CEB104-4EA0-4537-A7DD-D53E11DE3474	

Open with TextEdit



**EA26805A-  
DD2B-4608-  
9BD7-  
D0F6013888  
CE**

## macOS/iOS: Safari WebKit Cache (~10.12.4+/iOS 10.3) ~/Library/Caches/com.apple.Safari/WebKitCache

Metadata and Data may be stored in a variety of files

### /Version ##/Records

- Per Website Cached Metadata
- /Resources
  - Metadata/Cached Data
- /SubResources
  - Metadata

### /Version ##/Blobs

- Cached Data (Images, HTML, etc.)

Correlate with 20-byte SHA1 hash filenames

On newer versions of Safari (starting in 10.12.4 and iOS 10.3), the cached data is now being stored in the WebKitCache directory in a variety of nested directories and files.

The true format of the metadata files is not known yet but seems to be correlated together with 20-byte filename hashes.

Reference:

<https://webkit.org/blog/7477/new-web-features-in-safari-10-1/>

## Safari: Saved Session Data—macOS LastSession.plist

SessionWindows	Array	(1 item)
Item 0	Dictionary	(14 items)
SelectedTabIndex	Number	0
TabBarHidden	Boolean	NO
DateClosed	Date	Feb 8, 2017, 10:23:40 PM
FavoritesBarHidden	Boolean	YES
IsPopupWindow	Boolean	NO
PrefersReadingListSidebarVisible	Boolean	NO
FullScreenFavoritesBarHidden	Boolean	NO
WindowStateVersion	String	2.0
WindowUUID	String	3638500E-48C8-4F21-9B17-7C355FE278AB
Miniaturized	Boolean	NO
TabStates	Array	(4 items)
Item 0	Dictionary	(14 items)
IsDisposable	Boolean	NO
SessionState	Data	<9a617a80 24aac462 4ac9ff4a 2ed6d01d 15fe8>
AncestorTabIdentifiers	Array	(0 items)
DateClosed	Date	Feb 8, 2017, 10:23:40 PM
SessionStateIsEncrypted	Boolean	YES
TabIndex	Number	0
WindowUUID	String	3638500E-48C8-4F21-9B17-7C355FE278AB
LastVisitTime	Number	508,301,378.142209
TabUUID	String	20DF813F-6392-4B8B-80F3-D5CAD8AA51DF
TabURL	String	https://en.wikipedia.org/wiki/Bletchley_Park
TabIdentifier	Number	9
TabTitle	String	Bletchley Park - Wikipedia
ProcessIdentifier	Number	27,904
IsMuted	Boolean	NO

macOS: ~/Library/Safari/LastSession.plist

The LastSession.plist contains items from the last browsing session. If multiple browsing tabs were open, there will be multiple items showing the visited URLs.

Each tab will have a tab identifier, which will not necessarily be in numeric order if tabs were removed. The TabTitle and TabURL hold the webpage title and URL, respectively. The SessionState key may hold more information in older versions, shown on the next slide. However, since 10.10, this data is now encrypted.

On iOS, the SuspendState.plist file is similar to the LastSession.plist file on macOS: It keeps track of the opened tabs. The SafariStateDocuments key contains a separate item for each tab the user has open. The SafariStateDocumentSessionState key contains the tab history in an embedded plist file. The SafariStatePrivateDocuments key contains the tab information for tabs in “Private Mode”.

Physical: iOS 8+:

/private/var/mobile/Containers/Data/Application/<GUID>/Library/Safari/SuspendState.plist

Backup/FS: iOS 7+:

/mobile/Applications/com.apple.mobilesafari/Library/Safari/SuspendState.plist

Root	Dictionary	(5 items)
SafariStateActiveDocumentIndex	Number	0
SafariStateDocuments	Array	(1 item)
Item 0	Dictionary	(13 items)
SafariStateDocumentUsesPrivateBrowsingStyle	Boolean	NO
SafariStateDocumentLastRenderTreeSize	Number	0
SafariStateDocumentShouldRestoreReader	Boolean	NO
SafariStateDocumentLastViewedTime	Number	494,992,859.421361
SafariStateDocumentBookmark	Number	0
SafariStateDocumentWasOpenedFromLink	Boolean	NO
SafariStateDocumentURL	String	https://m.washingtonpost.com/
SafariStateDocumentUserVisibleURL	String	https://m.washingtonpost.com/
SafariStateDocumentUUID	String	A81ED39F-8DD4-4888-8CC2-71B88AD2FE80
SafariStateDocumentTitle	String	Washington Post: Breaking News, World, US, DC News & Analysis - The Washington Post
SafariStateDocumentReaderViewTopScrollOffset	Number	0
SafariStateDocumentSessionState	Data	<00000002 62706c68 73743030 a2010203 045e5295 6e648572 54726585 53697a65 5>
SafariStateDocumentDisplayingStandaloneImage	Boolean	NO
SafariStatePrivateDocuments	Array	(2 items)
SafariStatePrivateActiveDocumentIndex	Number	1
SafariStateFileVersion	Number	1

▼ Root	Dictionary	(2 items)
SessionVersion	String	1.0
▼ SessionWindows	Array	(1 item)
▼ Item 0	Dictionary	(10 items)
WindowStateVersion	String	2.0
LocationBarHidden	Boolean	NO
▼ TabStates	Array	(2 items)
▼ Item 0	Dictionary	(5 items)
▼ AncestorTabIdentifiers	Array	(0 items)
TabIdentifier	Number	1
TabTitle	String	Apple - Start
TabURL	String	http://www.apple.com/startpage/
SessionState	Data	<00000002 62706c69 73743030 d101025e 53657373 696f6e48 6973746f 7279d203 040
▼ Item 1	Dictionary	(5 items)
▼ AncestorTabIdentifiers	Array	(0 items)
TabIdentifier	Number	8
TabTitle	String	elephants - Google Search
TabURL	String	https://www.google.com/search?client=safari&rls=en&q=elephants&ie=UTF-8&oe=UTF-8
SessionState	Data	<00000002 62706c69 73743030 d101025e 53657373 696f6e48 6973746f 7279d203 040
Miniaturized	Boolean	NO
SelectedTabIndex	Number	1
TabBarHidden	Boolean	NO
PrefersReadingListSidebarVisible	Boolean	NO
WindowContentRect	String	{{0, 4}, {1366, 691}}
FavoritesBarHidden	Boolean	NO
StatusBarHidden	Boolean	YES



## Safari: Closed Tabs – ~/Library/Safari/RecentlyClosedTabs.plist [10.12+]

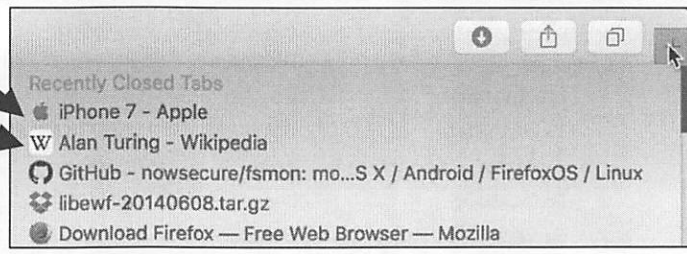
Item 0	Dictionary	(2 items)
PermanentStateType	Number	1
PermanentState	Dictionary	(14 items)
SelectedTabIndex	Number	1
TabBarHidden	Boolean	NO
TabClosed	Date	Oct 23, 2018, 3:50:54 PM
FavoritesBarHidden	Boolean	YES
NSPopUpWindow	Boolean	NO
PreferReadingListToCarv...	Boolean	NO
FullScreenFavoritesBarHidden	Boolean	NO
WindowStateVersion	String	2.0
WindowUUID	String	8A1AF415-43A5-405D-9471-E66AFD88A06
Miniaturized	Boolean	NO
TabDates	Array	(2 items)
Item 0	Dictionary	(14 items)
Disposable	Boolean	NO
SessionState	Date	4C395848F54683265ca7354548829c90a911f680a022064
AnchorTabIndexes	Array	(0 items)
DateClosed	Date	Oct 23, 2018, 3:50:54 PM
SessionStateEncrypted	Boolean	YES
TabIndex	Number	0
WindowUUID	String	8A1AF415-42A8-405D-9471-E66AFD88A06
LastVisitTime	Number	498.9643071879873
TabURL	String	https://www.apple.com/iphone-7
TabIdentifier	Number	19
TabTitle	String	iPhone 7 - Apple
ProcessIdentifier	Number	8,430
IsMuted	Boolean	NO
Item 1	Dictionary	(14 items)
Disposable	Boolean	NO
SessionState	Date	4e6F8F989431224851a3182c8822312901a8b18e6e4e6272
AnchorTabIndexes	Array	(0 items)
DateClosed	Date	Oct 23, 2018, 3:50:54 PM
SessionStateEncrypted	Boolean	YES
TabIndex	Number	1
WindowUUID	String	8A1AF415-42A8-405D-9471-E66AFD88A06
LastVisitTime	Number	498.9643071879873
TabURL	String	https://www.wikipedia.org/wiki/Karl_Turing
TabIdentifier	Number	28
TabTitle	String	Alan Turing - Wikipedia
ProcessIdentifier	Number	8,437
IsMuted	Boolean	NO
WindowContentRect	String	{150, 219, {1280, 760}}
SelectedProcessIdentifier	Number	8,223,372,236,914,775,807
IsVisibleWindow	Boolean	NO

No tab history

Only records tabs when closed

“Recent” tabs

“SessionState” key still encrypted



A new plist with 10.12 is RecentlyClosedTabs.plist, which appears very similar to LastSession.plist; however, it keeps a “recent” history.

Along with newer versions of LastSession.plist, the SessionState key is still encrypted. This plist only keeps track of the current visible tabs that the user had opened, and not the previous history within the tab itself.

▼ Item 30	Dictionary	(2 items)
PersistentStateType	Number	1
▼ PersistentState	Dictionary	(14 items)
SelectedTabIndex	Number	1
TabBarHidden	Boolean	NO
DateClosed	Date	Oct 23, 2016, 3:50:54 PM
FavoritesBarHidden	Boolean	YES
IsPopupWindow	Boolean	NO
PrefersReadingListSidebarVi...	Boolean	NO
FullScreenFavoritesBarHidden	Boolean	NO
WindowStateVersion	String	2.0
WindowUUID	String	8A1AF415-A2A8-40BD-9471-E666AFD88A06
Miniaturized	Boolean	NO
▼ TabStates	Array	(2 items)
▼ Item 0	Dictionary	(14 items)
IsDisposable	Boolean	NO
SessionState	Data	<5393848f 846633c6 cafc7345 b9825c50 a01cffe8 0d026f4
▼ AncestorTabIdentifiers	Array	(0 items)
DateClosed	Date	Oct 23, 2016, 3:50:54 PM
SessionStateIsEncrypted	Boolean	YES
TabIndex	Number	0
WindowUUID	String	8A1AF415-A2A8-40BD-9471-E666AFD88A06
LastVisitTime	Number	498,944,971.675673
TabUUID	String	91DED6F-92D3-4A16-8B29-0BBB2405A642
TabURL	String	http://www.apple.com/iphone-7/
TabIdentifier	Number	15
TabTitle	String	iPhone 7 - Apple
ProcessIdentifier	Number	9,430
IsMuted	Boolean	NO
▼ Item 1	Dictionary	(14 items)
IsDisposable	Boolean	NO
SessionState	Data	<c6799f68 d312c4b5 1a31b2c8 023f3901 ab835dfe efdbf2f2
▼ AncestorTabIdentifiers	Array	(0 items)
DateClosed	Date	Oct 23, 2016, 3:50:54 PM
SessionStateIsEncrypted	Boolean	YES
TabIndex	Number	1
WindowUUID	String	8A1AF415-A2A8-40BD-9471-E666AFD88A06
LastVisitTime	Number	498,944,980.518085
TabUUID	String	3C457AC6-905C-4E01-AA05-97E9F5DAB9F2
TabURL	String	https://en.wikipedia.org/wiki/Alan_Turing
TabIdentifier	Number	23
TabTitle	String	Alan Turing - Wikipedia
ProcessIdentifier	Number	9,437
IsMuted	Boolean	NO
WindowContentRect	String	{{159, 219}, {1280, 716}}
SelectedPinnedTabIndex	Number	9,223,372,036,854,775,807
IsPrivateWindow	Boolean	NO

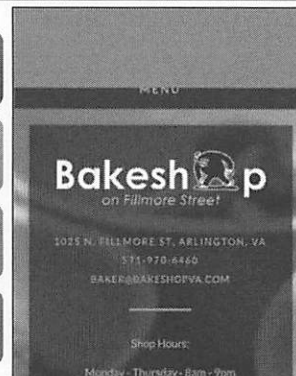
## Safari: Saved Session Data—iOS BrowserState.db [iOS 10+]

Last Viewed Timestamp

UUID: Correlate with /Thumbnails for Screenshots

Private Browsing?

Session Data in BLOB



```
1 select
2 datetime(last_viewed_time+978307200,'unixepoch','localtime') as last_viewed_time,
3 id, uuid, title, url, user_visible_url, order_index, private_browsing, opened_from_link,
4 session_data from tabs order by last_viewed_time
```

	last_viewed_time	id	uuid	title	url	user_visible_url	order_index	private_browsing	opened_from_link	session_data
1	2017-02-09 20:38:50	107	05E26657-8E25-409F-A...	commonwealth joe - Google Search	https://www.google.com/search?q=c...	https://www.google.com/search?q=c...	0	0	0	BLOB
2	2017-02-09 20:38:59	108	E42854A2-A325-4DB6-...	Apple	http://www.apple.com/	http://www.apple.com/	1	0	0	BLOB
3	2017-02-09 20:43:28	109	2A9991DD-D3DD-4AA7-...	Bakeshop	http://www.bakeshopva.com/	http://www.bakeshopva.com/	2	0	0	BLOB
4	2017-02-09 20:44:02	110	BC1B4CAA-13D6-4C49-...	how to keep thing secret - Google Search	https://www.google.com/search?q=h...	https://www.google.com/search?q=h...	0	1	0	BLOB

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 59

iOS 10 Physical:

/private/var/mobile/Containers/Data/Application/<GUID>/Library/Safari/BrowserState.db

iOS 10 Backup:

/mobile/Applications/com.apple.mobilesafari/Library/Safari/BrowserState.db

Like other files that have changed from plists to databases, there is a new format for iOS 10. The BrowserState.db file is a SQLite database that contains relatively the same information as the previous plist files for the browser session data.

The screenshot above shows the last viewed time for the tab contents (in Mac Epoch), tab identifiers, tab metadata, and browser visit information. There are a few more metadata items that are not shown that may be of use to the analyst.

```
select
datetime(last_viewed_time+978307200,'unixepoch','localtime') as
last_viewed_time,
id,
uuid,
title,
url,
user_visible_url,
order_index,
private_browsing,
opened_from_link,
session_data from tabs order by last_viewed_time
```

```

1 select
2 datetime(last_viewed_time+978307200,'unixepoch','localtime') as last_viewed_time,
3 id, uuid, title, url, user_visible_url, order_index, private_browsing, opened_from_link,
4 session_data from tabs order by last_viewed_time

```

	last_viewed_time	id	uuid	title	url	user_visible_url	order_index	private_browsing	opened_from_link	session_data
1	2017-02-09 20:38:50	107	05E25657-8E25-409F-A...	commonwealth joe - Google Search	https://www.google.com/search?q=c...	https://www.google.com/search?q=c...	0	0	0	BLOB
2	2017-02-09 20:38:59	108	E42854A2-A325-4DB6-...	Apple	http://www.apple.com/	http://www.apple.com/	1	0	0	BLOB
3	2017-02-09 20:43:28	109	2A9991DD-D3D0-4AA7-...	Bakeshop	http://www.bakeshopva.com/	http://www.bakeshopva.com/	2	0	0	BLOB
4	2017-02-09 20:44:02	110	BC1B4CAA-13D6-4C49-...	how to keep thing secret - Google Search	https://www.google.com/search?q=h...	https://www.google.com/search?q=h...	0	1	0	BLOB



Previous versions of iOS use a SuspendState.plist file. If the user uses “Private Mode” in Mobile Safari and does not close the tab after browsing, the data shown is still available in the SuspendState.plist file. The data is not cached in the Safari Cache or in the History.db files, however.

The last viewed time, URL, and webpage title are recorded, as well as possible SessionState tab histories.

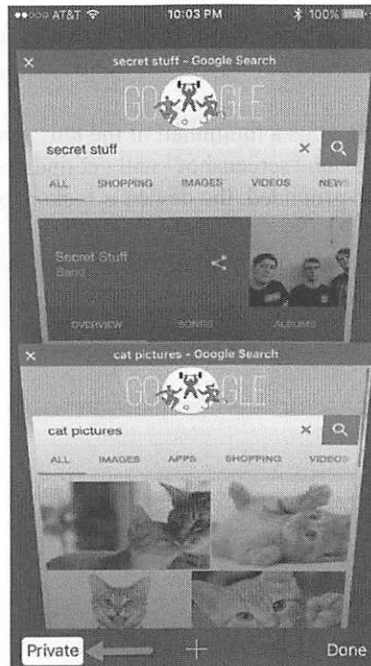
Physical: iOS 8+:

/private/var/mobile/Containers/Data/Application/<GUID>/Library/Safari/SuspendState.plist

Backup/FS: iOS 7+:

/mobile/Applications/com.apple.mobilesafari/Library/Safari/SuspendState.plist

▼ SafariStatePrivateDocuments	Array	(2 items)
▼ Item 0	Dictionary	(13 items)
SafariStateDocumentUsesPrivateBrowsingStyle	Boolean	YES
SafariStateDocumentLastRenderTreeSize	Number	0
SafariStateDocumentShouldRestoreReader	Boolean	NO
SafariStateDocumentLastViewedTime	Number	494,992,920.13129
SafariStateDocumentBookmark	Number	0
SafariStateDocumentWasOpenedFromLink	Boolean	NO
SafariStateDocumentURL	String	https://www.google.com/search?q=secret+stuff&ie=UTF-8&oe=UTF-8&hl=en-us&client=safari
SafariStateDocumentUserVisibleURL	String	https://www.google.com/search?q=secret+stuff&ie=UTF-8&oe=UTF-8&hl=en-us&client=safari
SafariStateDocumentUUID	String	AD15F3B8-39CE-45C2-8205-BBF686246E70
SafariStateDocumentTitle	String	secret stuff - Google Search
SafariStateDocumentReaderViewTopScrollOffset	Number	0
SafariStateDocumentSessionState	Data	<00000002 62706c69 73743030 d2010203 045e5265 6e646572 54726565 53697a65 5e5365
SafariStateDocumentDisplayingStandaloneImage	Boolean	NO
▼ Item 1	Dictionary	(13 items)
SafariStateDocumentUsesPrivateBrowsingStyle	Boolean	YES
SafariStateDocumentLastRenderTreeSize	Number	0
SafariStateDocumentShouldRestoreReader	Boolean	NO
SafariStateDocumentLastViewedTime	Number	494,992,920.131291
SafariStateDocumentBookmark	Number	0
SafariStateDocumentWasOpenedFromLink	Boolean	NO
SafariStateDocumentURL	String	https://www.google.com/search?q=cat+pictures&ie=UTF-8&oe=UTF-8&hl=en-us&client=safari
SafariStateDocumentUserVisibleURL	String	https://www.google.com/search?q=cat+pictures&ie=UTF-8&oe=UTF-8&hl=en-us&client=safari
SafariStateDocumentUUID	String	14C58F5D-D03E-4CE9-BB5B-DC8AEAC69A17
SafariStateDocumentTitle	String	cat pictures - Google Search
SafariStateDocumentReaderViewTopScrollOffset	Number	0
SafariStateDocumentSessionState	Data	<00000002 62706c69 73743030 d2010203 045e5265 6e646572 54726565 53697a65 5e5365
SafariStateDocumentDisplayingStandaloneImage	Boolean	NO





## Mobile Safari: /Thumbnails—Tab Screenshots

Screenshots of tab browsing activity (PNG or KTX)

Each “tab” GUID screenshot updated when:

- Correlate with "SafariStateDocumentUUID"/"UUID" GUID
- Device Locked, Back to Home Screen, Backgrounding Safari App

Includes Private Mode!

Thumbnail
3A370C54-79DC-408B-AFD0-8E59D47044B6.png
7C3F51EC-5914-4EF2-B0B6-863A140F5915.png
14C58F5D-D03E-4CE9-BB5B-DC8AEAC69A17.png
AD15F3B8-39CE-45C2-8205-BBF686246E70.png
C207ACED-F3D5-4401-B670-6C012E9EEBB1.png
D3DA0960-F6AE-4C49-BDB7-EC45D20012C8.png
ED84545D-F5ED-4DC7-A75E-77DF2DE63D6F.png



### Physical:

- iOS 8+:  
/private/var/mobile/Containers/Data/Application/<GUID>/Library/Safari/Thumbnails/
- iOS 7-: /private/var/mobile/Applications/<GUID>/Library/Safari/Thumbnails/
- iOS 6-: /mobile/Library/Caches/com.apple.mobilesafari/Thumbnails/

### Backup/FS:

- iOS 7+:  
/mobile/Applications/com.apple.mobilesafari/Library/Safari/Thumbnails/
- iOS 6-: /mobile/Library/Caches/com.apple.mobilesafari/Thumbnails/

Each GUID in the Thumbnails directory contains a thumbnail of the tab “screenshot”, including those in Private Mode. If the user visits another website, the tab “screenshot” will get updated with a new “screenshot”. This happens when the Safari application is backgrounded, the device is locked, or the user selects a different tab.

In the example above, this device has seven tabs open, some of which are in “Private Mode”.

## Safari Tab Snapshots [10.13]: ~/Library/Caches/com.apple.Safari/TabSnapshots [metadata.db]

Table: snapshot\_metadata

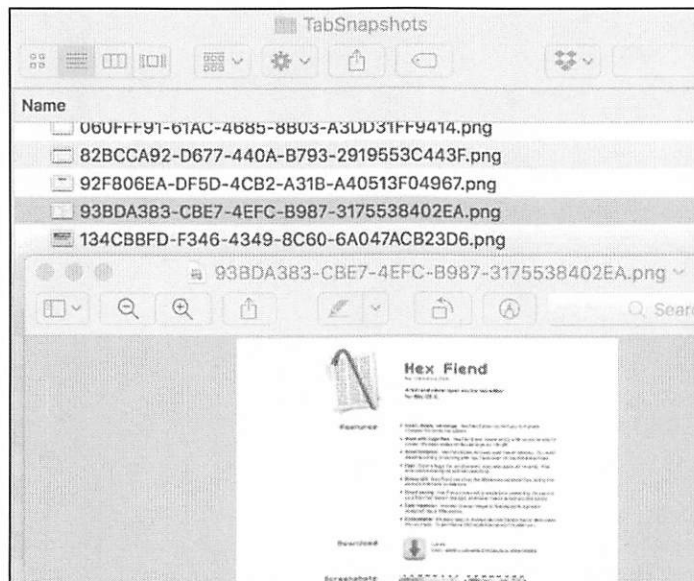
	uuid	date_created	filename	url
1	C1E07649-921F-445D-A085-7B257A664473	532120223.980392	C1E07649-921F-445D-A085-7B257A664473.png	https://www.dropbox.com/install#downloaded
2	B122771B-1A69-4D5C-88DC-E5ABE1FE6088	532120240.299834	B122771B-1A69-4D5C-88DC-E5ABE1FE6088.png	https://www.dropbox.com/sami_login
3	134CBBFD-F346-4349-8C60-6A047ACB23D6	532121564.447249	134CBBFD-F346-4349-8C60-6A047ACB23D6.png	https://www.office.com/?auth=1
4	BOEBD869-84BA-49C9-BC55-086E84C5CBC1	532122873.721492	BOEBD869-84BA-49C9-BC55-086E84C5CBC1.png	https://www.sublimetext.com/2
5	DE8A4EA9-56B9-4966-B98C-D266F0D87C0D	532122950.762556	DE8A4EA9-56B9-4966-B98C-D266F0D87C0D.png	https://www.sublimetext.com/
6	93BDA383-CBE7-4EFC-B987-3175538402EA	532123219.199598	93BDA383-CBE7-4EFC-B987-3175538402EA.png	http://ridiculousfish.com/hexfiend/



Previous tab screenshots may be found in the 10.13 Safari cache directory. These screenshots are not exactly high resolution; however, we can determine what website they are a screenshot of by looking in the related metadata.db SQLite database file.

In the example above, each tab screenshot is assigned a UUID, which is also used in the saved filename. Going to the TabSnapshots directory, we can open and view these screenshots.

Older versions of macOS may have the directory, ~/Library/Caches/com.apple.Safari/Webpage Previews/. These snapshots only appear to be created if the user selected the “Top Sites” button.



## macOS/iOS – Cloud Tabs – CloudTabs.db

```
1 select
2 cloud_tab_devices.device_name,
3 cloud_tabs.title, cloud_tabs.url,
4 datetime(cloud_tab_devices.last_modified+978307200,'unixepoch','localtime') as last_modified
5 from cloud_tabs
6 left join cloud_tab_devices where cloud_tabs.device_uuid = cloud_tab_devices.device_uuid
```

	device_name	title	url	last_modified
1	Sarah's MacBook Air	Twitter / Notifications	<a href="https://twitter.com/i/notifications">https://twitter.com/i/notifications</a>	2019-01-01 13:45:50
2	Sarah's MacBook Air	About Secure Boot - Apple Support	<a href="https://support.apple.com/en-us/HT208330">https://support.apple.com/en-us/HT208330</a>	2019-01-01 13:45:50
3	Sarah's MacBook Air	Twitter	<a href="https://twitter.com/">https://twitter.com/</a>	2019-01-01 13:45:50
4	Sarah's MacBook Air	Objective-See's Blog	<a href="https://objective-see.com/blog/blog_0x3C.html">https://objective-see.com/blog/blog_0x3C.html</a>	2019-01-01 13:45:50
5	Sarah's MacBook Air	Jailbreak - The iPhone Wiki	<a href="https://www.theiphonewiki.com/wiki/Jailbreak">https://www.theiphonewiki.com/wiki/Jailbreak</a>	2019-01-01 13:45:50
6	miPhoneX	Koala - Wikipedia	<a href="https://en.m.wikipedia.org/wiki/Koala">https://en.m.wikipedia.org/wiki/Koala</a>	2019-01-01 13:47:51
7	miPhoneX	Mac & IOS Forensic Analysis & Incident Resp...	<a href="https://www.sans.org/course/mac-and-ios-fo...">https://www.sans.org/course/mac-and-ios-fo...</a>	2019-01-01 13:47:51
8	miPhoneX	london - Google Search	<a href="https://www.google.com/search?q=london&amp;ie...">https://www.google.com/search?q=london&amp;ie...</a>	2019-01-01 13:47:51
9	miPhoneX	Apple Park Visitor Center - Apple	<a href="https://www.apple.com/retail/appleparkvisitor...">https://www.apple.com/retail/appleparkvisitor...</a>	2019-01-01 13:47:51

macOS: ~/Library/Safari/CloudTabs.db  
iOS: [/private/var/]mobile/Library/Safari/CloudTabs.db

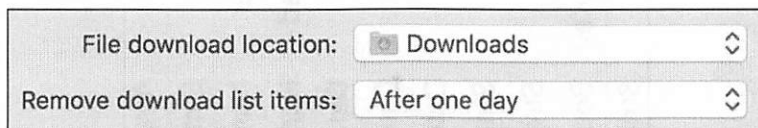
Devices that are syncing a user's Safari web history can be found in the CloudTabs.db database.

The example above shows two devices by hostname: Sarah's MacBook Air and miPhoneX. Each device has a few (non-private) tabs open in their respective Safari browsers. The last modification timestamp is a good place to see how stale these tabs are.

## Safari: macOS Downloads – ~/Library/Safari/Downloads.plist

▼ DownloadHistory	Array	(1 item)
▼ Item 0	Dictionary	(9 items)
DownloadEntryProgressBytesSoFar	Number	66,812,133
DownloadEntryProgressTotalToLoad	Number	66,812,133
DownloadEntryBookmarkBlob	Data	<626f6f6b 70030000 00000410 30000000 00000000 00000000 00000000
DownloadEntryDateAddedKey	Date	Dec 14, 2014, 4:51:26 PM
DownloadEntryDateFinishedKey	Date	Dec 14, 2014, 4:59:07 PM
DownloadEntryIdentifier	String	01856613-E26D-4488-B382-98A2F561782F
DownloadEntryURL	String	https://dl.google.com/chrome/mac/stable/GGRM/googlechrome.dmg
DownloadEntryRemoveWhenDoneKey	Boolean	NO
DownloadEntryPath	String	~/Downloads/googlechrome.dmg

- 10.12+: Default—Remove Downloads after 1 Day
  - Removed from Downloads.plist



The Safari directory contains the Downloads.plist file. This file contains many items that may interest a forensic analyst about the Safari download history:

- **DownloadEntryIdentifier:** Each download has a unique GUID
- **DownloadEntryURL:** A URL is saved showing where the download originated
- **DownloadEntryProgressTotalToLoad, ProgressBytesSoFar:** These keys store the total bytes and downloaded bytes for the download. The downloads file may not always have the same quantity of bytes if the download was canceled or otherwise stopped.
- **DownloadEntryPath:** Where the item was downloaded to; very likely will be in the default Downloads directory (~Downloads)
- **DownloadEntryBookmark/Alias Blob:** Bookmark or Alias data (Aliases are used in 10.6 and 10.7)
- **DownloadEntryDateAddedKey, DownloadEntryDateFinishedKey:** The timestamps of when the download started and finished (10.10+)

10.12 introduced a new default; instead of having the user manually remove items from the Safari Download list, they have instituted a “Remove download list items” configuration item that has “After one day” selected by default. Other options include: “When Safari Quits”, “Upon Successful Download”, or “Manually”.

▼ DownloadHistory	Array	(1 item)
▼ Item 0	Dictionary	(9 items)
DownloadEntryProgressBytesSoFar	Number	66,812,133
DownloadEntryProgressTotalToLoad	Number	66,812,133
DownloadEntryBookmarkBlob	Data	<626f6f6b 70030000 00000410 30000000 00000000 00000000 00000000 00000000
DownloadEntryDateAddedKey	Date	Dec 14, 2014, 4:51:26 PM
DownloadEntryDateFinishedKey	Date	Dec 14, 2014, 4:59:07 PM
DownloadEntryIdentifier	String	01856613-E26D-4488-B382-98A2F561782F
DownloadEntryURL	String	https://dl.google.com/chrome/mac/stable/GGRM/googlechrome.dmg
DownloadEntryRemoveWhenDoneKey	Boolean	NO
DownloadEntryPath	String	~/Downloads/googlechrome.dmg



## Safari: Cookies

Not just for Safari; also other apps!

`Cookies.binarycookies`

`com.apple.Safari.SafeBrowsing.binarycookies`

File signature = “cook”

Proprietary format

- Scripts!

macOS: `~/Library/Cookies/`  
iOS Backup: `/mobile/Library/Cookies/`  
iOS Physical: `/private/var/mobile/Library/Cookies/`

On 10.6 and older systems, the cookies are stored in a plaintext XML property list file. On 10.7+ (iOS 6+) systems, the cookies are stored in the `Cookies.binarycookies` and `com.apple.Safari.SafeBrowsing.binarycookies`. This file starts with the file header “cook”. The file format for these binary cookies is a proprietary format.

### References:

Mari DeGrazia’s absolutely fantastic Google Analytic Cookies Forensics presentation:

<https://github.com/mdegrazia/Presentations>

and her binary cookie parser, which is here: <https://github.com/mdegrazia/Safari-Binary-Cookie-Parser>

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.

---

## Section 4: Part 4

### Apple Mail

---

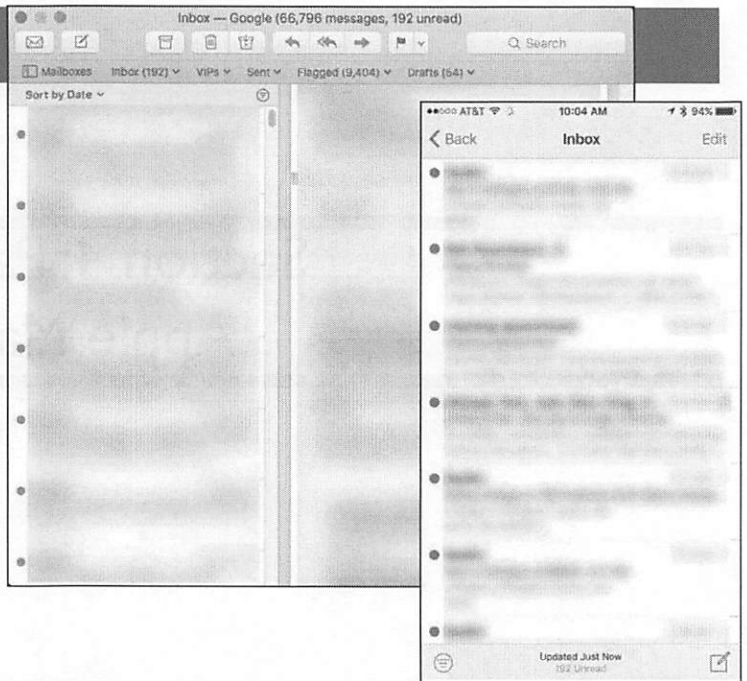
This page intentionally left blank.

## Apple Mail

Directory  
Structure

Email Metadata

Attachments and  
Downloads



The native email application on OS X is Apple Mail. Be sure to keep an eye out for other applications that may also be installed, such as Microsoft Outlook, Thunderbird, and even Lotus Notes!

## iOS Mail Directories

### /mobile/Library/DataAccess

- Mail Account Configuration and General Metadata
- Similar on Backup/File System Extraction

### /mobile/Library/Mail

- Mail Messages and Detailed Metadata
- **Very** different between Backup/File System Extraction and Physical Image

The content of email is very different, depending on what type of acquisition you have. Each acquisition type has two areas where email data is found.

The `DataAccess` directory in `/mobile/Library/` contains general email metadata and email account configuration information. This information is the same for all extractions.

The `Mail` directory in `/mobile/Library` is where things can be very different. Depending on the type of acquisition, you may have great information or you may have little to none.



## iOS Mail: /mobile/Library/DataAccess/

### Directories: Email Accounts

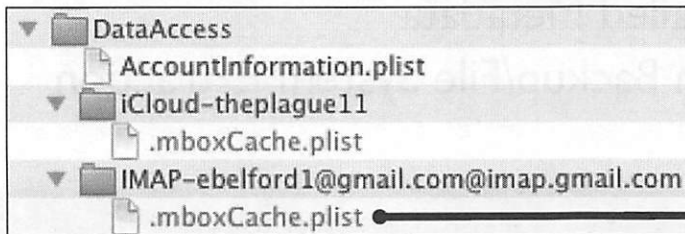
- Email Account Names/Protocol
- Exchange Active Sync (No Account Identity)

### .mboxCache.plist

- Account Organization, Custom Directories

### AccountInformation.plist

- Account GUIDS (Correlate with Accounts Database)



Key	Type	Value
capabilities	Array	(15 items)
mboxes	Array	(4 items)
Item 0	Dictio...	(3 items)
MailboxAttributes	Number	2
MailboxChildren	Array	(7 items)
Item 0	Dictio...	(4 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	All Mail
MailboxPermanentTag	String	\AllMail
Item 1	Dictio...	(4 items)
Item 2	Dictio...	(4 items)
Item 3	Dictio...	(4 items)
Item 4	Dictio...	(4 items)
Item 5	Dictio...	(4 items)
Item 6	Dictio...	(4 items)
MailboxName	String	[Gmail]
Item 1	Dictio...	(3 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	Notes
Item 2	Dictio...	(3 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	Sent Messages
Item 3	Dictio...	(3 items)
MailboxAttributes	Number	0
MailboxChildren	Array	(0 item)
MailboxName	String	INBOX
separator	String	/

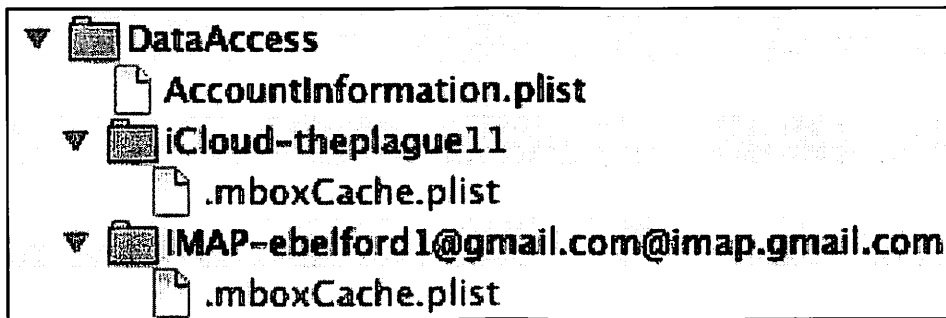
72

The DataAccess directory contains directories named after the email accounts they represent. For example, the iCloud-thePlague11 is an iCloud email account (theplague11@icloud.com), while the other is a Gmail account (ebelford1@gmail.com).

The .mboxCache.plist files contain the organizational structure of each email account. This property list file contains the folder names the email messages are organized into.

In these files, you may find a folder of interest, depending on how the user has organized their email.

The AccountInformation.plist file contains the GUIDs that can be correlated with the Accounts database to determine additional account setup information.



Key	Type	Value
▶ capabilities	Array	(15 items)
▼ mboxes	Array	(4 items)
▼ Item 0	Dictio...	(3 items)
MailboxAttributes	Number	2
▼ MailboxChildren	Array	(7 items)
▼ Item 0	Dictio...	(4 items)
MailboxAttributes	Number	0
▼ MailboxChildren	Array	(0 item)
MailboxName	String	All Mail
MailboxPermanentTag	String	\AllMail
▶ Item 1	Dictio...	(4 items)
▶ Item 2	Dictio...	(4 items)
▶ Item 3	Dictio...	(4 items)
▶ Item 4	Dictio...	(4 items)
▶ Item 5	Dictio...	(4 items)
▶ Item 6	Dictio...	(4 items)
MailboxName	String	[Gmail]
▼ Item 1	Dictio...	(3 items)
MailboxAttributes	Number	0
▼ MailboxChildren	Array	(0 item)
MailboxName	String	Notes
▼ Item 2	Dictio...	(3 items)
MailboxAttributes	Number	0
▼ MailboxChildren	Array	(0 item)
MailboxName	String	Sent Messages
▼ Item 3	Dictio...	(3 items)
MailboxAttributes	Number	0
▼ MailboxChildren	Array	(0 item)
MailboxName	String	INBOX
separator	String	/

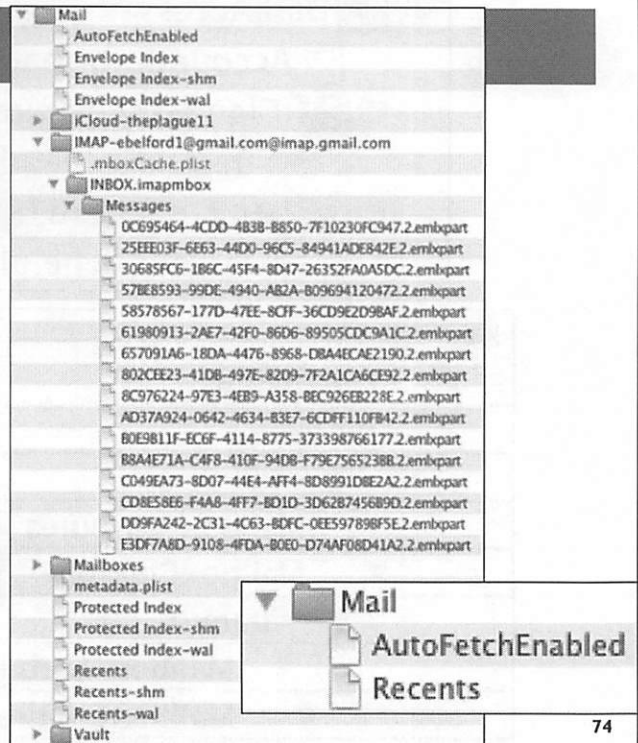
## iOS Mail: /mobile/Library/Mail

### Physical

- Cached Mail Messages
- Envelope Index
  - Message Metadata
- Protected Index
  - Message Sender, Subject, Summary
- Recents Database

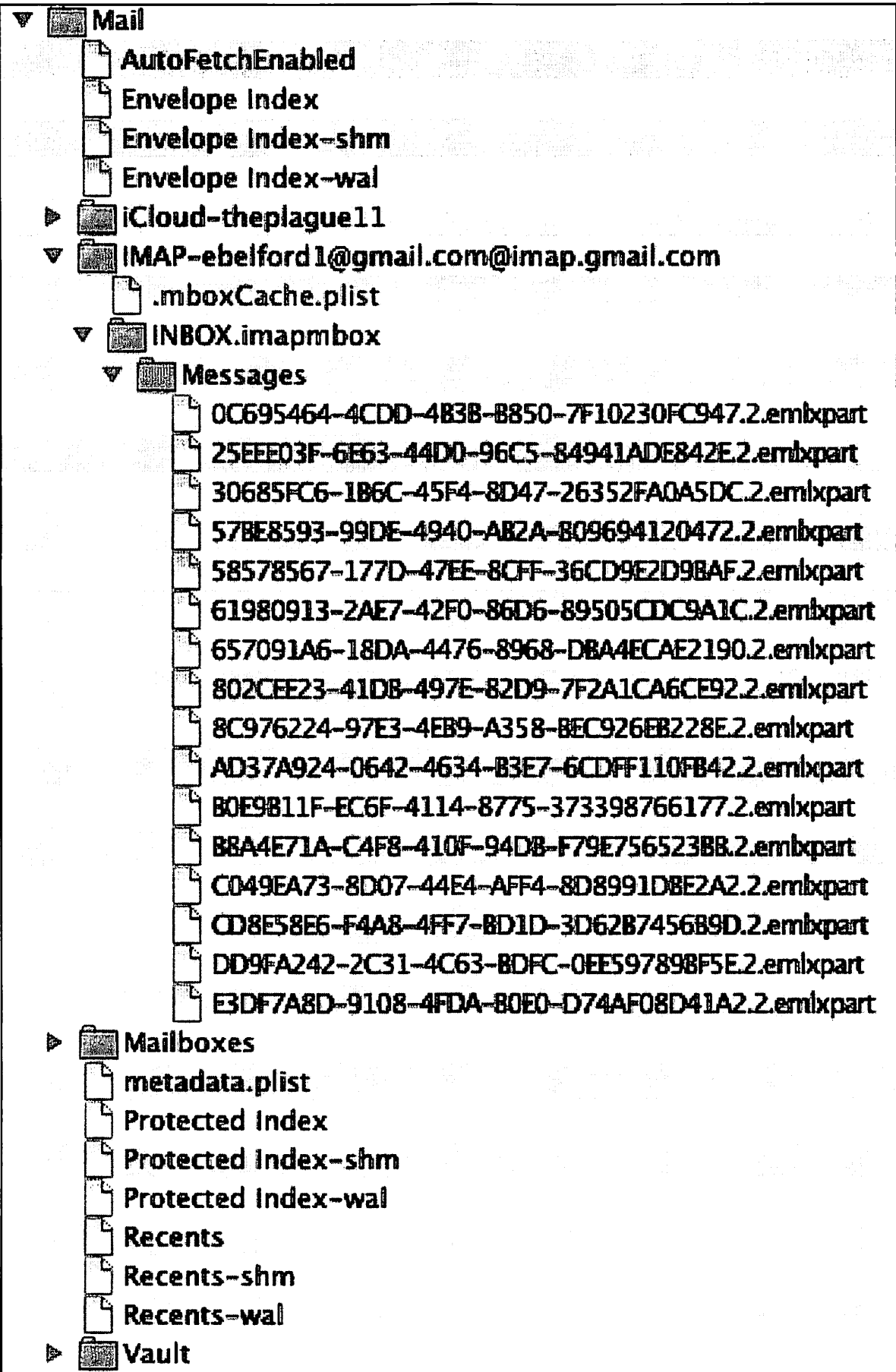
### Backup/FS Extraction

- Recents Database



The Mail directory contains lots of information if you acquired a full physical (decrypted) image. Similar to an OS X system, you will have full email messages, the Recents database, and detailed message metadata in the Envelope Index databases.

If you only have a Backup or File System extraction, you will only get the Recents database, which contains information about recently messaged contacts, similar to the same database on OS X.



## macOS Mail:V3/V4/V5 – ~/Library/Mail/V#/ [10.11 V3] [10.12 V4] [10.13 V5] [10.14 V6]

### Directory for each email account

- GUID directory name
- Can be correlated in the Accounts[3|4].sqlite database

### MailData: Mail Application Data

```
Sarahs-Air:V6 oompa$ pwd
/Users/oompa/Library/Mail/V6
Sarahs-Air:V6 oompa$ ls -la
total 0
drwxr-xr-x@  8 oompa  staff  256 Nov 29 19:37 .
drwx-----@  4 oompa  staff  128 Nov 29 19:37 ..
drwxr-xr-x@ 10 oompa  staff  320 Jan  1 14:01 49285778-1207-4AB3-8F55-EE3A063B6743
drwxr-xr-x@  5 oompa  staff  160 Dec 31 22:02 A1A1F4DD-809F-48A5-A15B-6714993A3D63
drwxr-xr-x@ 10 oompa  staff  320 Jan  1 14:01 C59BB561-B88F-4D21-B709-9F72F84DFF80
drwxr-xr-x@ 19 oompa  staff  608 Jan  1 13:41 F37CEC82-FE8E-46B6-B236-CBF1D57D5D85
drwxr-xr-x@  4 oompa  staff  128 Jan  1 11:48 F8B67AC8-357E-477E-93C3-82B9FBE1BDBB
drwxr-xr-x@ 21 oompa  staff  672 Jan  1 13:55 MailData
```

The “version” number of the Mail directory has changed over the last few macOS versions:

- 10.11 = V3
- 10.12 = V4
- 10.13 = V5
- 10.14 = V6

In these directories, you will find GUID-named directories. These can be correlated with various email accounts by using the Accounts3.sqlite and Accounts4.sqlite database files.

Each of these GUID-based files contains a similar structure to that found in older versions, as we will see in the next few slides.



macOS Mail: Email Accounts : ~/Library/Mail/V#[GUID]/

Each \*.mbox file is a mailbox

An account may have multiple mailboxes

- Inbox
- Sent Messages
- [Gmail]
- Deleted Messages
- Notes
- Drafts
- User Created Mailboxes

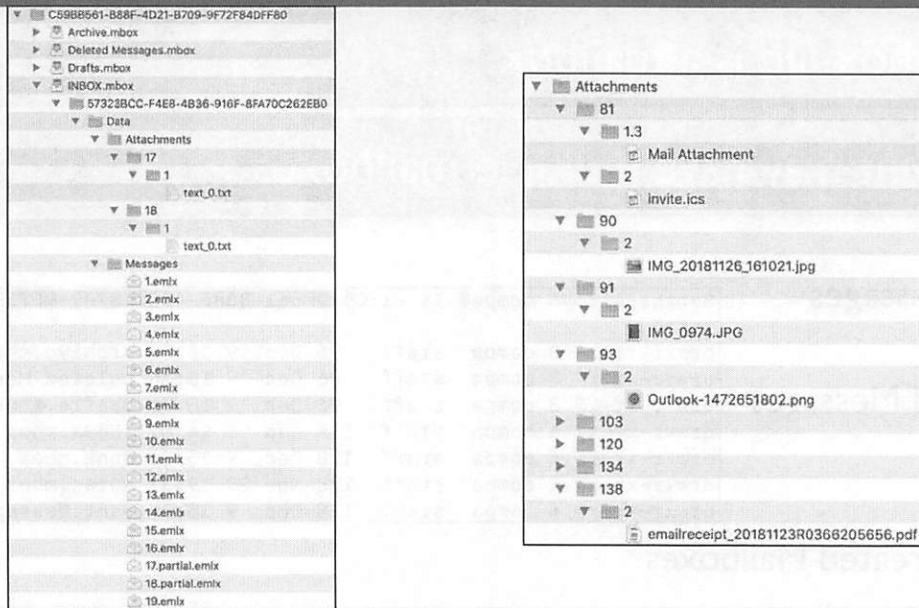
```
Sarahs-Air:V6 oompa$ ls -l C59BB561-B88F-4D21-B709-9F72F84DFF80/  
total 0  
drwxr-xr-x@ 3 oompa  staff  96 Dec  9 15:36 Archive.mbox  
drwxr-xr-x@ 3 oompa  staff  96 Dec  9 15:36 Deleted Messages.mbox  
drwxr-xr-x@ 3 oompa  staff  96 Dec  9 15:36 Drafts.mbox  
drwxr-xr-x@ 4 oompa  staff 128 Jan  1 14:02 INBOX.mbox  
drwxr-xr-x@ 4 oompa  staff 128 Dec  9 15:36 Junk.mbox  
drwxr-xr-x@ 4 oompa  staff 128 Nov 29 19:37 Notes.mbox  
drwxr-xr-x@ 4 oompa  staff 128 Dec  9 15:36 Sent Messages.mbox
```

Each Mailbox Account GUID directory contains one or more mailbox (.mbox) directories.

The screenshot shows three mailbox (.mbox) directories.

- Archive.mbox
- Deleted Messages.mbox
- Drafts.mbox
- INBOX.mbox
- Junk.mbox
- Notes.mbox
- Sent Messages.mbox

## Apple Mail: Mailbox (.mbox)



The GUID directory contains the raw email messages (.emlx) and email metadata in the Messages directory.

- Messages: Contains the raw email messages (.emlx), with an appended property list containing message metadata.
- Attachments: Contains the message file attachments.

## Apple Mail: Envelope Index

SQLite Database: Contains Email Metadata

Messages, Attachments, Recipients, Threads, Subjects, Mailboxes, etc.

```
1 select messages.rowid, read, flagged, deleted,
2 datetime(date_sent,'unixepoch','localtime') as date_sent, datetime(date_received,'unixepoch','localtime') as date_received,
3 datetime(date_created,'unixepoch','localtime') as date_created, datetime(date_last_viewed,'unixepoch','localtime') as date_last_viewed,
4 mailboxes.url, addresses.address, subjects.subject, snippet from messages
5 left join addresses on messages.sender == addresses.ROWID
6 left join subjects on messages.subject == subjects.ROWID
7 left join mailboxes on messages.mailbox == mailboxes.ROWID
```

	ROWID	read	flagged	deleted	date_sent	date_received	date_created	date_last_viewed	url	address	subject	snippet
20736	104757	1	0	0	2017-02-10 20:43:48	2017-02-10 20:43:50	2017-02-10 20:43:52	NULL	imap://F9B...			
20737	104788	1	1	0	2017-02-11 01:38:22	2017-02-11 01:38:27	2017-02-11 02:15:50	NULL	imap://F9B...			
20738	104785	1	0	1	2017-02-11 07:56:56	2017-02-11 07:56:58	2017-02-11 08:32:32	2017-02-11 08:34:08	imap://F9B...			
20739	104786	1	0	1	2017-02-11 07:57:52	2017-02-11 07:57:58	2017-02-11 08:32:32	2017-02-11 08:34:12	imap://F9B...			
20740	104787	1	0	0	2017-02-11 08:00:48	2017-02-11 08:00:55	2017-02-11 08:32:32	2017-02-11 08:34:27	imap://F9B...			
20741	104788	1	0	0	2017-02-11 08:24:50	2017-02-11 08:24:52	2017-02-11 08:32:32	2017-02-11 10:00:07	imap://F9B...			
20742	104795	1	0	1	2017-02-11 08:48:56	2017-02-11 08:49:06	2017-02-11 08:49:05	2017-02-11 10:00:07	imap://F9B...			

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 79

The SQLite database `Envelope Index`, located in the `MailData` directory, contains indexed mail data.

- The `addresses` table contains all the indexed email addresses and associated contact names.
- The `attachments` table contains the name of each attachment.
- The `mailboxes` table contains data for each mailbox, including total messages and unread messages.
- The `messages` table contains metadata for each email message, including To, From, Subject, email timestamps, and if the message was read or not.
- The `subjects` table contains the email subject for each email message.

In the screenshot above, the actual email can be found by looking for the `ROWID`. The message will be named `104757.emlx`. Related attachments will also have the associated number.

Emails that have been read will contain a '1' in the `read` column, as will messages that have been `flagged` or `deleted`, in their respective columns. The `url` column shows the path to the email account—this will contain the GUID associated with the account (also found in the `Accounts` database). The `address` column shows the sender's email address.

```
select messages.rowid, read, flagged, deleted,
datetime(date_sent,'unixepoch','localtime') as date_sent,
datetime(date_received,'unixepoch','localtime') as date_received,
datetime(date_created,'unixepoch','localtime') as date_created,
datetime(date_last_viewed,'unixepoch','localtime') as date_last_viewed,
mailboxes.url, addresses.address, subjects.subject, snippet from messages
left join addresses on messages.sender == addresses.ROWID
left join subjects on messages.subject == subjects.ROWID
left join mailboxes on messages.mailbox == mailboxes.ROWID
```

```

1 select messages.rowid, read, flagged, deleted,
2 datetime(date_sent,'unixepoch','localtime') as date_sent, datetime(date_received,'unixepoch','localtime') as date_received,
3 datetime(date_created,'unixepoch','localtime') as date_created, datetime(date_last_viewed,'unixepoch','localtime') as date_last_viewed,
4 mailboxes.url, addresses.address, subjects.subject, snippet from messages
5 left join addresses on messages.sender == addresses.ROWID
6 left join subjects on messages.subject == subjects.ROWID
7 left join mailboxes on messages.mailbox == mailboxes.ROWID

```

	ROWID	read	flagged	deleted	date_sent	date_received	date_created	date_last_viewed	url	address	subject	snippet
20736	104757	1	0	0	2017-02-10 20:43:48	2017-02-10 20:43:50	2017-02-10 20:43:52	NULL	imap://F98...			
20737	104766	1	1	0	2017-02-11 01:38:22	2017-02-11 01:38:27	2017-02-11 02:15:50	NULL	imap://F98...			
20738	104785	1	0	1	2017-02-11 07:56:55	2017-02-11 07:56:58	2017-02-11 08:32:32	2017-02-11 08:34:08	imap://F98...			
20739	104786	1	0	1	2017-02-11 07:57:52	2017-02-11 07:57:58	2017-02-11 08:32:32	2017-02-11 08:34:12	imap://F98...			
20740	104787	1	0	0	2017-02-11 08:00:49	2017-02-11 08:00:55	2017-02-11 08:32:32	2017-02-11 08:34:27	imap://F98...			
20741	104788	1	0	0	2017-02-11 08:24:50	2017-02-11 08:24:52	2017-02-11 08:32:32	2017-02-11 10:00:07	imap://F98...			
20742	104795	1	0	1	2017-02-11 08:48:56	2017-02-11 08:49:06	2017-02-11 08:49:05	2017-02-11 10:00:07	imap://F98...			

## macOS Mail: Attachments

### “Quick Look”

- ~/Library/Mail Downloads/
- ~/Library/Containers/com.apple.mail/Data/Library/Mail Downloads/

### “Save”

- ~/Downloads

### Extended Attributes



A user can view an attachment in a couple of ways:

- The “Save” button will save the attachment in the default Downloads directory, very likely ~/Downloads.
- The “Quick Look” button opens the attachment for the viewer to see quickly and saves the attachment in the ~/Library/Mail Downloads/ directory. The sandbox directory may also be used ~/Library/Containers/com.apple.mail/Data/Library/Mail Downloads/.



## macOS Mail Attachments: Extended Attributes

Date Received

Date Sent

Remote Attachment

Message Where Froms

Quarantine

```
bit:Mail Downloads oompa$ pwd
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/Mail Downloads
bit:Mail Downloads oompa$ xattr -xl 83EE57F9-4985-47CC-B83E-403A53843842/download.jpeg
com.apple.metadata:com_apple_mail_dataReceived:
00000000 62 70 6C 69 73 74 30 30 33 41 BE 4F 7C 8D 00 00 |plist003A.0|...|
00000010 00 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 |.....|
00000020 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 11 .....|
00000032
com.apple.metadata:com_apple_mail_dataSent:
00000000 62 70 6C 69 73 74 30 30 33 41 BE 4F 7C 8D 00 00 |plist003A.0|...|
00000010 00 00 00 00 00 00 00 00 01 01 00 00 00 00 00 00 |.....|
00000020 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 11 .....|
00000032
com.apple.metadata:com_apple_mail_isRemoteAttachment:
00000000 62 70 6C 69 73 74 30 30 33 41 BE 4F 7C 8D 00 00 |plist00.....|
00000010 01 01 00 00 00 00 00 00 00 00 01 00 00 00 00 00 |.....|
00000020 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
0000002a
com.apple.metadata:kMDItemWhereFroms:
00000000 42 70 6C 69 73 74 30 30 A3 01 02 03 5F 10 21 53 |plist00.....|S|
00000010 61 72 41 69 20 45 64 77 61 72 64 73 26 3C 6F 6F |arah Edwards <oo|
00000020 6D 70 61 48 63 73 68 2E 72 69 74 2E 65 64 75 3E |mna@csh.rut.edu|
00000030 5E 43 61 74 78 20 26 20 43 6F 66 46 45 65 21 5F |^Cats & Coffee!|
00000040 18 3E 6D 65 73 73 61 67 65 3A 25 33 43 34 39 43 |->message:43C49C|
00000050 42 41 34 33 34 2D 30 37 39 41 2D 34 38 42 32 2D |BAA34-B79A-48B2-|
00000060 41 36 41 05 2D 44 30 34 38 45 36 39 33 39 38 36 |A6A5-D84E693086|
00000070 41 48 63 73 68 2E 72 69 74 2E 65 64 75 25 33 45 |A@csh.rut.edu@SE|
00000080 08 0C 3F 00 00 00 00 00 00 01 01 00 00 00 00 |..0?.....|
00000090 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000A0 00 00 00 00 .....|
000000A4
com.apple.metadata:kMDLabel_r6h1m73c2owhai3h2gch5251a:
00000000 F2 45 74 8D FD 48 E2 EA F8 78 B7 76 9E FE 58 D1 |.Et..H...x.v..[|
00000010 1B A1 8B BA F9 C7 20 9D 76 40 A3 80 AP 52 03 2B |..... @...R +|
00000020
0000w280 10 F4 3a 1f 5v 2c 25 u9 52 .....|..1..MYR|
0000w289
com.apple.quarantine:
00000000 30 30 38 32 38 35 38 39 66 34 35 32 34 38 4D 61 |0002;589f4624;Ma|
00000010 69 6C 3B |ll|
00000013
```

Each email attachment downloaded will have a set of metadata stored in its extended attributes. The example above shows the data from the previous slide's image attachment.

The Date Received, Sent, and whether it is a remote attachment are all stored in a binary plist. The email message "Where Froms" data shows the specific email message information, while the quarantine information shows where it came from (and when!).

```

bit:Mail Downloads oompa$ pwd
/Users/oompa/Library/Containers/com.apple.mail/Data/Library/Mail Downloads
bit:Mail Downloads oompa$ xattr -xl 83EE57F9-4985-47CC-BB3E-403A53843842/download.jpeg
com.apple.metadata:com_apple_mail_dateReceived:
00000000 62 70 6C 69 73 74 30 30 33 41 BE 4F 7C 8D 00 00 |bplist003A.0|...|
00000010 00 08 00 00 00 00 00 00 01 01 00 00 00 00 00 00 |.....|
00000020 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 11 |..|
00000032
com.apple.metadata:com_apple_mail_dateSent:
00000000 62 70 6C 69 73 74 30 30 33 41 BE 4F 7C 8C 00 00 |bplist003A.0|...|
00000010 00 08 00 00 00 00 00 00 01 01 00 00 00 00 00 00 |.....|
00000020 00 01 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 11 |..|
00000032
com.apple.metadata:com_apple_mail_isRemoteAttachment:
00000000 62 70 6C 69 73 74 30 30 08 08 00 00 00 00 00 00 |bplist00.....|
00000010 01 01 00 00 00 00 00 00 00 01 00 00 00 00 00 00 |.....|
00000020 00 00 00 00 00 00 00 00 00 00 09 |.....|
0000002a
com.apple.metadata:kMDItemWhereFroms:
00000000 62 70 6C 69 73 74 30 30 A3 01 02 03 5F 10 21 53 |bplist00....!S|
00000010 61 72 61 68 20 45 64 77 61 72 64 73 20 3C 6F 6F |arah Edwards <oo|
00000020 6D 70 61 40 63 73 68 2E 72 69 74 2E 65 64 75 3E |mpa@csh.rit.edu>|
00000030 5E 43 61 74 73 20 26 20 43 6F 66 66 65 65 21 5F |^Cats & Coffee!_|
00000040 10 3E 6D 65 73 73 61 67 65 3A 25 33 43 34 39 43 |.>message:%3C49C|
00000050 42 41 34 33 34 2D 30 37 39 41 2D 34 38 42 32 2D |BA434-079A-48B2-|
00000060 41 36 41 35 2D 44 30 34 38 45 36 39 33 30 38 36 |A6A5-D048E693086|
00000070 41 40 63 73 68 2E 72 69 74 2E 65 64 75 25 33 45 |A@csh.rit.edu%3E|
00000080 08 0C 30 3F 00 00 00 00 00 00 01 01 00 00 00 00 |..0?.....|
00000090 00 00 00 04 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000A0 00 00 00 80 |....|
000000a4
com.apple.metadata:kMDLabel_r6hlhm73c2owhai3h2gch5251a:
00000000 F2 45 74 8D FD 48 E2 EA F8 78 B7 76 9E FE 5B D1 |.Et..H...x.v..|.|
00000010 1B A1 8B B8 F9 C7 20 9D 36 40 A3 80 A8 52 03 2B |. .... 6@...R +|
00000020 20 5C 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000030 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000040 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000050 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000060 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000070 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000080 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000090 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000000F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000100 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000110 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000120 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000130 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000140 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000150 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000160 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000170 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000180 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000190 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001A0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001B0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001C0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001D0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001E0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
000001F0 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 |.....|
00000200 10 F4 31 1F 59 2C 25 09 52 |..1.,%YR|
00000289
com.apple.quarantine:
00000000 30 30 38 32 3B 35 38 39 66 34 35 32 34 3B 4D 61 |0082;589f4524;Ma|
00000010 69 6C 3B |il;|
00000013

```



**SANS DFIR**

DIGITAL FORENSICS & INCIDENT RESPONSE

---

## Lab 4.1

# Safari and Mail

---

This page intentionally left blank.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.

---

## Section 4: Part 5

### Communication

---

This page intentionally left blank.



## Communication

iChat and Messages App

FaceTime

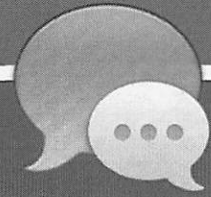
SMS/iMessage

Call History

Voicemail

This page intentionally left blank.

## iChat and Messages Application



Jabber (XMPP)

iMessage (iOS: Messages)

SMS

Deprecated: Yahoo!, ICQ, AOL AIM, etc.

iChat (10.6 and 10.7) and its reincarnation, Messages (10.8+), are the native instant messaging applications of OS X. These applications can be used with a variety of instant messaging programs and protocols. Messages replaced iChat in 10.8 (Beta was available earlier) and introduced integration with iMessage (iOS) and FaceTime.

## macOS Messages: Stored Chats ~/Library/Messages/Archive/YYYY-MM-DD/

<buddy's\_username> on YYYY-MM-DD at HH:MM:SS

Archive	2018-09-19	on 2018-09-05 at 14.08.12
Attachments	2018-09-20	on 2018-09-21 at 11.10.02
chat.db	2018-09-21	on 2018-08-12 at 23.37.15
chat.db-shm	2018-09-22	on 2018-09-21 at 09.32.18
chat.db-wal	2018-09-23	on 2018-09-04 at 13.05.26
CloudKitMetaData	2018-09-24	on 2018-09-21 at 13.41.49
StickerCache	2018-09-25	Chat with [redacted] on 2018-08-11 at 17.59.27
	2018-09-26	Chat with [redacted] on 2018-09-07 at 19.44.51
	2018-09-27	Chat with [redacted] on 2018-05-12 at 19.08.07
	2018-09-28	on 2018-09-21 at 13.21.11
	2018-09-29	on 2018-09-21 at 18.14.45

Messages will also store chat conversations if configured to do so. These files are stored in the ~/Library/Messages/Archive/ directory, also organized by date. These files use the same iChat binary property list format saved by date and time.

## Chat Database and Attachments: sms.db/chat.db

### Database

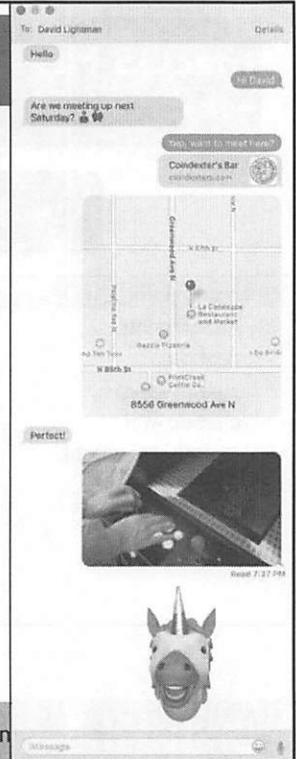
- iOS: sms.db
- macOS: chat.db

### Attachments

- Nested Folder Structure
- Paths in Database

#### Name

▶	Folder	Archive
▶	Folder	Attachments
	File	chat.db
	File	chat.db-shm
	File	chat.db-wal
▶	Folder	CloudKitMetaData
	File	prewarm.db
	File	prewarm.db-shm
	File	prewarm.db-wal
▶	Folder	StickerCache



macOS: ~/Library/Messages/chat.db  
Physical: iOS 6+: /private/var/mobile/Library/SMS/sms.db  
Backup/FS: iOS 6+: /mobile/Library/SMS/sms.db

The Messages application on iOS can be used for both iMessage and SMS-based messaging. The SMS directory will contain a SQLite database, sms.db, and an /Attachments directory. On macOS the database is in the User's Messages directory and named chat.db.

The sms.db/chat.db database file contains the message and related metadata, while the /Attachments directory contains media attachments.

## Chat Database (sms.db/chat.db): Message Metadata

```

1 SELECT
2 CASE
3   WHEN LENGTH(message.date)=18 THEN DATETIME(message.date/1000000000+978307200,'unixepoch','localtime')
4   WHEN LENGTH(message.date)=9 THEN DATETIME(message.date + 978307200,'unixepoch','localtime')
5   ELSE 'N/A'
6 END AS 'MESSAGE DATE',
7 CASE
8   WHEN LENGTH(MESSAGE.DATE_DELIVERED)=18 THEN DATETIME(MESSAGE.DATE_DELIVERED/1000000000+978307200,'unixepoch','localtime')
9   WHEN LENGTH(MESSAGE.DATE_DELIVERED)=9 THEN DATETIME(MESSAGE.DATE_DELIVERED + 978307200,'unixepoch','localtime')
10  ELSE 'N/A'
11 END AS 'DATE DELIVERED',
12 CASE
13   WHEN LENGTH(MESSAGE.DATE_READ)=18 THEN DATETIME(MESSAGE.DATE_READ/1000000000+978307200,'unixepoch','localtime')
14   WHEN LENGTH(MESSAGE.DATE_READ)=9 THEN DATETIME(MESSAGE.DATE_READ + 978307200,'unixepoch','localtime')
15  ELSE 'N/A'
16 END AS 'DATE READ',
17 MESSAGE.TEXT, HANDLE_ID AS 'CONTACT ID', MESSAGE.SERVICE, MESSAGE.ACCOUNT, MESSAGE.IS_DELIVERED, MESSAGE.IS_FROM_ME, ATTACHMENT.FILENAME, ATTACHMENT.MIME_TYPE, ATTACHMENT.TRANSFER_NAME, ATTACHMENT.TOTAL_BYTES
18 FROM MESSAGE
19 LEFT OUTER JOIN MESSAGE_ATTACHMENT_JOIN ON MESSAGE.ROWID = MESSAGE_ATTACHMENT_JOIN.MESSAGE_ID
20 LEFT OUTER JOIN ATTACHMENT ON MESSAGE_ATTACHMENT_JOIN.ATTACHMENT_ID = ATTACHMENT.ROWID
21 LEFT OUTER JOIN HANDLE ON MESSAGE.HANDLE_ID = HANDLE.ROWID

```

MESSAGE DATE	DATE DELIVERED	DATE READ	text	CONTACT ID	service	account	is_delivered	is_from_me	filename
2019-05-05 18:55:55	2019-05-05 19:22:39	2019-05-05 19:22:39	Hello	d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	0	NULL
2019-05-05 19:26:05	2019-05-05 19:26:08	2019-05-05 19:26:08	Hi David	d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	1	NULL
2019-05-05 19:27:33	N/A	N/A	Are we meeting up next Saturday? 轟轟	d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	0	NULL
2019-05-05 19:33:25	2019-05-05 19:35:00	2019-05-05 19:35:00	Yep, want to meet here?	d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	1	NULL
2019-05-05 19:33:29	2019-05-05 19:35:00	2019-05-05 19:35:00	http://www.colndexters.com	d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	1	~/Library/Messages/Attachments/76/06/BC1...
2019-05-05 19:34:47	2019-05-05 19:35:00	2019-05-05 19:35:00		d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	1	~/Library/Messages/Attachments/e1/01/D6BD...
2019-05-05 19:35:19	2019-05-05 19:35:39	2019-05-05 19:35:39	Perfect!	d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	0	NULL
2019-05-05 19:37:16	2019-05-05 19:37:19	2019-05-05 19:37:19		d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	1	~/Library/Messages/Attachments/71/01/28FF...
2019-05-05 19:41:10	N/A	N/A		d.1ghtm4n@gmail.com	iMessage	e.oempa@csh.rit.edu	1	1	~/Library/Messages/Attachments/99/09/32B...

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 91

macOS: ~/Library/Messages/chat.db

Physical: iOS 6+: /private/var/mobile/Library/SMS/sms.db

Backup/FS: iOS 6+: /mobile/Library/SMS/sms.db

A query on the chat.db SQLite database file shown below will extract the messages, timestamps, text, contact numbers/emails, originating account, attachment data, whether the message was delivered, and who originated it. The timestamps with “N/A” above originally contained a “0”. These timestamps are in Mac Epoch. A specific CASE statement was created to deal with variable lengths of timestamps in the timestamp columns.

Reference:

<https://smarterforensics.com/2017/09/time-is-not-on-our-side-when-it-comes-to-messages-in-ios-11/>



```

1 SELECT
2 CASE
3 WHEN LENGTH(message.date)=18 THEN DATETIME(message.date/1000000000+978307200,'unixepoch','localtime')
4 WHEN LENGTH(message.date)=9 THEN DATETIME(message.date +978307200,'unixepoch','localtime')
5 else "N/A"
6 END AS "MESSAGE DATE",
7 CASE
8 WHEN LENGTH(MESSAGE.DATE_DELIVERED)=18 THEN DATETIME(MESSAGE.DATE_DELIVERED/1000000000+978307200,'unixepoch','localtime')
9 WHEN LENGTH(MESSAGE.DATE_DELIVERED)=9 THEN DATETIME(MESSAGE.DATE_DELIVERED +978307200,'unixepoch','localtime')
10 else "N/A"
11 END AS "DATE DELIVERED",
12 CASE
13 WHEN LENGTH(MESSAGE.DATE_READ)=18 THEN DATETIME(MESSAGE.DATE_READ/1000000000+978307200,'unixepoch','localtime')
14 WHEN LENGTH(MESSAGE.DATE_READ)=9 THEN DATETIME(MESSAGE.DATE_READ +978307200,'unixepoch','localtime')
15 else "N/A"
16 END AS "DATE READ",
17 MESSAGE.TEXT,HANDLE.ID AS "CONTACT ID", MESSAGE.SERVICE, MESSAGE.ACCOUNT, MESSAGE.IS_DELIVERED, MESSAGE.IS_FROM_ME, ATTACHMENT.FILENAME, ATTACHMENT.MIME_TYPE, ATTACHMENT.TRANSFER_NAME, ATTACHMENT.TOTAL_BYTES
18 FROM MESSAGE
19 LEFT OUTER JOIN MESSAGE_ATTACHMENT_JOIN ON MESSAGE.ROWID = MESSAGE_ATTACHMENT_JOIN.MESSAGE_ID
20 LEFT OUTER JOIN ATTACHMENT ON MESSAGE_ATTACHMENT_JOIN.ATTACHMENT_ID = ATTACHMENT.ROWID
21 LEFT OUTER JOIN HANDLE ON MESSAGE.HANDLE_ID = HANDLE.ROWID

```

	MESSAGE DATE	DATE DELIVERED	DATE READ	text	CONTACT ID	service	account	is_delivered	is_from_me	filename
1	2019-05-05 18:55:55	2019-05-05 19:22:39	2019-05-05 19:22:39	Hello	d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	0	NULL
2	2019-05-05 19:26:05	2019-05-05 19:26:08	2019-05-05 19:26:08	Hi David	d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	1	NULL
3	2019-05-05 19:27:33	N/A	N/A	Are we meeting up next Saturday? 🍷🍷	d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	0	NULL
4	2019-05-05 19:33:25	2019-05-05 19:35:00	2019-05-05 19:35:00	Yep, want to meet here?	d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	1	NULL
5	2019-05-05 19:33:29	2019-05-05 19:35:00	2019-05-05 19:35:00	<a href="http://www.coindexters.com">http://www.coindexters.com</a>	d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	1	~/Library/Messages/Attachments/76/06/BC1...
6	2019-05-05 19:34:47	2019-05-05 19:35:00	2019-05-05 19:35:00		d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	1	~/Library/Messages/Attachments/e1/01/D6BD...
7	2019-05-05 19:35:19	2019-05-05 19:35:39	2019-05-05 19:35:39	Perfect!	d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	0	NULL
8	2019-05-05 19:37:16	2019-05-05 19:37:19	2019-05-05 19:37:19		d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	1	~/Library/Messages/Attachments/71/01/2BFF...
9	2019-05-05 19:41:10	N/A	N/A		d.l1ghtm4n@gmail.com	iMessage	e.oompa@csh.rit.edu	1	1	~/Library/Messages/Attachments/99/09/32B...

## Chat Database (sms.db/chat.db): Emoji

Are we meeting up next Saturday? 🧑🏻‍🦱 🧑🏻‍🦱

0000	41	72	65	20	77	65	20	6d	65	65	74	69	6e	67	20	75	Are we meeting u
0010	70	20	6e	65	78	74	20	53	61	74	75	72	64	61	79	3f	p next Saturday?
0020	20	f0	9f	95	b9	f0	9f	91	be								ð 1ð ¾

						U+1F47E	\xF0\x9F\x91\xBE	alien monster
---	---	---	---	---	---	---------	------------------	---------------

**SANS** | **DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 93

macOS: ~/Library/Messages/chat.db

Physical: iOS 6+: /private/var/mobile/Library/SMS/sms.db

Backup/FS: iOS 6+: /mobile/Library/SMS/sms.db

Messages may contain emojis. These messages can be reviewed from the binary view to see the Unicode that each emoji uses. This is handy if your SQLite viewer does not support Unicode.

The example above shows the message “Are we meeting up next Saturday?” with two emojis following. The highlighted emoji hex can be looked up in an emoji table like the one shown to determine which one it is.

References:

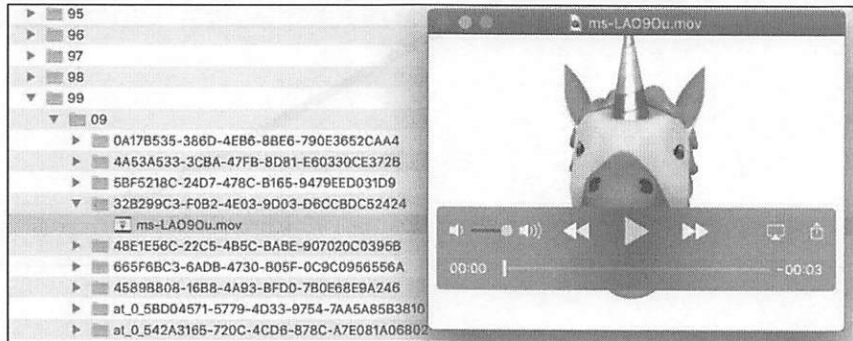
<https://apps.timwhitlock.info/emoji/tables/unicode>

<http://www.unicode.org/emoji/charts/full-emoji-list.html>

## Chat Database (sms.db/chat.db): Chat Attachments

- Photos, Videos, Maps, Safari Links, Locations, GIFs, Animoji, etc.

filename	mime_type	transfer_name	total_bytes
~/Library/Messages/Attachments/76/06/BC1C3FDF-70F5-4B11-89E0-46566E58F25D/88DB275B-D8FF-4ACB-9331-6D7101B97A90.pluginPayloadAttachment	NULL	88DB275B-DBFF-4ACB-9331-6D7101B97...	30534
~/Library/Messages/Attachments/e1/D1/D6BD16C8-4757-4854-9088-9137C3199B12/8556 Greenwood Ave N.loc.vcf	text/x-vlocation	8556 Greenwood Ave N.loc.vcf	800
NULL	NULL	NULL	NULL
~/Library/Messages/Attachments/71/01/2BFF9457-0ED1-41A6-9FCB-DC64DD7F5A9A/ms-Oysl5s.gif	image/gif	rms-Oysl5s.gif	1007854
~/Library/Messages/Attachments/99/09/32B299C3-F0B2-4E03-9D03-D6CCBDC52424/ms-LAO9Ou.mov	video/quicktime	ms-LAO9Ou.mov	522190



macOS: ~/Library/Messages/chat.db

Physical: iOS 6+: /private/var/mobile/Library/SMS/sms.db

Backup/FS: iOS 6+: /mobile/Library/SMS/sms.db

The “filename” column will hold a path to the attachment on the file system. An attachment does not have to be a picture or video. The “mime\_type” column can suggest that other types of attachments are available, to include this list:

- "image/jpeg"
- "video/3gpp"
- "text/x-vlocation"
- "text/vcard"
- "image/png"
- "image/tiff"
- "video/quicktime"
- "audio/x-m4a"
- "image/gif"
- "audio/amr"

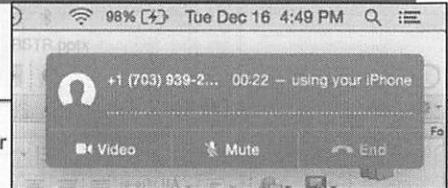
filename	mime_type	transfer_name	total_bytes
~/Library/Messages/Attachments/76/06/BC1C3FDF-70F5-4611-89E0-46566E58F25D/88DB275B-DBFF-4ACB-9331-6D7101B97A90.pluginPayloadAttachment	NULL	88DB275B-DBFF-4ACB-9331-6D7101B97...	30534
~/Library/Messages/Attachments/e1/01/D6BD16C8-4757-4854-9088-9137C3199B12/8556 Greenwood Ave N.loc.vcf	text/x-vlocation	8556 Greenwood Ave N.loc.vcf	600
NULL	NULL	NULL	NULL
~/Library/Messages/Attachments/71/01/2BFF9457-0ED1-41A6-9FCB-DC64DD7F5A9A/ms-0ysL5s.gif	image/gif	ms-0ysL5s.gif	1007854
~/Library/Messages/Attachments/99/09/32B299C3-F0B2-4E03-9D03-D6CCBDC52424/ms-LAO9Ou.mov	video/quicktime	ms-LAO9Ou.mov	522190



## Newer Call Records: CallHistory.storedata [macOS 10.10+, iOS 8+]

Call Records on macOS/iOS may be synced with macOS 10.12!

```
1 select datetime(zdate+978307200,'unixepoch','localtime') as zdate,
2 zaddress, zanswered, zcalltype, zoriginated, zduration, zlocation, zservice_provider
3 from ZCALLRECORD order by zdate desc
```



	zdate	ZADDRESS	ZANSWERED	ZCALLTYPE	ZORIGINATED	ZDURATION	ZLOCATION	ZSERVICE_PROVIDER
1	2017-02-09 14:33:14	+1 [REDACTED]	0	1	0	0.0	Virginia Beach, VA	com.apple.Telephony
2	2017-02-09 14:32:52	+1 [REDACTED]	0	1	0	0.0	Virginia Beach, VA	com.apple.Telephony
3	2017-02-07 18:09:10	+1 [REDACTED]	1	1	0	21.0	United States	com.apple.Telephony
4	2017-02-07 17:43:21	+1 [REDACTED]	1	1	0	42.0	United States	com.apple.Telephony
5	2017-02-07 17:42:17	+1 [REDACTED]	0	1	1	8.0	United States	com.apple.Telephony
6	2017-01-31 09:52:17	75 [REDACTED]	0	1	0	0.0	Portsmouth, VA	com.apple.Telephony
7	2017-01-30 20:09:42	+1 [REDACTED]	1	1	0	12.0	Arlington, VA	com.apple.Telephony
8	2017-01-26 19:36:00	+1 [REDACTED]	1	1	0	30.0	Arlington, VA	com.apple.Telephony

SANS DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 96

macOS: ~/Library/Application Support/CallHistoryDB/CallHistory.storedata  
 iOS Physical: /private/var/mobile/Library/CallHistoryDB/CallHistory.storedata  
 iOS Backup: /mobile/Library/CallHistoryDB/CallHistory.storedata

The SQLite database CallHistory.storedata located in the directory contains the calls made. The screenshot in the upper right-hand corner contains the notification that the user will see when taking a call using their OS X system. This call data is then written to the database.

Each record contains the following data:

- ZDATE: Mac Epoch Timestamp
- ZADDRESS: Contact phone number or email address (FaceTime)
- ZANSWERED: 0 = No, 1 = Yes
- ZCALLTYPE: (If duration is 0, call was missed/canceled)
  - 1: Call
  - 8: FaceTime
  - 16: FaceTime Voice Call
- ZORIGINATED: 1 = Outgoing with this user, 0 = Incoming
- ZDURATION: Time in seconds of the call
- ZLOCATION: Location (from Mobile Records)
- ZSERVICE\_PROVIDER: Application (com.apple.Telephony—Regular Call, may also see com.apple.FaceTime)

10.10–10.11: Only calls that were taken with the laptop or desktop system were recorded in this database. As of 10.12, the entire iPhone call history may be synced and can be accessed within the FaceTime application on macOS.



```
select
datetime(zdate+978307200,'unixepoch','localtime') as zdate,
zaddress,
zanswered,
zcalltype,
zoriginated,
zduration,
zlocation,
zservice_provider
from ZCALLRECORD
order by zdate desc
```

```

1 select datetime(zdate+978307200,'unixepoch','localtime') as zdate,
2 address, zanswered, zcalltype, zoriginated, zduration, zlocation, zservice_provider
3 from ZCALLRECORD order by zdate desc

```

	zdate	ZADDRESS	ZANSWERED	ZCALLTYPE	ZORIGINATED	ZDURATION	ZLOCATION	ZSERVICE_PROVIDER
1	2017-02-09 14:33:14	+1 [REDACTED]	0	1	0	0.0	Virginia Beach, VA	com.apple.Telephony
2	2017-02-09 14:32:52	+1 [REDACTED]	0	1	0	0.0	Virginia Beach, VA	com.apple.Telephony
3	2017-02-07 18:09:10	+1 [REDACTED]	1	1	0	21.0	United States	com.apple.Telephony
4	2017-02-07 17:43:21	+1 [REDACTED]	1	1	0	42.0	United States	com.apple.Telephony
5	2017-02-07 17:42:17	+1 [REDACTED]	0	1	1	8.0	United States	com.apple.Telephony
6	2017-01-31 09:52:17	75 [REDACTED]	0	1	0	0.0	Portsmouth, VA	com.apple.Telephony
7	2017-01-30 20:09:42	+1 [REDACTED]	1	1	0	12.0	Arlington, VA	com.apple.Telephony
8	2017-01-26 19:36:00	+1 [REDACTED]	1	1	0	30.0	Arlington, VA	com.apple.Telephony

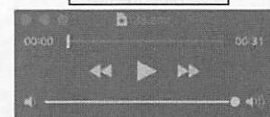
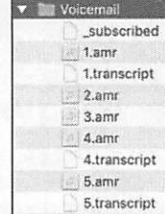
## Voicemail: voicemail.db, AMR, and \*.transcript Files

```

1 select
2 ROWID, datetime(date, 'unixepoch', 'localtime') as "Timestamp",
3 sender, duration, trashed_date, flags
4 from voicemail
5 where ROWID=15 or ROWID > 31

```

ROWID	Timestamp	sender	duration	trashed_date	flags
1	2015-10-29 12:06:33	703...	16	467851098	11
2	2016-03-04 11:52:44	703...	27	0	67
3	2016-06-07 14:25:58	703	31	0	67
4	2016-06-07 15:17:06	703	28	0	67
5	2016-08-13 17:16:49	703...	74	0	3
6	2016-08-23 17:17:45	703...	28	0	3
7	2016-08-24 07:56:55	703...	30	0	2



Item 0	String	NSArray
Item 1	String	NSObject
Item 185	String	From federal database please call immediately after this Heidi are recording at our headquarters number that is 360-639-8384 I repeat 360-639-8384 internal revenue service agent is waiting for your call back
Item 186	Dictionary	{2 items}
\$classname	String	VMVoicemailTranscript
\$classes	Array	{2 items}
Item 0	String	VMVoicemailTranscript
Item 1	String	NSObject
\$archiver	String	NSKeyedArchiver

Physical: iOS 6+: /private/var/mobile/Library/Voicemail/  
 Backup/FS: iOS 6+: /mobile/Library/Voicemail/

If the iOS device has the visual voicemail functionality—not all cellular carriers implement this—the device will download the voicemail audio files to the phone. The Voicemail directory will contain a voicemail.db SQLite database file and downloaded AMR files.

The database query performed extracts the voicemail ID, timestamp, sender info, duration, if and when it was trashed, and flag information. If the voicemail is set to be deleted but is still on the device, it will show a timestamp of that date in Mac Epoch. The flags determine if the voicemail is deleted, downloaded, or unheard.

- 11: Trashed (may still be downloaded)
- 3/67: Downloaded Voicemail
- 2: Unheard

Each voicemail that is downloaded can be listened to using most audio players that support the Adaptive Multi-rate Audio (AMR) audio format. The voicemail filename will use the “ROWID” (voicemail ID) as its filename.

If the voicemail has an accompanying \*.transcript file, you’ll find an NSKeyedArchiver plist file with the transcript information.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.

---

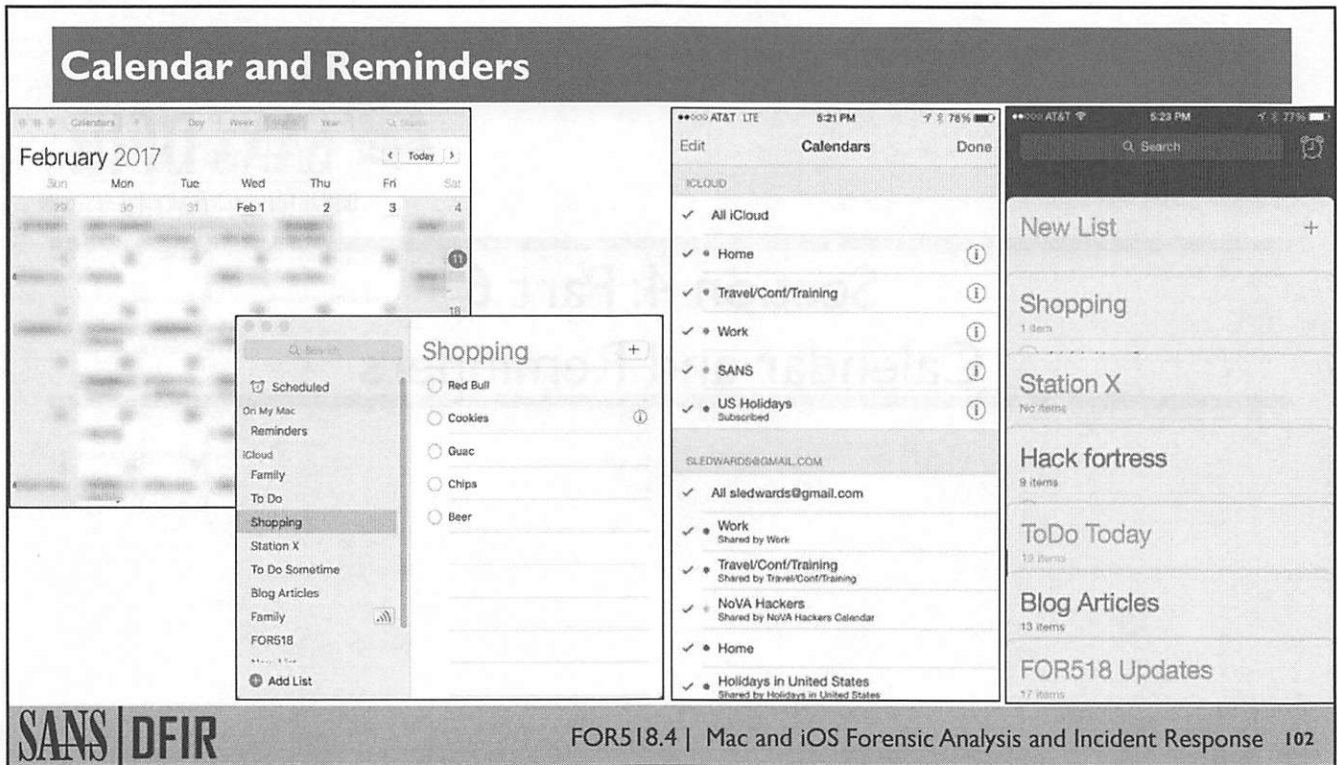
## Section 4: Part 6

# Calendar and Reminders

---

This page intentionally left blank.





The native calendar application is iCal or Calendar. The name changed to Calendar in Mountain Lion (10.8). The Reminders application was added later.

The Calendar and Reminders applications use the same database to store their contents.

These items can be synced from a variety of accounts, including iCloud and Google, as shown above. It can include both personal calendars and shared calendars.

The Reminders application stores lists created by the user that may have deadlines and other information associated with each reminder item.

## macOS: iCal / Calendar—Calendar Directories

```
Elwoods-Mac:Calendars elwoodblues$ pwd
/Users/elwoodblues/Library/Calendars
Elwoods-Mac:Calendars elwoodblues$ ls -l
total 424
drwxr-xr-x  4 elwoodblues  staff    136 Sep 23 11:28 788EEBAA-B298-47D5-AABB-2E7D00D2DECB.calendar
drwxr-xr-x  4 elwoodblues  staff    136 Sep 23 11:28 93B25A30-19AF-46BB-8416-AE184B886E00.calendar
drwxr-xr-x  3 elwoodblues  staff    102 Oct  4 13:03 Attachments
-rw-r--r--@ 1 elwoodblues  staff  217088 Oct  4 13:04 Calendar Cache
drwxr-xr-x  2 elwoodblues  staff     68 Oct  4 13:04 Calendar Sync Changes
```

```
Elwoods-Mac:Calendars elwoodblues$ pwd
/Users/elwoodblues/Library/Calendars
Elwoods-Mac:Calendars elwoodblues$ ls -laR 788EEBAA-B298-47D5-AABB-2E7D00D2DECB.calendar/
total 8
drwxr-xr-x  4 elwoodblues  staff    136 Sep 23 11:28 .
drwxr-xr-x  7 elwoodblues  staff    238 Oct  4 13:04 ..
drwxr-xr-x  3 elwoodblues  staff    102 Oct  4 13:03 Events
-rw-r--r--  1 elwoodblues  staff    565 Oct  4 13:03 Info.plist

788EEBAA-B298-47D5-AABB-2E7D00D2DECB.calendar//Events:
total 8
drwxr-xr-x  3 elwoodblues  staff    102 Oct  4 13:03 .
drwxr-xr-x  4 elwoodblues  staff    136 Sep 23 11:28 ..
-rw-r--r--  1 elwoodblues  staff    421 Oct  4 13:03 C5E7B415-079F-4BDB-81B9-1ECC8B483590.ics
```

macOS: ~/Library/Calendars/  
iOS Physical: /private/var/mobile/Library/Calendar/  
iOS Backup: /mobile/Library/Calendar/

Each calendar is saved as a separate directory, which is named after the calendar GUID and ends with a `.calendar` or `.caldav` extension. Each calendar directory contains an `Events` directory and an `Info.plist`.

The `Info.plist` contains information about the calendar, such as calendar name, GUI preferences, and other configurable items.

The `Events` directory contains the calendar `.ics` files. These files contain the calendar entries. Each `.ics` file contains information for a single calendar event.

## macOS: iCal / Calendar—Info.plist

```
Elwoods-Mac:788EEBAA-B298-47D5-AA8B-2E7D00D2DECB.calendar elwoodblues$ cat Info.plist
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd"
">
<plist version="1.0">
<dict>
  <key>AlarmsDisabled</key>
  <false/>
  <key>Checked</key>
  <integer>1</integer>
  <key>Color</key>
  <string>#44A703FF</string>
  <key>Editable</key>
  <true/>
  <key>Enabled</key>
  <true/>
  <key>Key</key>
  <string>788EEBAA-B298-47D5-AA8B-2E7D00D2DECB</string>
  <key>Order</key>
  <integer>2</integer>
  <key>Title</key>
  <string>Work</string>
  <key>Type</key>
  <string>Local</string>
</dict>
</plist>
```

In the example `Info.plist` above, the calendar is named “Work”. This calendar has a color associated with it (a lovely green color if you look up HTML color codes).

Other configurable items include alarm preferences, if the calendar can be edited, and the type of calendar. This one happens to be a local calendar, rather than a CalDAV calendar.

## macOS: iCal / Calendar Info.plist—CalDAV

Editable

Notes (User Account)

Time Zone

Title

Type (CalDAV)

```
<plist version="1.0">
<dict>
  <key>AlarmsDisabled</key>
  <false/>
  <key>Availability</key>
  <true/>
  <key>CTag</key>
  <string>63485046441</string>
  <key>CalendarPath</key>
  <string>/calendar/dav/sledwards%40gmail.com/events/</string>
  <key>CanBePublished</key>
  <false/>
  <key>Checked</key>
  <integer>1</integer>
  <key>Color</key>
  <string>#D06B64FF</string>
  <key>Delegate</key>
  <false/>
  <key>Editable</key>
  <true/>
  <key>Enabled</key>
  <true/>
  <key>EventContainer</key>
  <true/>
  <key>Key</key>
  <string>DDC1673C-7700-4CB0-9FF3-644EE27C4279</string>
  <key>Notes</key>
  <string>sledwards@gmail.com</string>
  <key>Order</key>
  <integer>1073741865</integer>
  <key>OwnerPrincipalPath</key>
  <string>/calendar/dav/sledwards%40gmail.com/user/</string>
  <key>Permission</key>
  <integer>4</integer>
  <key>Renameable</key>
  <true/>
  <key>ShareDefaultAlarmSettings</key>
  <true/>
  <key>TaskContainer</key>
  <false/>
  <key>TimeZone</key>
  <string>America/New_York</string>
  <key>Title</key>
  <string>Home</string>
  <key>Type</key>
  <string>CalDAV</string>
</dict>
</plist>
```

This slide shows an example of a CalDAV calendar `Info.plist`. CalDAV is a standard that allows multiple applications, such as iCal or Google Calendar, to save the same information and keep it synced between clients.

The CalDAV `Info.plist` file may contain more information than a normal calendar `Info.plist` file.

- Account Information, such as Google Calendars or iCloud accounts
- Time Zone Information
- Last Sync Date
- Sync Settings

```

<plist version="1.0">
<dict>
  <key>AlarmsDisabled</key>
  <false/>
  <key>Availability</key>
  <true/>
  <key>CTag</key>
  <string>63485046441</string>
  <key>CalendarPath</key>
  <string>/calendar/dav/sledwards%40gmail.com/events/</string>
  <key>CanBePublished</key>
  <false/>
  <key>Checked</key>
  <integer>1</integer>
  <key>Color</key>
  <string>#D06B64FF</string>
  <key>Delegate</key>
  <false/>
  <key>Editable</key>
  <true/>
  <key>Enabled</key>
  <true/>
  <key>EventContainer</key>
  <true/>
  <key>Key</key>
  <string>DDC1673C-7700-4CB0-9FF3-644EE27C4279</string>
  <key>Notes</key>
  <string>sledwards@gmail.com</string>
  <key>Order</key>
  <integer>1073741865</integer>
  <key>OwnerPrincipalPath</key>
  <string>/calendar/dav/sledwards%40gmail.com/user/</string>
  <key>Permission</key>
  <integer>4</integer>
  <key>Renameable</key>
  <true/>
  <key>ShareDefaultAlarmSettings</key>
  <true/>
  <key>TaskContainer</key>
  <false/>
  <key>TimeZone</key>
  <string>America/New_York</string>
  <key>Title</key>
  <string>Home</string>
  <key>Type</key>
  <string>CalDAV</string>
</dict>
</plist>

```



## iCal / Calendar: Events – \*.ics Files

```
Elwoods-Mac:Events elwoodblues$ cat C5E7B415-079F-4BDB-81B9-1ECC8B483590.ics
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Apple Inc.//iCal 5.0//EN
CALSCALE:GREGORIAN
BEGIN:VEVENT
CREATED:20121004T180254Z
UID:C5E7B415-079F-4BDB-81B9-1ECC8B483590
DTEND;TZID=America/Chicago:20121004T160000
TRANSP:OPAQUE
X-APPLE-DONTSCHEDULE:TRUE
SUMMARY:Meeting with Jake
DTSTART;TZID=America/Chicago:20121004T150000
DTSTAMP:20121004T180325Z
X-APPLE-EWS-BUSYSTATUS:BUSY
SEQUENCE:2
END:VEVENT
END:VCALENDAR
```

Each `.ics` file contained within a calendar is a calendar event. Shown above, it may contain:

- Create Date
- Event Start Time
- Event End Time
- Unique ID
- Time Zone Information
- Event Summary

Elwoods-Mac:Events elwoodblues\$ cat C5E7B415-079F-4BDB-81B9-1ECC8B483590.ics  
BEGIN:VCALENDAR  
VERSION:2.0  
PRODID:-//Apple Inc.//iCal 5.0//EN  
CALSCALE:GREGORIAN  
BEGIN:VEVENT  
CREATED:20121004T180254Z  
UID:C5E7B415-079F-4BDB-81B9-1ECC8B483590  
DTEND;TZID=America/Chicago:20121004T160000  
TRANSP:OPAQUE  
X-APPLE-DONTSCHEDULE:TRUE  
SUMMARY:Meeting with Jake  
DTSTART;TZID=America/Chicago:20121004T150000  
DTSTAMP:20121004T180325Z  
X-APPLE-EWS-BUSYSTATUS:BUSY  
SEQUENCE:2  
END:VEVENT  
END:VCALENDAR

## iCal / Calendar: CalDAV Events – \*.ics Files

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Apple Inc.//Mac OS X 10.8//EN
CALSCALE:GREGORIAN
BEGIN:VEVENT
DTEND;VALUE=DATE:20101202
TRANSP:TRANSPARENT
UID:va0av53o19f24vtam3l9f2s7po@google.com
DTSTAMP:20101102T005302Z
LOCATION:
DESCRIPTION:
STATUS:CONFIRMED
X-APPLE-SCHEDULETAG:
X-APPLE-SERVERFILENAME:va0av53o19f24vtam3l9f2s7po%40google.com.ics
SEQUENCE:0
X-APPLE-NEWS-BUSYSSTATUS:FREE
SUMMARY:Shmocon Tickets
LAST-MODIFIED:20101102T005302Z
DTSTART;VALUE=DATE:20101201
CREATED:20101020T200408Z
X-APPLE-ETAG:"63424342302"
BEGIN:VALARM
X-WR-ALARMUID:C0E39F79-2073-4050-B553-4674EF592ADA
UID:C0E39F79-2073-4050-B553-4674EF592ADA
TRIGGER;VALUE=DATE-TIME:20101130T215000Z
DESCRIPTION:This is an event reminder
ACTION:DISPLAY
END:VALARM
BEGIN:VALARM
X-WR-ALARMUID:BE34309D-161E-4200-8418-537F3779CB14
UID:BE34309D-161E-4200-8418-537F3779CB14
TRIGGER:-PT15H
X-APPLE-DEFAULT-ALARM:TRUE
ATTACH;VALUE=URI:Basso
ACTION:AUDIO
END:VALARM
END:VEVENT
END:VCALENDAR
```

CalDAV calendar events may have additional information, depending on the client the event was created with. The example above shows a Google Calendar event.

BEGIN:VCALENDAR  
VERSION:2.0  
PRODID:-//Apple Inc.//Mac OS X 10.8//EN  
CALSCALE:GREGORIAN  
BEGIN:VEVENT  
DTEND;VALUE=DATE:20101202  
TRANSP:TRANSPARENT  
UID:vaoav53oi9f24vtam3lgf2s7po@google.com  
DTSTAMP:20101102T005302Z  
LOCATION:  
DESCRIPTION:  
STATUS:CONFIRMED  
X-APPLE-SCHEDULETAG:  
X-APPLE-SERVERFILENAME:vaoav53oi9f24vtam3lgf2s7po%40google.com.ics  
SEQUENCE:0  
X-APPLE-EWS-BUSYSTATUS:FREE  
SUMMARY:Shmoocoon Tickets  
LAST-MODIFIED:20101102T005302Z  
DTSTART;VALUE=DATE:20101201  
CREATED:20101020T200408Z  
X-APPLE-ETAG:"63424342382"  
BEGIN:VALARM  
X-WR-ALARMUID:C0E39F79-2073-4050-B553-4674EF592ADA  
UID:C0E39F79-2073-4050-B553-4674EF592ADA  
TRIGGER;VALUE=DATE-TIME:20101130T215000Z  
DESCRIPTION:This is an event reminder  
ACTION:DISPLAY  
END:VALARM  
BEGIN:VALARM  
X-WR-ALARMUID:BE34309D-161E-420B-8418-537F3779CB14  
UID:BE34309D-161E-420B-8418-537F3779CB14  
TRIGGER:-PT15H  
X-APPLE-DEFAULT-ALARM:TRUE  
ATTACH;VALUE=URI:Basso  
ACTION:AUDIO  
END:VALARM  
END:VEVENT  
END:VCALENDAR

## iCal / Calendar: Events with Location

```
BEGIN:VCALENDAR
VERSION:2.0
PRODID:-//Apple Inc.//Mac OS X 10.10.1//EN
CALSCALE:GREGORIAN
BEGIN:VEVENT
TRANSP:OPAQUE
DTEND;TZID=America/New_York:20141213T200000
X-APPLE-STRUCTURED-LOCATION;VALUE=URI;X-ADDRESS=2911 District Ave\\nMerrifield VA 22031;X-APPLE-RADIUS=14161.30926963496;X-TITLE=Angelika Film Center & Cafe at Mosaic;geo:38.872212,-77.229711
UID:61BE4907-3EA5-456E-95C6-6552322AF6C2
DTSTAMP:20141213T172708Z
LOCATION:Angelika Film Center & Cafe at Mosaic\\n2911 District Ave\\nMerrifield VA 22031
DESCRIPTION:
STATUS:CONFIRMED
X-APPLE-SCHEDULETAG:"87756a35d7205eb3"
X-APPLE-SERVERFILENAME:61BE4907-3EA5-456E-95C6-6552322AF6C2.ics
SEQUENCE:0
X-APPLE-EWS-BUSYSTATUS:BUSY
SUMMARY:Movies
DTSTART;TZID=America/New_York:20141213T170000
LAST-MODIFIED:20141213T172708Z
CREATED:20141213T172708Z
X-APPLE-ETAG:"63554174828"
END:VEVENT
END:VCALENDAR
```

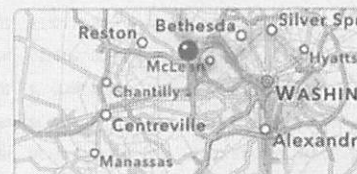
### Movies

Angelika Film Center & Cafe at Mosaic  
2911 District Ave  
Merrifield VA 22031

Dec 13, 2014 5 PM to 8 PM

Add Invitees

Add Notes or URL



Calendar events may also be created with location information.

In the screenshot above, a calendar entry was created at a particular theater in Merrifield, VA. The user is able to search and select a particular location, which is embedded in the ICS file, as shown above.



## macOS Calendar Cache (iOS Calendar.sqlitedb): Calendar Events

```

1 select
2 datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",
3 datetime(zdatestamp+978307200,'unixepoch','localtime') as "Last Modified",
4 datetime(zstartdate+978307200,'unixepoch','localtime') as "Event Start",
5 datetime(zenddate+978307200,'unixepoch','localtime') as "Event End",
6 ztimezone, zcalendaritem.ztitle as "Event", znode.ztitle as "Calendar/List",
7 zcompleteddate, zstatus, from zcalendaritem
8 left join znode on zcalendaritem.zcalendar = znode.z_pk

```

	Created	Last Modified	Event Start	Event End	ZTIMEZONE	Event	Calendar/Reminder List
36	2017-02-08 21:42:39	2017-02-08 21:42:39	2016-03-17 19:00:00	2016-03-17 20:30:00	America/New...	Reservation: Charley's Ste...	Found in Apps
37	2017-02-08 21:42:39	2017-02-08 21:42:39	2016-03-13 16:00:00	2016-03-13 17:30:00	GMT-0700	Reservation: Skillet Diner - ...	Found in Apps
38	2017-02-08 21:42:36	2017-02-08 21:42:36	2016-05-31 19:00:00	2016-05-31 20:30:00	GMT-0400	Reservation: Texas Jacks ...	Found in Apps
39	2017-02-08 21:42:34	2017-02-08 21:42:34	2016-08-03 21:00:00	2016-08-04 00:00:00	GMT-0700	Ticket: Volatility Black Hat ...	Found in Apps
40	2017-02-08 21:42:34	2017-02-08 21:42:34	2016-07-04 16:00:00	2016-07-04 17:30:00	America/New...	Ticket: 4th of July BBQ	Found in Apps
41	2017-02-08 21:42:33	2017-02-08 21:42:33	2016-08-13 12:00:00	2016-08-13 13:30:00	GMT-0400	Ticket: MacDMV August M...	Found in Apps
42	2017-02-08 19:27:20	2017-02-08 19:27:20	1904-06-17 19:00:00	1904-06-18 19:00:00	NULL		Birthdays
43	2017-02-08 18:56:56	2017-02-08 16:35:22	2017-02-24 20:00:00	2017-02-24 22:00:00	GMT		Facebook Events
44	2017-02-08 15:38:32	2017-02-08 15:38:32	1984-09-06 20:00:00	1984-09-07 20:00:00	NULL		Birthdays
45	2017-02-07 19:45:29	2017-02-07 19:45:29	1904-09-26 19:00:00	1904-09-27 19:00:00	NULL		Birthdays
46	2013-08-08 12:00:55	2017-02-07 18:01:28	2016-01-12 19:00:00	2016-01-13 19:00:00	NULL		Home
47	2017-02-07 13:00:10	2017-02-07 13:00:10	2017-03-24 10:30:00	2017-03-24 11:30:00	US/Eastern		Home

SANS DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 112

The Calendar Cache (or Calendar.sqlitedb on iOS) SQLite database contains information for the Calendars. These databases have slightly different table names and have also changed column names over time, but contain generally the same data.

The ZCALENDARITEM (previously ZICSELEMENT) table contains calendar event information. Each tuple contains data similar to that found in the .ics file.

These calendar items or reminders contain a variety of data, partially shown here for brevity.

Items included are:

- Summary/item name
- Start and end dates
- If the calendar item is an all-day event
- What calendar it belongs to (review the Calendar table)
- When the event or reminder was last modified/created
- When the reminder was completed
- ... so much more!

```

select
datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",
datetime(zdatestamp+978307200,'unixepoch','localtime') as "Last Modified",
datetime(zstartdate+978307200,'unixepoch','localtime') as "Event Start",
datetime(zenddate+978307200,'unixepoch','localtime') as "Event End",
ztimezone,
zcalendaritem.ztitle as "Event", znode.ztitle as "Calendar/List" ,
zcompleteddate, zstatus
from zcalendaritem
left join znode on zcalendaritem.zcalendar = znode.z_pk

```

```

1 select
2 datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",
3 datetime(zdatestamp+978307200,'unixepoch','localtime') as "Last Modified",
4 datetime(zstartdate+978307200,'unixepoch','localtime') as "Event Start",
5 datetime(zenddate+978307200,'unixepoch','localtime') as "Event End",
6 timezone, zcalendaritem.ztitle as "Event", znode.ztitle as "Calendar/List",
7 zcompleteddate, zstatus, from zcalendaritem
8 left join znode on zcalendaritem.zcalendar = znode.z_pk

```

	Created	Last Modified	Event Start	Event End	ZTIMEZONE	Event	Calendar/Reminder List
36	2017-02-08 21:42:39	2017-02-08 21:42:39	2016-03-17 19:00:00	2016-03-17 20:30:00	America/New_...	Reservation: Charley's Ste...	Found in Apps
37	2017-02-08 21:42:39	2017-02-08 21:42:39	2016-03-13 16:00:00	2016-03-13 17:30:00	GMT-0700	Reservation: Skillet Diner - ...	Found in Apps
38	2017-02-08 21:42:36	2017-02-08 21:42:36	2016-05-31 19:00:00	2016-05-31 20:30:00	GMT-0400	Reservation: Texas Jacks ...	Found in Apps
39	2017-02-08 21:42:34	2017-02-08 21:42:34	2016-08-03 21:00:00	2016-08-04 00:00:00	GMT-0700	Ticket: Volatility Black Hat ...	Found in Apps
40	2017-02-08 21:42:34	2017-02-08 21:42:34	2016-07-04 16:00:00	2016-07-04 17:30:00	America/New_...	Ticket: 4th of July BBQ	Found in Apps
41	2017-02-08 21:42:33	2017-02-08 21:42:33	2016-08-13 12:00:00	2016-08-13 13:30:00	GMT-0400	Ticket: MacDMV August M...	Found in Apps
42	2017-02-08 19:27:20	2017-02-08 19:27:20	1904-06-17 19:00:00	1904-06-18 19:00:00	NULL		Birthdays
43	2017-02-06 18:56:56	2017-02-08 16:35:22	2017-02-24 20:00:00	2017-02-24 22:00:00	GMT		Facebook Events
44	2017-02-08 15:38:32	2017-02-08 15:38:32	1984-09-06 20:00:00	1984-09-07 20:00:00	NULL		Birthdays
45	2017-02-07 19:45:29	2017-02-07 19:45:29	1904-09-26 19:00:00	1904-09-27 19:00:00	NULL		Birthdays
46	2013-08-08 12:00:55	2017-02-07 18:01:28	2016-01-12 19:00:00	2016-01-13 19:00:00	NULL		Home
47	2017-02-07 13:00:10	2017-02-07 13:00:10	2017-03-24 10:30:00	2017-03-24 11:30:00	US/Eastern		Home

## macOS Calendar Cache (iOS Calendar.sqlitedb): Reminder Items

### Shopping +

- Red Bull
- Chips
- Coke

```

1 select
2 datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",
3 datetime(zdatestamp+978307200,'unixepoch','localtime') as "Last Modified",
4 datetime(zstartdate+978307200,'unixepoch','localtime') as "Event Start",
5 datetime(zenddate+978307200,'unixepoch','localtime') as "Event End",
6 ztimezone, zcalendaritem.ztitle as "Event", znode.ztitle as "Calendar/List",
7 zcompleteddate, zstatus, from zcalendaritem
8 left join znode on zcalendaritem.zcalendar = znode.z_pk
    
```

	Created	Last Modified	Event Start	Event End	ZTIMEZONE	Event	Calendar/Reminder List	ZCOMPLETEDDATE	ZSTATUS
1	2017-02-11 20:10:43	2017-02-11 21:27:44	NULL	NULL	NULL	Guac	Shopping	508559264	COMPLETED
2	2017-02-11 20:06:27	2017-02-11 21:27:41	NULL	NULL	NULL	Cookies	Shopping	508559261	COMPLETED
3	2017-02-11 20:10:47	2017-02-11 21:08:10	NULL	NULL	NULL	Coke	Shopping	NULL	NULL
4	2017-02-11 20:10:45	2017-02-11 20:10:46	NULL	NULL	NULL	Chips	Shopping	NULL	NULL
5	2017-02-11 20:06:25	2017-02-11 20:06:26	NULL	NULL	NULL	Red Bull	Shopping	NULL	NULL

**SANS** | **DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 114

macOS: ~/Library/Calendars/  
 iOS Physical: /private/var/mobile/Library/Calendar/  
 iOS Backup: /mobile/Library/Calendar/

The Reminders application also uses the (Calendar Cache/Calendar.sqlitedb) file.

Each reminder may be associated with a specific list; in this example, Shopping. This list's names can be correlated with data found in the ZNODE table of this database.

A creation timestamp is recorded when the list item is put into the application. After the item is specified as completed by the user, a completed date is recorded and the status set to "COMPLETED". While these items no longer show up in the current list, an investigator may still see them in the database.

```

select
datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",
datetime(zdatestamp+978307200,'unixepoch','localtime') as "Last Modified",
datetime(zstartdate+978307200,'unixepoch','localtime') as "Event Start",
datetime(zenddate+978307200,'unixepoch','localtime') as "Event End",
ztimezone,
zcalendaritem.ztitle as "Event", znode.ztitle as "Calendar/List",
zcompleteddate, zstatus from zcalendaritem
left join znode on zcalendaritem.zcalendar = znode.z_pk
    
```

```

1  select
2  datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",
3  datetime(zdatestamp+978307200,'unixepoch','localtime') as "Last Modified",
4  datetime(zstartdate+978307200,'unixepoch','localtime') as "Event Start",
5  datetime(zenddate+978307200,'unixepoch','localtime') as "Event End",
6  ztimezone, zcalendaritem.ztitle as "Event", znode.ztitle as "Calendar/List" ,
7  zcompleteddate, zstatus, from zcalendaritem
8  left join znode on zcalendaritem.zcalendar = znode.z_pk

```

	Created	Last Modified	Event Start	Event End	ZTIMEZONE	Event	Calendar/Reminder List	ZCOMPLETEDDATE	ZSTATUS
1	2017-02-11 20:10:43	2017-02-11 21:27:44	NULL	NULL	NULL	Guac	Shopping	508559264	COMPLETED
2	2017-02-11 20:06:27	2017-02-11 21:27:41	NULL	NULL	NULL	Cookies	Shopping	508559261	COMPLETED
3	2017-02-11 20:10:47	2017-02-11 21:08:10	NULL	NULL	NULL	Coke	Shopping	NULL	NULL
4	2017-02-11 20:10:45	2017-02-11 20:10:46	NULL	NULL	NULL	Chips	Shopping	NULL	NULL
5	2017-02-11 20:06:25	2017-02-11 20:06:26	NULL	NULL	NULL	Red Bull	Shopping	NULL	NULL

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



---

## Section 4: Part 7

### Contacts

---

This page intentionally left blank.

## Address Book / Contacts



macOS: ~/Library/Application Support/AddressBook/  
iOS Physical: /private/var/mobile/Library/AddressBook/  
iOS Backup: /mobile/Library/AddressBook/

The Address Book, or Contacts application, holds the user's contact information. This database gets populated by the user but also “automagically” from other sources if the user associates their email and social media accounts with the operating system.

The left-hand side of the macOS screenshot shows contacts from iCloud, local (On My Mac), Facebook, and Google.

Just like iCal/Calendar, the Address Book was renamed to Contacts in Mountain Lion (10.8).

## macOS Address Book / Contacts: Different Sources ~/Library/Application Support/AddressBook/Sources/

```
[bit:Sources oompa$ pwd
/Users/oompa/Library/Application Support/AddressBook/Sources
[bit:Sources oompa$ ls -l
total 0
drwxr-xr-x  10 oompa  staff  340 Feb  6 18:57 BF7B48B9-0A83-4190-9F64-BB8779BA6128
drwxr-xr-x  11 oompa  staff  374 Feb 12 07:39 ED106899-0EA3-4DAB-86AB-83DDF17B159C
drwxr-xr-x  10 oompa  staff  340 Feb  6 18:57 EE54CA64-23CD-4CFB-9D8D-EB96FC79E8E2
[bit:Sources oompa$ ls -l ED106899-0EA3-4DAB-86AB-83DDF17B159C/
total 5920
-rw-r--r--@  1 oompa  staff   962560 Feb  6 18:57 AddressBook-v22.abcd.db
-rw-r--r--@  1 oompa  staff   32768 Feb  7 18:05 AddressBook-v22.abcd.db-shm
-rw-r--r--@  1 oompa  staff  2031192 Feb 12 07:39 AddressBook-v22.abcd.db-wal
drwx-----  43 oompa  staff   1462 Feb  6 07:48 Images
drwx----- 376 oompa  staff  12784 Feb 12 07:39 Metadata
-rw-r--r--@  1 oompa  staff    181 Feb 12 07:39 OfflineDeletedItems.plist
-rwxr-xr-x@  1 oompa  staff     0 Feb  6 07:48 OfflineDeletedItems.plist.lockfile
-rwxr-xr-x@  1 oompa  staff     0 Feb  6 07:48 Sync.lockfile
-rwxr-xr-x@  1 oompa  staff     0 Feb  6 07:48 SyncOperations.plist.lockfile
```

macOS: ~/Library/Application Support/AddressBook/  
iOS Physical: /private/var//mobile/Library/AddressBook/  
iOS Backup: /mobile/Library/AddressBook/

Each Address Book database under Sources will have contact information separated out. One could be the contacts for Facebook, Google, iCloud—or another source.

These sources can be determined by matching the GUID up in the Accounts database.

## macOS Address Book / Contacts: Metadata Directories

\*.abcdp: Per Person

\*.abcdg: Per Group

\*.abcds: One Subscription Record (hidden file)

```
bit:Metadata oompa$ pwd
/Users/oompa/Library/Application Support/AddressBook/Sources/ED106899-0EA3-4DAB-86AB-83DDF17B159C/Metadata
bit:Metadata oompa$ ls -l
total 3016
-rw-----  1 oompa  staff   1374 Feb  6 07:48 004F7096-2B53-43B0-9854-B543ABB8AA8E:ABPerson.abcdp
-rw-----@ 1 oompa  staff   1895 Feb  7 08:39 006A5B28-152C-4EAB-805C-40EB05B5C861:ABPerson.abcdp
-rw-----  1 oompa  staff   1366 Feb  6 18:58 0128C2A0-07F1-4E41-949A-B62C7900260B:ABPerson.abcdp
-rw-----  1 oompa  staff   1391 Feb  6 07:48 019FB7C2-1FA1-4001-A015-D53C7456C3E6:ABPerson.abcdp
```

The Metadata directory contains a file for each person, group, or subscription. Each file is named accordingly: “p” for person, “g” for group, and “s” for subscription. Each file is a binary property list containing the information for that particular Address Book entry.

Much of the same information found in the Metadata directory may be found in the SQLite database AddressBook-v22.abcd.db, in the ~/Library/Application Support/AddressBook/ directory.

Newer versions of OS X have a Sources directory that contains the GUID of the specific Address Book source. You will find the Metadata directory here.

## macOS Address Book / Contacts: Person Record Metadata Directory – \*.abcdp

- Timestamps (Creation, Modification)
- Contact Name
- Contact Address Book GUID
- Email Addresses
- Physical Address
- Phone Numbers
- Social Media Accounts
- Websites
- Birthday
- Etc.

Key	Type	Value
▼ Root	Dictionary	(10 items)
UID	String	5F7FB365-A760-4486-B60E-03318C40376A:ABPerson
Creation	Date	Sep 23, 2012 12:29:43 PM
First	String	Elwood
Modification	Date	Oct 5, 2012 9:12:54 PM
▼ Phone	Dictionary	(4 items)
▶ identifiers	Array	(1 item)
▼ values	Array	(1 item)
Item 0	String	515 5551212
primary	String	408E586B-3CCE-44E7-AC4D-5C69D6C33E1D
▶ labels	Array	(1 item)
ABPersonFlags	Number	0
▼ Address	Dictionary	(4 items)
▶ identifiers	Array	(1 item)
▼ values	Array	(1 item)
▼ Item 0	Dictionary	(6 items)
Street	String	1060 W Addison
ZIP	String	60613
CountryCode	String	us
City	String	Chicago
State	String	IL
Country	String	United States
primary	String	6DB3EBED-561B-4204-AA8E-58F9A2AD8AF4
▶ labels	Array	(1 item)
▶ ABPropertyTypes	Dictionary	(37 items)
Last	String	Blues
▶ Email	Dictionary	(4 items)

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 121

To view the raw Address Book records with Xcode, you will need to rename them as `.plist`; otherwise, they will show up in a view similar to what is found in the Address Book/Contacts application.

Each `Person` record has its own GUID in the `UID` key. This GUID may be used in the `Group` record if the `Person` record is part of a group.

The `Creation` key contains the date the record was created, while the `Modification` date contains the date when the contact was last modified.

Each `Person` record contains the phone, address, email, or other information the user has supplied. If the contact has multiple entries under the “values” key, such as multiple phone numbers, the “primary” key will determine which of these is listed as the primary contact number.



Key	Type	Value
▼ Root	Dictionary	(10 items)
UID	String	5F7FB365-A760-4486-B60E-03318C40376A:ABPerson
Creation	Date	Sep 23, 2012 12:29:43 PM
First	String	Elwood
Modification	Date	Oct 5, 2012 9:12:54 PM
▼ Phone	Dictionary	(4 items)
▶ Identifiers	Array	(1 item)
▼ values	Array	(1 item)
Item 0	String	515 5551212
primary	String	408E5B6B-3CCE-44E7-AC4D-5C69D6C33E1D
▶ labels	Array	(1 item)
ABPersonFlags	Number	0
▼ Address	Dictionary	(4 items)
▶ Identifiers	Array	(1 item)
▼ values	Array	(1 item)
▼ Item 0	Dictionary	(6 items)
Street	String	1060 W Addison
ZIP	String	60613
CountryCode	String	us
City	String	Chicago
State	String	IL
Country	String	United States
primary	String	6DB3EBED-561B-4204-AA8E-58F9A2AD8AF4
▶ labels	Array	(1 item)
▶ ABPropertyTypes	Dictionary	(37 items)
Last	String	Blues
▶ Email	Dictionary	(4 items)

## macOS Address Book / Contacts: Group Record Metadata Directory – \*.abcdg

Key	Type	Value
▼ Root	Dictionary	(8 items)
UID	String	C45523F6-048D-44FB-8574-0042F21D4633:ABGroup
ABGroupClassKey	String	ABGroup
▶ ABAddressDistributionList	Dictionary	(0 items)
ABPersonFlags	Number	0
▶ ABEmailDistributionList	Dictionary	(0 items)
▼ ABMembers	Array	(2 items)
Item 0	String	5F7FB365-A760-4486-B60E-03318C40376A:ABPerson
Item 1	String	98BD4905-88A0-4201-9E75-C61E6FB92E77:ABPerson
▶ ABPhoneDistributionList	Dictionary	(0 items)
GroupName	String	Family

The Group record contains all the GUIDs of Person records that are part of that group in the ABMembers key. Each group also has its own GUID that can be referenced by various applications.

The keys ABEmailDistributionList, ABPhoneDistributionList, and ABAddressDistributionList may be populated if the records are part of an email, phone, or mailing distribution list.

Key	Type	Value
▼ Root	Dictionary	(8 items)
UID	String	C45523F6-04BD-44FB-8574-0042F21D4633:ABGroup
ABGroupClassKey	String	ABGroup
▶ ABAddressDistributionList	Dictionary	(0 items)
ABPersonFlags	Number	0
▶ ABEmailDistributionList	Dictionary	(0 items)
▼ ABMembers	Array	(2 items)
Item 0	String	5F7FB365-A760-4486-B60E-03318C40376A:ABPerson
Item 1	String	9BBD4905-88A0-4201-9E75-C61E6FB92E77:ABPerson
▶ ABPhoneDistributionList	Dictionary	(0 items)
GroupName	String	Family

## macOS Address Book / Contacts: Images Directory

	0E7E1EBD-B76D-4507-B48C-094EF1E517A3
	0E7E1EBD-B76D-4507-B48C-094EF1E517A3.jpeg
	3C611C99-2B24-43D6-9916-1A9861A1BFAD
	3C611C99-2B24-43D6-9916-1A9861A1BFAD.jpeg
	4F1A0E82-4122-4EF5-BA9D-6F0702578AC7
	4F1A0E82-4122-4EF5-BA9D-6F0702578AC7.jpeg
	06FF5CA8-70AC-4493-B1B5-AC8B393FD625
	06FF5CA8-70AC-4493-B1B5-AC8B393FD625.jpeg
	8C39839D-A8EB-401E-8740-4058CC79F20E
	8DA73B19-D645-47EA-8D46-A3A3A95D33D7
	8DA73B19-D645-47EA-8D46-A3A3A95D33D7.jpeg

The Images directory contains contact profile pictures named by that contact's GUID.

Some contacts may have more than one icon picture associated with them. This may be due to the pictures being resized or cropped to show well in the Address Book or Contacts application.

Newer versions of OS X have a Sources directory that contains the GUID of the specific Address Book source. You will find the Images directory here.

## Contacts DB:AddressBook-v22.abcd.db (iOS—AddressBook.sqlite)

```
1 select zabcdrecord.zuniqueid,  
2 datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",  
3 datetime(zmodificationdate+978307200,'unixepoch','localtime') as "Modified",  
4 zfirstname, zlastname, zabcdemailaddress.zaddress, zabcdphonenum.zfullnumber  
5 from zabcdrecord  
6 left join zabcdemailaddress on zabcdrecord.z_pk = zabcdemailaddress.zowner  
7 left join zabcdphonenum on zabcdrecord.z_pk = zabcdphonenum.zowner
```

	ZUNIQUEID	Created	Modified	ZFIRSTNAME	ZLASTNAME	ZADDRESS	ZFULLNUMBER
525	56F477C7-94...	2017-02-06 07:48:42	2017-02-12 07:38:46				
526	56F477C7-94...	2017-02-06 07:48:42	2017-02-12 07:38:46				
527	C980A3C3-56...	2017-02-06 07:48:42	2017-02-06 07:48:42				
528	C980A3C3-56...	2017-02-06 07:48:42	2017-02-06 07:48:42				
529	56E5ECFE-94...	2017-02-06 07:48:42	2017-02-06 07:48:42				
530	9F9F2A44-BF...	2017-02-06 07:48:42	2017-02-06 18:57:30				
531	5F9CBD5C-78...	2017-02-06 07:48:42	2017-02-06 07:48:42				
532	21191F1E-1D23...	2017-02-06 07:48:42	2017-02-06 18:57:30				
533	0E02DA2C-4F...	2017-02-06 07:48:42	2017-02-06 07:48:42				
534	D7D90F6A-38...	2017-02-06 07:48:42	2017-02-06 18:57:30				

macOS: ~/Library/Application Support/AddressBook/AddressBook-v22.abcd.db

iOS Physical: /private/var/mobile/Library/AddressBook/AddressBook.sqlite

iOS Backup: /mobile/Library/AddressBook/AddressBook.sqlite

The Contacts application stores its data in SQLite databases, AddressBook.sqlite on iOS and AddressBook-v22.abcd.db on macOS. The example shown above is from macOS 10.12. The table and column names from iOS and macOS will be slightly different but will contain the same type of information—a similar SQL query can be constructed to gather the same information or additional information as needed for the analyst.

The databases contain contacts information including names, emails, addresses, phone numbers, social media accounts, etc.

The ABCDRECORD table (ABPerson on iOS) stores the contact name, organization, job title, etc. This table also stores the contact creation and modification dates. There are many more tables that hold additional information. In the example, the email addresses and phone numbers were extracted from other tables.

```
select  
zabcdrecord.zuniqueid,  
datetime(zcreationdate+978307200,'unixepoch','localtime') as "Created",  
datetime(zmodificationdate+978307200,'unixepoch','localtime') as "Modified",  
zfirstname,  
zlastname,  
zabcdemailaddress.zaddress,  
zabcdphonenum.zfullnumber  
from zabcdrecord  
left join zabcdemailaddress on zabcdrecord.z_pk = zabcdemailaddress.zowner  
left join zabcdphonenum on zabcdrecord.z_pk = zabcdphonenum.zowner
```



---

## Lab 4.2

# Applications: Part I

---

This page intentionally left blank.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.

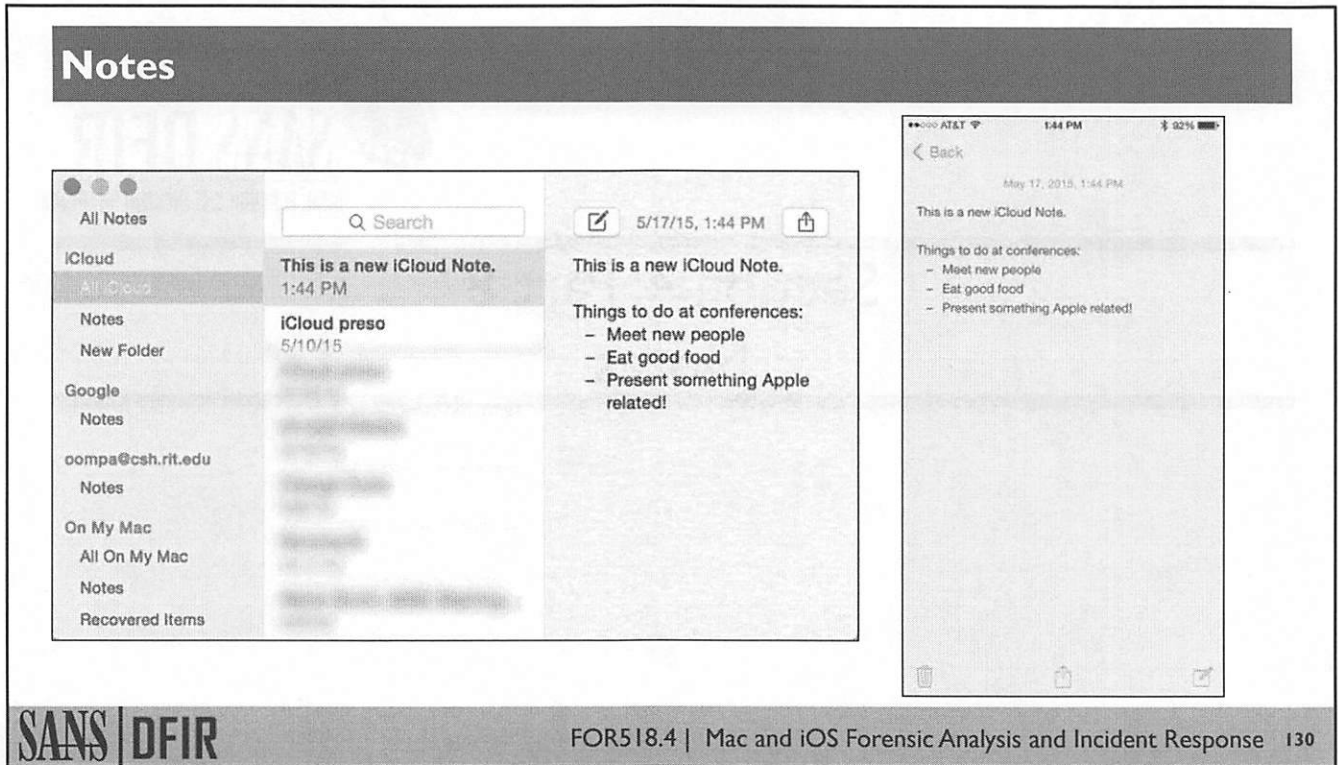
---

## Section 4: Part 8

### Notes

---

This page intentionally left blank.



The Notes application allows a user to create notes of various types on macOS, iCloud.com, or on iOS devices. If the user selects one of the syncing services, these notes will be pushed to all devices associated with these accounts.

The iCloud service will sync all iCloud notes to all devices the user has specified to use iCloud services. The user may also choose to create local device notes that do not get synced (“On My Mac”).

The Notes application and data files have been rapidly changing ever since it was introduced in 10.8.

## Notes: Files and Directories

### iOS Physical

iCloud [iOS 9+]: /var/mobile/Containers/Shared/AppGroup/<GUID>/NoteStore.sqlite  
Legacy iCloud/Local Notes: /private/var/mobile/Library/Notes/notes.sqlite  
Cached Snapshots [iOS 9+]:  
/private/var/mobile/Containers/Data/Application/<GUID>/Library/Caches/Snapshots/com.apple.mobilenotes/

### iOS Backup/File System

/mobile/Applications/ [com.apple.notes/ or Notes/]  
iCloud [iOS 9+]: NoteStore.sqlite  
Legacy iCloud/Local Notes: notes.sqlite

### OS X

Legacy iCloud/Local Notes: ~/Library/Containers/com.apple.Notes/Data/Library/Notes/NotesV#.storedata [V4 10.10, V5/V6 10.11+, V7 10.13]  
iCloud (newer): ~/Library/Group Containers/group.com.apple.notes/NoteStore.sqlite

The files and paths displayed above show where and how the notes are stored. The Notes application has changed from the `notes.sqlite` database to the `NoteStore.sqlite` database. The introduction of notes that contain images, movies, and other attachments has changed the notes database quite a bit in recent versions.

#### Attachments:

iOS Physical/File System: /mobile/Applications/com.apple.notes/Media/<GUID>/  
OS X: ~/Library/Group Containers/group.com.apple.notes/Media/  
OS X: ~/Library/Containers/com.apple.Notes/Data/Library/CoreData/Attachments/



## Differences in Notes Application Versions

### Legacy iCloud and Local Notes

#### Database Filename

- iOS: notes.sqlite
- OS X: NotesV#.storedata
  - (V4 10.10, V5/V6 10.11, V7 10.13)

#### Note Storage

- HTML

#### Data Types

- Rich Text
- Photos (in V6 or iOS 8+)

### Newer iCloud Notes [iOS 9+]

#### Database Filename

- NoteStore.sqlite

#### Note Storage

- Gzip Archive or Encrypted

#### Data Types

- Text, Photo/Video, Sketch, Map Links, Safari Links, Checklist

### Secure Notes (iOS 9.3)

The Legacy iCloud/Local Notes were stored in a simpler database named `notes.sqlite`. On macOS, these databases had a version number associated with them. The newer database file is `NoteStore.sqlite`. Each version of the database may have slight differences in it. The basic note body storage was plaintext HTML, while the newer versions are using Gzip files as a BLOB in a database column.

The newer versions of Notes allow users to embed media such as photo, video, sketches, checklists, and links to Maps, Safari, etc.

Secure notes were introduced with iOS 9.3 and macOS 10.11—in this case, the Gzip data is encrypted. The cryptographic information can be found in various associated columns.

## Notes Database: notes.sqlite / NotesV#.storedata Notes and Content

- ZCONTENT: HTML-formatted content
- ZFILENAME: Look in /attachments directory for files

```
1 select
2 znote.z_pk,
3 datetime(znote.zcreationdate+978307200,'unixepoch','localtime') as "Creation Date",
4 datetime(znote.zmodificationdate+978307200,'unixepoch','localtime') as "Modification Date",
5 znote.zdeletedflag, znote.ztitle,zaccount.zname as "Account Name",znotebody.zcontent,znoteattachment.zfilename
6 from znote
7 left join zstore on znote.zstore == zstore.z_pk
8 left join zaccount on zstore.zaccount == zaccount.z_pk
9 left join znotebody on znote.zbody == znotebody.z_pk
10 left join znoteattachment on znote.z_pk == znoteattachment.znote
11 order by "Creation Date"
```

Z_PK	Creation Date	Modification Date	ZDELETEDFLAG	ZTITLE	Account Name	ZCONTENT	ZFILENAME	
41	828	2016-09-04 17:50:16	2016-09-04 17:50:36	0	Picture Note	sledwards@gmail.com	<html><head></head><body style="word-wrap: break-word; -webkit-nbsp-mode: space; -webkit-line-break: after-white-space;">Picture Note...	IMG_0793.JPG
42	831	2016-09-04 17:50:43	2016-09-04 17:52:16	0	password 1	sledwards@gmail.com	<html><head></head><body style="word-wrap: break-word; -webkit-nbsp-mode: space; -webkit-line-break: after-white-space;"><div> </...</div></body></html>	NULL
43	832	2016-09-04 19:59:09	2016-09-04 19:59:27	0	Secrets	sledwards@gmail.com	Secrets </div></div> Password is skjchBxhd23</div></div> </div></div>	NULL
44	833	2016-09-04 19:59:30	2016-09-04 20:00:23	0	Neat photo	sledwards@gmail.com	Neat photo </div></div></div></div><object type="application/x-apple-msg-attachment" data="cid:4F975426-267B-4EBA-964B-7622BFB5A5..."></object></div></div></div></div></div></div></div>	unknown.jpg
45	834	2016-09-04 20:10:52	2016-09-04 20:11:54	0	A few of my favorite things:	sledwards@gmail.com	A few of my favorite things </div></div></div></div> </div></div></div></div></div></div></div></div></div>	NULL

This SQLite query extracts each note item, its timestamps, whether it's marked deleted or not, note title, associated account, HTML content, and attachment filename from the older notes.sqlite/NotesV#.storedata files. The attachments can be found in the /attachments directory. Look for the matching filename.

The timestamps in this database are stored in Mac Epoch.

```
select
znote.z_pk,
datetime(znote.zcreationdate+978307200,'unixepoch','localtime') as "Creation
Date",
datetime(znote.zmodificationdate+978307200,'unixepoch','localtime') as
"Modification Date",
znote.zdeletedflag,
znote.zsummary,
znote.ztitle,
znote.zauthor,
znote.zguid,
zaccount.zname as "Account Name",
znotebody.zcontent,
znoteattachment.zcontentid,
znoteattachment.zfilename
from znote
left join zstore on znote.zstore == zstore.z_pk
left join zaccount on zstore.zaccount == zaccount.z_pk
left join znotebody on znote.zbody == znotebody.z_pk
left join znoteattachment on znote.z_pk == znoteattachment.znote
order by "Creation Date"
```

```

1 select
2 znote.z_pk,
3 datetime(znote.zcreationdate+978387200,'unixepoch','localtime') as "Creation Date",
4 datetime(znote.zmodificationdate+978387200,'unixepoch','localtime') as "Modification Date",
5 znote.zdeletedflag, znote.ztitle,zaccount.zname as "Account Name",znotebody.zcontent,znoteattachment.zfilename
6 from znote
7 left join zstore on znote.zstore == zstore.z_pk
8 left join zaccount on zstore.zaccount == zaccount.z_pk
9 left join znotebody on znote.zbody == znotebody.z_pk
10 left join znoteattachment on znote.z_pk == znoteattachment.znote
11 order by "Creation Date"

```

Z_PK	Creation Date	Modification Date	ZDELETEDFLAG	ZTITLE	Account Name	ZCONTENT	ZFILENAME
41	2016-09-04 17:50:16	2016-09-04 17:50:36	0	Picture Note	sledwards@gmail.com	<html><head></head><body style="word-wrap: break-word; -webkit- nbsp-mode: space; -webkit-line-break: after-white-space;">Picture Note...	IMG_0793.JPG
42	2016-09-04 17:50:43	2016-09-04 17:52:16	0	password 1	sledwards@gmail.com	<html><head></head><body style="word-wrap: break-word; -webkit- nbsp-mode: space; -webkit-line-break: after-white-space;"><div> </div>...	NULL
43	2016-09-04 19:59:09	2016-09-04 19:59:27	0	Secrets	sledwards@gmail.com	Secrets<div> </div><div>Password is skjchBxhd23</div><div> </div>	NULL
44	2016-09-04 19:59:30	2016-09-04 20:00:23	0	Neat photo	sledwards@gmail.com	Neat photo <div> </div><div><object type="application/x-apple- msg-attachment" data="cid:4F975426-267B-4EBA-964B-7622BF5A5...	unknown.jpg
45	2016-09-04 20:10:52	2016-09-04 20:11:54	0	A few of my favorite things:	sledwards@gmail.com	A few of my favorite things:<div> </div><div>Cats</div><div>Dogs</div><div>Cars</div>...	NULL

# Notes Database [iOS 9+]: NotesStore.sqlite Note Folders

- Folders:
- ZICCLOUDSYNCINGOBJECT.Z\_ENT == 12

Folder	Count
All iCloud	36
Notes	28
My Secrets	6
Research	2
Recently Deleted	3

```

1 select
2 ZICCLOUDSYNCINGOBJECT.Z_PK,
3 ZICCLOUDSYNCINGOBJECT.zaccountnameforaccountlistsorting as "Account Name",
4 ZICCLOUDSYNCINGOBJECT.zidentifier,
5 ZICCLOUDSYNCINGOBJECT.ztitle2 as "Folder Title",
6 ZICCLOUDSYNCINGOBJECT.zdateforlasttitlemodification,
7 datetime(ZICCLOUDSYNCINGOBJECT.zdateforlasttitlemodification+978307200,'unixepoch','localtime') as "Folder Title Last Modified",
8 ZICCLOUDSYNCINGOBJECT.zmarkedfordeletion
9 from ZICCLOUDSYNCINGOBJECT
10 where ZICCLOUDSYNCINGOBJECT.z_ent == 12 --Note Folders are Z_ENT = 12
    
```

Z_PK	Account Name	ZIDENTIFIER	Folder Title	ZDATEFORLASTTITLEMODIFICATION	Folder Title Last Modified	ZMARKEDFORDELETION
1 2	1_iCloud	DefaultFolder-CloudKit	Notes	NULL	NULL	0
2 3	1_iCloud	TrashFolder-CloudKit	Recently Deleted	NULL	NULL	0
3 66	1_iCloud	336F4A56-18D0-421A-AA9A-DAAA3C1C9D1C	Research	469407828.849662	2015-11-16 18:03:48	0
4 104	1_iCloud	EBE5D44E-310E-4018-8F41-7842C3676851	My Secrets	493489715.001946	2016-08-21 12:28:35	0

In the newer Notes SQLite database, this query extracts each note folder, associated account name, when the folder was last modified, and whether it's marked for deletion.

The “Z\_ENT” 12 is selected here to gather folder data. 12 is the key for “ICFolder” information. The screenshot to the right shows other data that may be extracted.

The Last Modified timestamp is stored as Mac Epoch.

```

select
ZICCLOUDSYNCINGOBJECT.Z_PK,
ZICCLOUDSYNCINGOBJECT.zaccountnameforaccountlistsorting
as "Account Name",
ZICCLOUDSYNCINGOBJECT.zidentifier,
ZICCLOUDSYNCINGOBJECT.ztitle2 as "Folder Title",
ZICCLOUDSYNCINGOBJECT.zdateforlasttitlemodification,datetime(ZICCLOUDSYNCINGOBJECT.zdateforlasttitlemodification+978307200,'unixepoch','localtime') as "Folder Title
Last Modified",
ZICCLOUDSYNCINGOBJECT.zmarkedfordeletionfrom
ZICCLOUDSYNCINGOBJECT
where ZICCLOUDSYNCINGOBJECT.z_ent == 12 are Z_ENT = 12
    
```

Z_ENT	Z_NAME
1	ICAuthor
2	ICCloudState
3	ICCloudSyncingObject
4	ICAttachment
5	ICAttachmentPreviewImage
6	ICDeviceMigrationState
7	ICLegacyTombstone
8	ICMedia
9	ICNote
10	ICNoteContainer
11	ICAccount
12	ICFolder
13	ICDDevice
14	ICGroup
15	ICLocation
16	ICAttachmentLocation
17	ICNoteChange
18	ICNoteData
19	ICPerson
20	ICSearchIndexTransaction
21	ICServerChangeToken
22	NextId

```

1  select
2  ZICLOUDSYNCINGOBJECT.Z_PK,
3  ZICLOUDSYNCINGOBJECT.zaccountnameforaccountlistsoring as "Account Name",
4  ZICLOUDSYNCINGOBJECT.zidentifier,
5  ZICLOUDSYNCINGOBJECT.ztitle2 as "Folder Title",
6  ZICLOUDSYNCINGOBJECT.zdateforlasttitlemodification,
7  datetime(ZICLOUDSYNCINGOBJECT.zdateforlasttitlemodification+978307200,'unixepoch','localtime') as "Folder Title Last Modified",
8  ZICLOUDSYNCINGOBJECT.zmarkedfordeletion
9  from ZICLOUDSYNCINGOBJECT
10 where ZICLOUDSYNCINGOBJECT.z_ent == 12 —Note Folders are Z_ENT = 12

```

Z_PK	Account Name	ZIDENTIFIER	Folder Title	ZDATEFORLASTTITLEMODIFICATION	Folder Title Last Modified	ZMARKEDFORDELETION
1 2	1_iCloud	DefaultFolder-CloudKit	Notes	NULL	NULL	0
2 3	1_iCloud	TrashFolder-CloudKit	Recently Deleted	NULL	NULL	0
3 66	1_iCloud	336F4A56-18D0-421A-AA9A-DAAA3C1C9D1C	Research	469407828.849662	2015-11-16 18:03:48	0
4 104	1_iCloud	EBE5D44E-310E-4018-8F41-7842C3676851	My Secrets	493489715.001946	2016-08-21 12:28:35	0



## Notes Database [iOS 9+]: NotesStore.sqlite Notes

- Notes under “My Secrets” Folder (#104)
- Secure Notes: Password-Protected Notes
- ZDATA = Note Contents



```

2 ZICCLOUDSYNCINGOBJECT.Z_PK,ZICCLOUDSYNCINGOBJECT.zidentifier,Z_12NOTES.Z_12FOLDERS as "Note Folder",
3 datetime(ZICCLOUDSYNCINGOBJECT.zcreationdate+978307200,'unixepoch','localtime') as "Create Time",
4 datetime(ZICCLOUDSYNCINGOBJECT.zmodificationdate1+978307200,'unixepoch','localtime') as "Modification Time",
5 datetime(ZICCLOUDSYNCINGOBJECT.zfoldersmodificationdate+978307200,'unixepoch','localtime') as "Folder Modification Time",
6 ZICCLOUDSYNCINGOBJECT.ztitle1 as "Note Title",ZICCLOUDSYNCINGOBJECT.zsnippet,ZICCLOUDSYNCINGOBJECT.zispasswordprotected as "Protected",
7 ZICCLOUDSYNCINGOBJECT.zpasswordhint,ZICCLOUDSYNCINGOBJECT.zmarkedfordeletion as "Deleted",ZICNOTEDATA.ZDATA
8 from ZICCLOUDSYNCINGOBJECT
9 left join ZICNOTEDATA on ZICCLOUDSYNCINGOBJECT.znotedata == ZICNOTEDATA.Z_pk
10 left join Z_12NOTES on ZICCLOUDSYNCINGOBJECT.z_pk == Z_12NOTES.Z_9NOTES
11 where ZICCLOUDSYNCINGOBJECT.z_ent == 9 and "Note Folder" == 104
12 order by "Create Time"

```

Z_PK	ZIDENTIFIER	Note Folder	Create Time	Modification Time	Folder Modification Time	Note Title	ZSNIPPET	Protected	ZPASSWORDHINT	Deleted	ZDATA
1	3E8C7FCC-A1F2-4641-8320-41F5EA81B0B7	104	2016-08-21 12:28:37	2016-08-21 12:29:33	2016-08-21 13:04:33	Passwords:	NULL	1	it's password	0	BLOB
2	5DD1257F-EE19-42CF-9A25-89CE7575D203	104	2016-08-21 12:29:39	2016-08-21 12:31:19	2016-08-21 13:04:17	Credits Cards:	NULL	1	it's password	0	◆◆◆...
3	B0F078A1-855B-43DE-B945-7F60758D66DF1	104	2016-08-21 12:32:41	2016-08-21 12:33:52		New Note		0		0	
4	23511203-24B4-47F4-8066-281864C5CE0B	104	2016-08-21 12:34:01	2016-08-21 12:34:20	NULL	New Note		0	NULL	0	BLOB
5	76B28680-E815-43D7-BFF2-A4821C02058C	104	2016-08-21 12:34:26	2016-08-21 18:39:19	NULL	Things to do!	The next great app	0	NULL	0	BLOB
6	76B0A953-0B2C-4FC7-8DD0-D0F7EA69841D	104	2016-08-21 12:36:42	2016-08-21 12:37:43	NULL	Kitty!		0	NULL	0	BLOB

**SANS** | **DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 137

This query extracts the Note data to include timestamps, title, snippets, and whether or not it is encrypted and/or marked for deletion. The highlighted row contains a note that is named “New Note” and is not encrypted. If it was, the Protected column would show a “1”. The note body is stored in a GZIP archive in the ZDATA column in the BLOB.

```

select
ZICCLOUDSYNCINGOBJECT.Z_PK,
ZICCLOUDSYNCINGOBJECT.zidentifier,
Z_12NOTES.Z_12FOLDERS as "Note Folder",
datetime(ZICCLOUDSYNCINGOBJECT.zcreationdate+978307200,'unixepoch','localtime')
as "Create Time",
datetime(ZICCLOUDSYNCINGOBJECT.zmodificationdate1+978307200,'unixepoch','localtime')
as "Modification Time",
datetime(ZICCLOUDSYNCINGOBJECT.zfoldersmodificationdate+978307200,'unixepoch','localtime')
as "Folder Modification Time",
ZICCLOUDSYNCINGOBJECT.ztitle1 as "Note Title",
ZICCLOUDSYNCINGOBJECT.zsnippet,
ZICCLOUDSYNCINGOBJECT.zispasswordprotected as "Protected",
ZICCLOUDSYNCINGOBJECT.zpasswordhint,
ZICCLOUDSYNCINGOBJECT.zmarkedfordeletion as "Deleted",
ZICNOTEDATA.ZDATA from ZICCLOUDSYNCINGOBJECT
left join ZICNOTEDATA on ZICCLOUDSYNCINGOBJECT.znotedata == ZICNOTEDATA.Z_pk
left join Z_12NOTES on ZICCLOUDSYNCINGOBJECT.z_pk == Z_12NOTES.Z_9NOTES
where ZICCLOUDSYNCINGOBJECT.z_ent == 9 and "Note Folder" == 104
order by "Create Time"

```



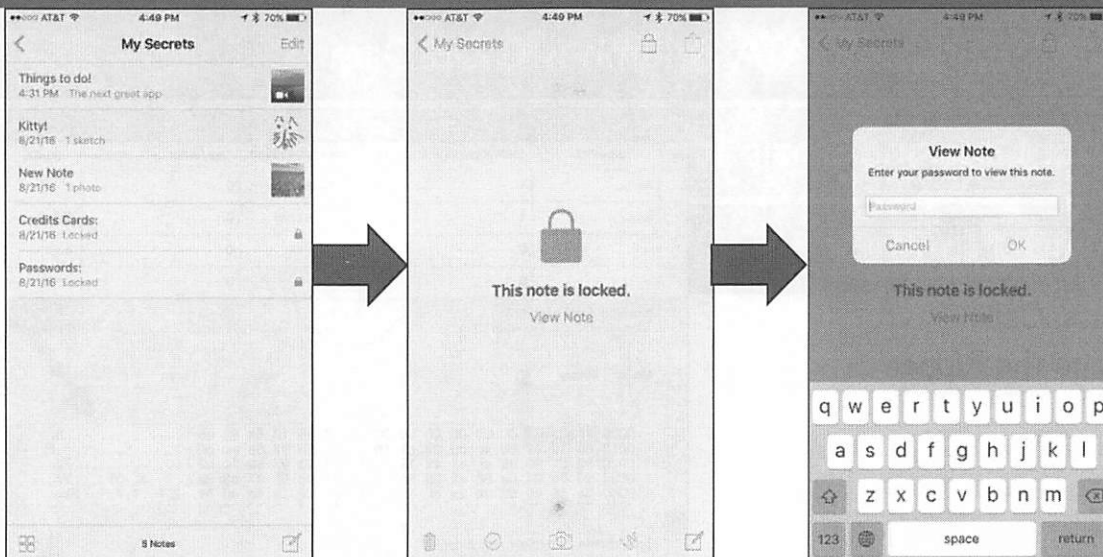
```

2 ZICLOUDSYNCINGOBJECT.Z_PK,ZICLOUDSYNCINGOBJECT.zidentifler,Z_12NOTES.Z_12FOLDERS as "Note Folder",
3 datetime(ZICLOUDSYNCINGOBJECT.zcreationdate+978307200,'unixepoch','localtime') as "Create Time",
4 datetime(ZICLOUDSYNCINGOBJECT.zmodificationdate+978307200,'unixepoch','localtime') as "Modification Time",
5 datetime(ZICLOUDSYNCINGOBJECT.zfoldersmodificationdate+978307200,'unixepoch','localtime') as "Folder Modification Time",
6 ZICLOUDSYNCINGOBJECT.ztitle as "Note Title",ZICLOUDSYNCINGOBJECT.zsnippet,ZICLOUDSYNCINGOBJECT.zispasswordprotected as "Protected",
7 ZICLOUDSYNCINGOBJECT.zpasswordhint,ZICLOUDSYNCINGOBJECT.zmarkedfordeletion as "Deleted",ZICNOTEDATA.ZDATA
8 from ZICLOUDSYNCINGOBJECT
9 left join ZICNOTEDATA on ZICLOUDSYNCINGOBJECT.znotedata == ZICNOTEDATA.Z_pk
10 left join Z_12NOTES on ZICLOUDSYNCINGOBJECT.z_pk == Z_12NOTES.Z_9NOTES
11 where ZICLOUDSYNCINGOBJECT.z_ent == 9 and "Note Folder" == 104
12 order by "Create Time"

```

Z_PK	ZIDENTIFIER	Note Folder	Create Time	Modification Time	Folder Modification Time	Note Title	ZSNIPPET	Protected	ZPASSWORDHINT	Deleted	ZDATA
1	136	3E8C7FCC-A1F2-4641-8320-41F5EA81B0B7	104	2016-08-21 12:28:37	2016-08-21 12:29:33	2016-08-21 13:04:33	Passwords: <i>NULL</i>	1	it's password	0	<i>BLOB</i>
2	135	5DD1257F-EE19-42CF-9A25-89CE7575D203	104	2016-08-21 12:29:39	2016-08-21 12:31:19	2016-08-21 13:04:17	Credits Cards: <i>NULL</i>	1	it's password	0	◆◆◆◆...
3	107	B0F078A1-855B-43DE-B945-7F80759D6DF1	104	2016-08-21 12:32:41	2016-08-21 12:33:52	<i>NULL</i>	New Note	0	<i>NULL</i>	0	
4	112	23511203-24B4-47F4-8066-281864C5CE0B	104	2016-08-21 12:34:01	2016-08-21 12:34:20	<i>NULL</i>	New Note	0	<i>NULL</i>	0	<i>BLOB</i>
5	119	76B2B660-E815-43D7-BFF2-A4921C02059C	104	2016-08-21 12:34:26	2016-08-21 18:39:19	<i>NULL</i>	Things to do! The next great app	0	<i>NULL</i>	0	<i>BLOB</i>
6	126	76B0A953-0B2C-4FC7-8DD0-D0F7EA69841D	104	2016-08-21 12:36:42	2016-08-21 12:37:43	<i>NULL</i>	Kitty!	0	<i>NULL</i>	0	<i>BLOB</i>

## Notes Database [iOS 9+]: NotesStore.sqlite Notes—Secure Notes



SANS DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 139

With iOS 9.3 and macOS 10.11, the secure notes feature was introduced. Secure notes may contain sensitive data, pictures, and/or other media. The example above shows the “My Secrets” note folder, which contains a couple of Secure Notes. These notes are marked by the small lock icon.

If a secure note is selected, the user will get a “This note is locked.” message and will have to input their Notes password to view the data.

Secure Notes is only supported for iCloud notes; it is not available for local device notes.

## Notes Database [iOS 9+]: NotesStore.sqlite Notes—Note Contents [1]

“BLOB” data in ZDATA column is GZip Data

Extract Data and Unzip It

If Secure Note:

- HashCat
- John the Ripper

The screenshot shows the SQLite Database Browser interface. A table with columns ZSNIPPET, ZISPASSWORDPROTECTED, ZPASSWORDHINT, ZMARKEDFORDELETION, and ZDATA is displayed. The ZDATA column contains BLOB data. A dialog box titled 'Edit Database Cell' is open, showing the BLOB data in a hex viewer. The hex viewer displays the following data:

Hex	ASCII
0000 1f 8b 08 00 00 00 00 00 03 e3 60 10 6a 62 e5	..... b.
0010 60 10 60 90 fa cd 22 e4 16 92 91 99 97 5e ac 50	.....^P
0020 92 af 90 92 af c8 c5 15 92 91 aa 90 97 5a 51 a2	.....ZC
0030 90 5e 94 9a 58 a2 90 58 50 c0 e5 9c 9f 57 58 9a	.....VX
0040 5a a4 50 02 94 2a cf 2f ca 49 e1 72 4b 4d 4d 01	Z.P.*./l.fKw.

The dialog box also shows 'Mode: Binary', 'Type of data currently in cell: Binary', and '484 byte(s)'. An arrow points to the hex viewer.

Assuming the note is not encrypted, the user can extract the note data by viewing the BLOB data in the ZDATA column. The screenshot shows the SQLite Database Browser opening the BLOB in a hex viewer. The highlighted file header “1f 8b 08” is the file signature for Gzip data. This archive can be saved to disk and opened with any GZIP compatible software.

# Notes Database [iOS 9+]: NotesStore.sqlite Notes—Note Contents [2]

The screenshot displays an iPhone Notes app interface. The note is titled "My Secrets" and is dated "September 4, 2016 at 4:31 PM". The content includes a checklist titled "Things to do!" with three items: "The next great app", "Conquer the world", and "Feed the cat". Below the checklist is a video thumbnail showing a cityscape at night. The background of the screenshot is a hex dump of the note's data, with the following highlighted sections:

```

06E6578  Ç      *  FThings to do!  The nex
5642074 t great app Conquer the world Feed t
A040801 he cat 0ø°      (
A100A04 E      (      K      (
B051A10 R      (      @      (
B071A10 ?      (      (      (
B091A12      (      ø      (
00D2001 <      (
A040802 (      (
0011A04 (      (
FFFFFF0F (      =      (      '---
EED1202 -----"8 x!1f+?DRÇ61Ä'! İ
B112A08 Z      ";f,qD ±±5B-É . Y *
1F58054 *      g      *      4- ~j»D:OR;iÄT
00D1000 (      *      g      *      ÝhóAðoF ±É fiñ p
A450801 *      g      *      µmâ9>@Gá=â319µ... *E
4343946 bA $05163C3E-E4C0-4F6B-8267-FF9E449F
20801 A1DC com.apple.quicktime-movie*
    
```

The example note contains a list of items and a movie.

Once unarchived, the note data will have a proprietary format, as shown above. The highlighted sections show the checklist; however, it may be hard to determine which items are checklist titles and which are checklist items.

The movie can be reviewed by taking note of the GUID next to the "com.apple.quicktime-movie" file-type string.

# Notes Database [iOS 9+]: NotesStore.sqlite Note Attachments/Media—In Database

- Step 1: Find the note number of interest that contains an attachment; look up in ZNOTE column
- Step 2: Get the Attachment Metadata number in ZMEDIA column (Z\_ENT == 4—timestamps, height/width, file size, type, duration)
- Step 3: Get the attachment GUID and filename (Z\_ENT == 8); go find media in file system

```

1 select ZICLOUDSYNCINGOBJECT.Z_PK,
2 ZICLOUDSYNCINGOBJECT.Z_ENT,ZICLOUDSYNCINGOBJECT.znote,ZICLOUDSYNCINGOBJECT.zmedia,ZICLOUDSYNCINGOBJECT.zidentifier,
3 datetime(ZICLOUDSYNCINGOBJECT.zmodificationdate+978307200,'unixepoch','localtime') as "Modification Time",
4 datetime(ZICLOUDSYNCINGOBJECT.zpreviewupdate+978307200,'unixepoch','localtime') as "Preview Update Time",
5 ZICLOUDSYNCINGOBJECT.ztypeuti,ZICLOUDSYNCINGOBJECT.zfilesize,ZICLOUDSYNCINGOBJECT.zduration,
6 ZICLOUDSYNCINGOBJECT.zsizeheight,ZICLOUDSYNCINGOBJECT.zsizewidth,ZICLOUDSYNCINGOBJECT.zfilename
7 from ZICLOUDSYNCINGOBJECT
8 left join Z_12NOTES on ZICLOUDSYNCINGOBJECT.z_pk == Z_12NOTES.Z_9NOTES
9 where ZICLOUDSYNCINGOBJECT.z_ent == 4 or ZICLOUDSYNCINGOBJECT.z_ent == 8
10 order by "Create Time"

```

Z_PK	Z_ENT	ZNOTE	ZMEDIA	ZIDENTIFIER	Modification Time	Preview Update Time	ZTYPEUTI	ZFILESIZE	ZDURATION	ZSIZEHEIGHT	ZSIZEWIDTH	ZFILENAME	
1	79	4	78	80	8A0976B7-4A8B-4CDA-A0D1-366171E42C2B	2016-02-11 16:09:34	2016-02-11 16:09:35	public.jpeg	1872097	0.0	3264.0	2448.0	NULL
2	108	4	107	109	C6EEEFDB-F5F7-4F40-8321-D1A5D0DAE153	2016-08-21 12:33:51	2016-08-21 12:33:52	public.jpeg	3380346	0.0	3264.0	2448.0	NULL
3	113	4	112	114	FA020FE0-D089-4C82-8F41-096DBE1AB583	2016-08-21 12:34:19	2016-08-21 12:34:19	com.apple.notes.sketch	0	0.0	566.4	629.6	NULL
4	122	4	119	123	05163C3E-E4C0-4F68-B267-FF9E449FA1DC	2016-08-21 12:36:30	2016-08-21 12:36:31	com.apple.quicktime-movie	24409651	40.06	720.0	1280.0	NULL
5	127	4	126	128	EEE54145-8653-4982-94CE-7CEF6A63AE17	2016-08-21 12:37:17	2016-08-21 12:37:17	com.apple.notes.sketch	0	0.0	871.2	744.8	NULL
6	80	8	NULL	NULL	48299DC6-97A3-4C0F-A83B-BCCDDCF744E0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	IMG_0132.JPG
7	109	8	NULL	NULL	1E115883-2111-4604-A85C-6F635836F601	NULL	NULL	NULL	NULL	NULL	NULL	NULL	IMG_0438.JPG
8	114	8	NULL	NULL	499BDC14-CF9C-45C9-8D60-6D83A9F208C4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	New Note drawing
9	123	8	NULL	NULL	F1034C91-83E4-42BD-B543-47D93C5DC4F3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	trim.B6A8F9CF-7EFD-4ABC-BF69-71A3CFB51237.MOV
10	128	8	NULL	NULL	C5F29B15-BBEA-41B6-A53B-00A6B2310BF3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	New Note.drawing

To associate an attachment (media) with an actual file, this query will have to be performed. This query extracts the note number, media number, GUID, timestamps, file type, file metadata, and filename. The numbered steps above can get you to associate a certain file's metadata with the actual filename.

Other types of "attachments" like URLs and location data may also want to be extracted. Examples of these are shown below:

ZSUMMARY	ZTITLE	ZTYPEUTI	ZURLSTRING
Filter	Filter	public.url	Filter
NULL	The White House	public.url	https://maps.apple.com/maps?address=1600%20Pennsylvania%2...
See the President's daily schedule, explore behind-the-s...	The White House	public.url	https://www.whitehouse.gov/
	Find a Shop - Stewart's Shops Stewart's Shops	public.url	http://www.stewartsshops.com/lind-a-shop-2/

Table:  ZICLOCATION

Z_PK	Z_ENT	Z_OPT	'PLACEUPDATEI	ZATTACHMENT	ZLATITUDE	ZLONGITUDE	ZPLACEMARK
Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	16	1	183	38.897517	-77.036542	BLOB



```
select
ZICCLOUDSYNCINGOBJECT.Z_PK,
ZICCLOUDSYNCINGOBJECT.Z_ENT,
ZICCLOUDSYNCINGOBJECT.znote,
ZICCLOUDSYNCINGOBJECT.zmedia,
ZICCLOUDSYNCINGOBJECT.zidentifier,
datetime(ZICCLOUDSYNCINGOBJECT.zmodificationdate+978307200,'unixepoch','local
time') as "Modification Time",
datetime(ZICCLOUDSYNCINGOBJECT.zpreviewupdatedate+978307200,'unixepoch','loca
ltime') as "Preview Update Time",
ZICCLOUDSYNCINGOBJECT.ztypeuti,
ZICCLOUDSYNCINGOBJECT.zfilesize,
ZICCLOUDSYNCINGOBJECT.zduration,
ZICCLOUDSYNCINGOBJECT.zsizeheight,
ZICCLOUDSYNCINGOBJECT.zsizewidth,
ZICCLOUDSYNCINGOBJECT.zfilename
from ZICCLOUDSYNCINGOBJECTleft join Z_12NOTES
on ZICCLOUDSYNCINGOBJECT.z_pk == Z_12NOTES.Z_9NOTES
where ZICCLOUDSYNCINGOBJECT.z_ent == 4 or ZICCLOUDSYNCINGOBJECT.z_ent == 8
order by "Create Time"
```

```

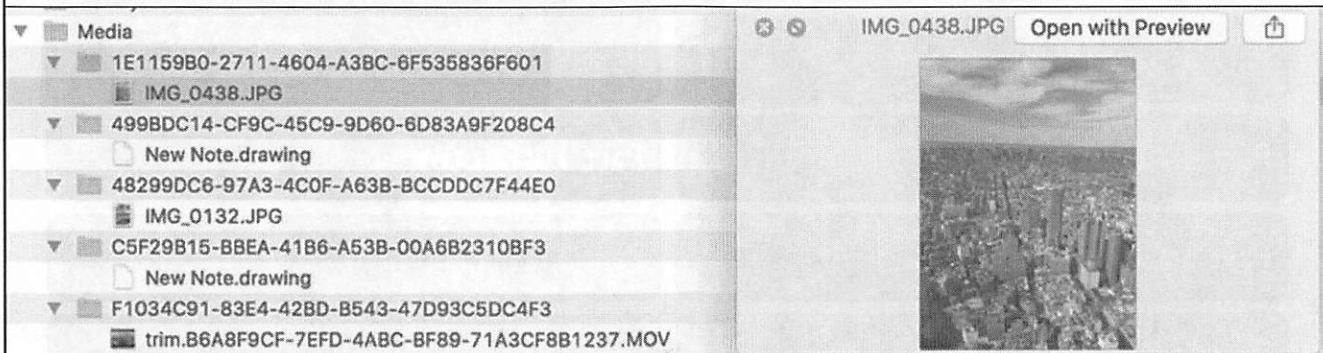
1 select ZICLOUDSYNCINGOBJECT.Z_PK,
2 ZICLOUDSYNCINGOBJECT.Z_ENT, ZICLOUDSYNCINGOBJECT.znote, ZICLOUDSYNCINGOBJECT.zmedia, ZICLOUDSYNCINGOBJECT.zidentifier,
3 datetime(ZICLOUDSYNCINGOBJECT.zmodificationdate+978307200, 'unixepoch', 'localtime') as "Modification Time",
4 datetime(ZICLOUDSYNCINGOBJECT.zpreviewupdatedate+978307200, 'unixepoch', 'localtime') as "Preview Update Time",
5 ZICLOUDSYNCINGOBJECT.ztypeuti, ZICLOUDSYNCINGOBJECT.zfilesize, ZICLOUDSYNCINGOBJECT.zduration,
6 ZICLOUDSYNCINGOBJECT.zsizeheight, ZICLOUDSYNCINGOBJECT.zsizewidth, ZICLOUDSYNCINGOBJECT.zfilename
7 from ZICLOUDSYNCINGOBJECT
8 left join Z_12NOTES on ZICLOUDSYNCINGOBJECT.z_pk == Z_12NOTES.Z_9NOTES
9 where ZICLOUDSYNCINGOBJECT.z_ent == 4 or ZICLOUDSYNCINGOBJECT.z_ent == 8
10 order by "Create Time"

```

	Z_PK	Z_ENT	ZNOTE	ZMEDIA	ZIDENTIFIER	Modification Time	Preview Update Time	ZTYPEUTI	ZFILESIZE	ZDURATION	ZSIZEHEIGHT	ZSIZEWIDTH	ZFILENAME
1	79	4	78	80	8A0976B7-4A8B-4CDA-A001-366171E42C2B	2016-02-11 16:09:34	2016-02-11 16:09:35	public.jpeg	1872097	0.0	3264.0	2448.0	NULL
2	108	4	107	109	C6EEEFDB-F5F7-4F40-8321-D1A5D0DAE153	2016-08-21 12:33:51	2016-08-21 12:33:52	public.jpeg	3380346	0.0	3264.0	2448.0	NULL
3	113	4	112	114	FA020FE0-D069-4C82-8F41-096DBE1AB583	2016-08-21 12:34:19	2016-08-21 12:34:19	com.apple.notes.sketch	0	0.0	566.4	629.6	NULL
4	122	4	119	123	05163C3E-E4C0-4F6B-B267-FF9E449FA1DC	2016-08-21 12:36:30	2016-08-21 12:36:31	com.apple.quicktime-movie	24409651	40.06	720.0	1280.0	NULL
5	127	4	126	128	EEE54145-B653-49B2-94CE-7CEF6A63AE17	2016-08-21 12:37:17	2016-08-21 12:37:17	com.apple.notes.sketch	0	0.0	871.2	744.8	NULL
6	80	8	NULL	NULL	48299DC6-97A3-4C0F-A63B-BCCDDC7F44E0	NULL	NULL	NULL	NULL	NULL	NULL	NULL	IMG_0132.JPG
7	109	8	NULL	NULL	1E1159B0-2711-4604-A3BC-8F535836F601	NULL	NULL	NULL	NULL	NULL	NULL	NULL	IMG_0438.JPG
8	114	8	NULL	NULL	499BDC14-CF9C-45C9-9D60-6D83A9F208C4	NULL	NULL	NULL	NULL	NULL	NULL	NULL	New Note.drawing
9	123	8	NULL	NULL	F1034C91-83E4-42BD-B543-47D93C5DC4F3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	trim.B6A8F9CF-7EFD-4ABC-BF89-71A3CF8B1237.MOV
10	128	8	NULL	NULL	C5F29B15-BBEA-41B6-A53B-00A6B2310BF3	NULL	NULL	NULL	NULL	NULL	NULL	NULL	New Note.drawing

## Notes Database [iOS 9+]: NotesStore.sqlite Note Attachments/Media—In File System

- Attachments are located in: /Media/<GUID\_from\_DB>/
- Filenames match those in database correlated with note/media rows



### Attachments:

iOS Physical/File System: <notesdir>/Media or /attachments

OS X: ~/Library/Group Containers/group.com.apple.notes/Media/

OS X: ~/Library/Containers/com.apple.Notes/Data/Library/CoreData/Attachments/

Once the attachment GUIDs are identified, the analyst can look for the GUID in the /Attachments or /Media file paths and find the attached file.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

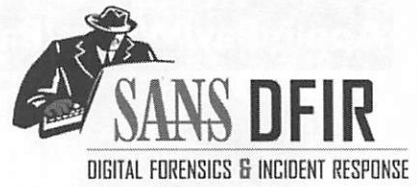
Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



---

## Section 4: Part 9

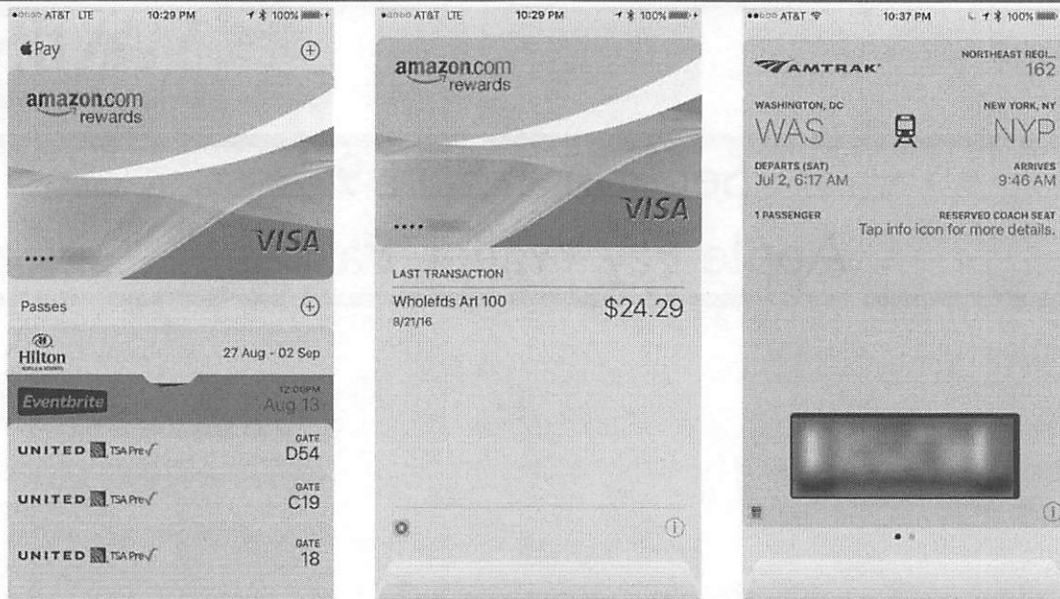
# Apple Pay, Wallet, Passes

---

This page intentionally left blank.



## Apple Pay/Wallet/Passes [1]



The Passes application, now called Wallet, can be used to keep track of various tickets and cards. If the user selects to add a credit card to the Apple Pay portion of this application, it can be used to purchase items as well, as shown in the middle screenshot.

## Apple Pay/Wallet/Passes [2]

### Apple Pay

- Credit Cards, Potential Transaction History

### Passes

- Tickets (Airline, Train, Movie, Events), Gift Cards, Reservations

### File/Directory Locations

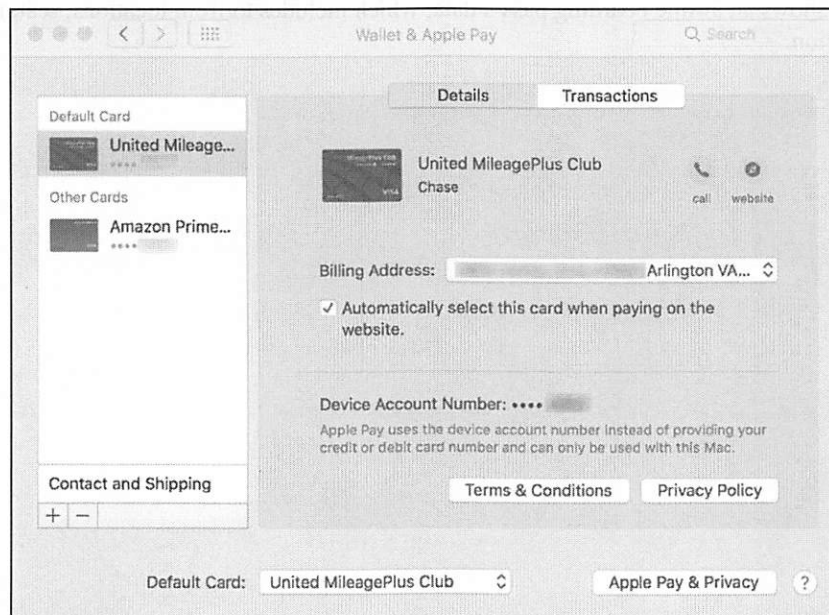
- iOS: May be duplicative
  - /private/var/mobile/Library/Passes/Cards/
  - (iCloud) /private/var/mobile/Library/Mobile Documents/com~apple~shoebox/UbiquitousCards/
- macOS (iCloud): ~/Library/Mobile Documents/com~apple~shoebox/UbiquitousCards/
- macOS: ~/Library/Passes/



A potential for transaction history may be available if Apple Pay is used in this application. If Wallet/Passes are used, each item is called a “Card”. Cards can be anything from an airline boarding pass to a train or event ticket to a booked hotel stay.

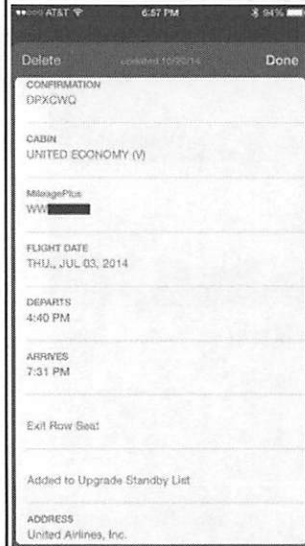
These items are synced using iCloud to all devices signed into the user’s iCloud account.

For those users with a new Mac with Touch ID, they can now set up credit cards on the Mac itself as shown below.



# Apple Pay/Wallet/Passes: Pass Structure

- \*.pkpass Directories
- Package Format



```
word:Cards oompa$ tree XhD-aahUoR1ADlmgTg0mB0MWA64=.pkpass
XhD-aahUoR1ADlmgTg0mB0MWA64=.pkpass
├── en.lproj
│   ├── logo.png
│   ├── logo@2x.png
│   ├── pass.strings
│   ├── icon.png
│   ├── icon@2x.png
│   ├── manifest.json
│   ├── pass.json
│   ├── signature
│   ├── strip.png
│   └── strip@2x.png
```

```
"headerFields": [
  {
    "label": "GATE",
    "key": "gate",
    "value": "D18",
    "changeMessage": "Your gate has changed to %@"
  }
],
"primaryFields": [
  {
    "label": "WASHINGTON-DULLES",
    "key": "origin",
    "value": "IAD"
  },
  {
    "label": "SAN FRANCISCO",
    "key": "destination",
    "value": "SFO"
  }
]
```

```
{
  "boardingPass": {
    "transitType": "PKTransitTypeAir",
    "auxiliaryFields": [
      {
        "label": "BOARDS",
        "key": "boardingTime",
        "value": "4:05 PM"
      },
      {
        "label": "FLIGHT",
        "key": "flight",
        "value": "UA 644"
      },
      {
        "label": "SEAT",
        "key": "seat",
        "value": "20A"
      }
    ],
    {
      "label": "SEQ",
      "key": "seq",
      "value": "36"
    },
    {
      "label": "GROUP",
      "key": "group",
      "value": "2"
    }
  ]
}
```

```
"secondaryFields": [
  {
    "label": "PREMIER SILVER / *S",
    "key": "passenger",
    "value": "EDWARDS/SARAH LMS"
  },
  {
    "key": "status",
    "value": "Premier Access"
  }
],
"backFields": [
  {
    "label": "CONFIRMATION",
    "key": "confirmation",
    "value": "DPXCWQ"
  },
  {
    "label": "CABIN",
    "key": "cabin",
    "value": "UNITED ECONOMY (V)"
  },
  {
    "label": "MileagePlus",
    "key": "mileagePlusNumber",
    "value": "WW"
  },
  {
    "label": "FLIGHT DATE",
    "key": "flightDate",
    "value": "THU., JUL 03, 2014"
  },
  {
    "label": "DEPARTS",
    "key": "departTime",
    "value": "4:40 PM"
  },
  {
    "label": "ARRIVES",
    "key": "arriveTime",
    "value": "7:31 PM"
  },
  {
    "key": "exitRow",
    "value": "Exit Row Seat"
  }
]
```

Each “Card” is stored in a \*.pkpass directory. This directory is actually in a “Package” format. This directory may include lots of files and pictures of various logos associated with that Card. The actual pass data is stored in the pass.json file.

The pass.json file can be reviewed in any text editor. Sometimes the developer of the “pass” doesn’t include whitespace, so reviewing them can be difficult. Look for an online or offline JSON file formatter.

The example above shows an airline boarding pass’s data, which includes to/from locations, seat, timeframes, and other flight information.

## Apple Pay/Wallet/Passes: passes23.sqlite (iOS) [1]

```
1 select
2 datetime(pass.ingested_date+978307200,'unixepoch','localtime') as "Ingested Date",
3 datetime(pass.modified_date+978307200,'unixepoch','localtime') as "Modified Date",
4 pass.unique_id,pass.organization_name,pass_type.identifier,payment_transaction.amount,payment_transaction.currency_code,
5 payment_transaction.merchant_industry_category,payment_transaction.merchant_name,
6 datetime(payment_transaction.transaction_date+978307200,'unixepoch','localtime') as "Transaction Date",
7 location.latitude,location.longitude,location.relevant_text,
8 datetime(pass.relevant_date+978307200,'unixepoch','localtime') as "Relevant Time",
9 datetime(pass.push_registration_date+978307200,'unixepoch','localtime') as "Push Reg Time",
10 pass.last_modified_tag
11 from pass
12 left join pass_type on pass.pass_type_pid == pass_type.pid
13 left join payment_transaction on pass.pid == payment_transaction.pass_pid
14 left join location_source on location_source.url like '%||pass.unique_id||%'
15 left join location on location.location_source_pid == location_source.pid
16 order by "Ingested Date" desc
```

	Ingested Date	Modified Date	unique_id	organization_name	Identifier
1	2016-08-29 07:49:29	2016-08-29 07:49:29	XhD-aahUoR1ADlmgG0mBOMWA64=	Hilton Worldwide	pass.com.hilton.hhonors.staypass
2	2016-08-20 22:29:53	2016-08-20 22:30:47	jNPZaLC9Nz5Sc8z5tgG-3nQaBM=	Chase	paymentpass.com.apple
3	2016-08-13 11:24:03	2016-08-13 11:24:03	r3YRuCrngxzjXNyCtElaqEPkdSMQ=	MacDMV August Meetup	pass.eventbrite.ticket
4	2016-08-07 13:33:18	2016-08-07 13:33:25	v4aox00e6iJo0YYuoX76qW83zsQ=	United Airlines	pass.united.UnitedMobileBoardingPass
5	2016-08-03 06:14:20	2016-08-03 06:14:24	hu0oQlQfHhKklg8cbqq5UEL-Ekk=	United Airlines	pass.united.UnitedMobileBoardingPass

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 151

To extract the information for each Card from the `passes23.sqlite` database, the following SQLite query was used. This query extracts lots of information. This is only a partial view of the extracted data; it is continued on the next slide. The data shown includes the “ingested data”, when the Card was downloaded to the applications, a last modified timestamp, the unique Card ID, organization name, and bundle identifier. The timestamps are in Mac Epoch format.

```
select
datetime(pass.ingested_date+978307200,'unixepoch','localtime') as "Ingested Date",
datetime(pass.modified_date+978307200,'unixepoch','localtime') as "Modified Date",
pass.unique_id,
pass.organization_name,
pass_type.identifier,
payment_transaction.amount,
payment_transaction.currency_code,
payment_transaction.merchant_industry_category,
payment_transaction.merchant_name,
datetime(payment_transaction.transaction_date+978307200,'unixepoch','localtime') as
"Transaction Date",
location.latitude,
location.longitude,
location.relevant_text,
datetime(pass.relevant_date+978307200,'unixepoch','localtime') as "Relevant Time",
datetime(pass.push_registration_date+978307200,'unixepoch','localtime') as "Push Reg
Time",
pass.last_modified_tag
from pass
left join pass_type on pass.pass_type_pid == pass_type.pid
left join payment_transaction on pass.pid == payment_transaction.pass_pid
left join location_source on location_source.url like '%||pass.unique_id||%'
left join location on location.location_source_pid == location_source.pid
order by "Ingested Date" desc
```



```

1  select
2  datetime(pass.ingested_date+978307200,'unixepoch','localtime') as "Ingested Date",
3  datetime(pass.modified_date+978307200,'unixepoch','localtime') as "Modified Date",
4  pass.unique_id,pass.organization_name,pass_type.identifier,payment_transaction.amount,payment_transaction.currency_code,
5  payment_transaction.merchant_industry_category,payment_transaction.merchant_name,
6  datetime(payment_transaction.transaction_date+978307200,'unixepoch','localtime') as "Transaction Date",
7  location.latitude,location.longitude,location.relevant_text,
8  datetime(pass.relevant_date+978307200,'unixepoch','localtime') as "Relevant Time",
9  datetime(pass.push_registration_date+978307200,'unixepoch','localtime') as "Push Reg Time",
10 pass.last_modified_tag
11 from pass
12 left join pass_type on pass.pass_type_pid == pass_type.pid
13 left join payment_transaction on pass.pid == payment_transaction.pass_pid
14 left join location_source on location_source.url like '%|||pass.unique_id|||%'
15 left join location on location.location_source_pid == location_source.pid
16 order by "Ingested Date" desc

```

	Ingested Date	Modified Date	unique_id	organization_name	identifier
1	2016-08-29 07:49:29	2016-08-29 07:49:29	XhD-aahUoR1ADlmgTg0mBOMWA64=	Hilton Worldwide	pass.com.hilton.hhonors.staypass
2	2016-08-20 22:29:53	2016-08-20 22:30:47	jnPZaLC9Nz5Sc8z5tgG-3nQaBM=	Chase	paymentpass.com.apple
3	2016-08-13 11:24:03	2016-08-13 11:24:03	r3YRuCnxgzjXNyCtEiaqEPkdSMQ=	MacDMV August Meetup	pass.eventbrite.ticket
4	2016-08-07 13:33:18	2016-08-07 13:33:25	v4aox00e6iJo0YYuoX76qW83zsQ=	United Airlines	pass.united.UnitedMobileBoardingPass
5	2016-08-03 06:14:20	2016-08-03 06:14:24	hu0oQIQfhHKkIg8cbqq5UEL-Ekk=	United Airlines	pass.united.UnitedMobileBoardingPass



## Apple Pay/Wallet/Passes: passes23.sqlite (iOS) [2]

- Apple Pay Transactions and/or Location/Geofence Information

```

1 select
2 datetime(pass.ingested_date+978387200,'unixepoch','localtime') as "Ingested Date",
3 datetime(pass.modified_date+978387200,'unixepoch','localtime') as "Modified Date",
4 pass.unique_id,pass.organization_name,pass_type.identifier,payment_transaction.amount,payment_transaction.currency_code,
5 payment_transaction.merchant_industry_category,payment_transaction.merchant_name,
6 datetime(payment_transaction.transaction_date+978387200,'unixepoch','localtime') as "Transaction Date",
7 location.latitude,location.longitude,location.relevant_text,
8 datetime(pass.relevant_date+978387200,'unixepoch','localtime') as "Relevant Time",
9 datetime(pass.push_registration_date+978387200,'unixepoch','localtime') as "Push Reg Time",
10 pass.last_modified_tag
11 from pass
12 left join pass_type on pass.pass_type_pid == pass_type.pid
13 left join payment_transaction on pass.pid == payment_transaction.pass_pid
14 left join location_source on location_source.url like '%[pass.unique_id]%'
15 left join location on location.location_source_pid == location_source.pid
16 where amount > 0 or latitude > 0
    
```

	organization_name	identifier	amount	currency_code	merchant_industry_category	merchant_name	Transaction Date	latitude	longitude	relevant_text	Relevant Time
1	Chase	paymentpass.com.apple	2429	USD	SUPERMARKETS	WHOLEFDS ARL 100	2016-08-21 14:41:29	NULL	NULL	NULL	NULL
2	Amtrak	pass.com.amtrak.rider.pnr	NULL	NULL	NULL	NULL	NULL	40.750327	-73.994459	Your departure station is nearby.	2016-07-03 15:00:00
3	Amtrak	pass.com.amtrak.rider.pnr	NULL	NULL	NULL	NULL	NULL	38.896993	-77.006422	Your departure station is nearby.	2016-07-02 08:17:00
4	Lufthansa	pass.com.lufthansa.mbp	NULL	NULL	NULL	NULL	NULL	48.3539	11.7861	Lufthansa LH414 Boarding 15:15	2014-07-26 09:15:00
5	Lufthansa	pass.com.lufthansa.mbp	NULL	NULL	NULL	NULL	NULL	59.6519	17.9186	Lufthansa LH2415 Boarding 11:25	2014-07-26 05:25:00
6	Lufthansa	pass.com.lufthansa.mbp	NULL	NULL	NULL	NULL	NULL	38.9444	-77.4558	Lufthansa LH419 Boarding 17:30	2014-07-17 17:30:00
7	Hilton Worldwide	pass.com.hilton.hhonors.staypass	NULL	NULL	NULL	NULL	NULL	36.858906	-75.97771	Hilton Virginia Beach Oceanfront	NULL

**SANS** | **DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 153

The Apple Pay transactions and location/geofence data are continued here from the previous slide.

The transactions information includes the amount in US dollars (`currency_code`) and shows the vendor information. This data is for a transaction at a Whole Foods supermarket for \$24.29.

The location and geofence data may appear if the developer of the pass included it so that the pass/ticket would show up on the user's screen when the user entered that specific location at the "relevant time".

```

1 select
2 datetime(pass.ingested_date+978307200,'unixepoch','localtime') as "Ingested Date",
3 datetime(pass.modified_date+978307200,'unixepoch','localtime') as "Modified Date",
4 pass.unique_id,pass.organization_name,pass_type.identifier,payment_transaction.amount,payment_transaction.currency_code,
5 payment_transaction.merchant_industry_category,payment_transaction.merchant_name,
6 datetime(payment_transaction.transaction_date+978307200,'unixepoch','localtime') as "Transaction Date",
7 location.latitude,location.longitude,location.relevant_text,
8 datetime(pass.relevant_date+978307200,'unixepoch','localtime') as "Relevant Time",
9 datetime(pass.push_registration_date+978307200,'unixepoch','localtime') as "Push Reg Time",
10 pass.last_modified_tag
11 from pass
12 left join pass_type on pass.pass_type_pid == pass_type.pid
13 left join payment_transaction on pass.pid == payment_transaction.pass_pid
14 left join location_source on location_source.url like '%"||pass.unique_id||'%'
15 left join location on location.location_source_pid == location_source.pid
16 where amount > 0 or latitude > 0

```

	organization_name	identifier	amount	currency_code	merchant_industry_category	merchant_name	Transaction Date	latitude	longitude	relevant_text	Relevant Time
1	Chase	paymentpass.com.apple	2429	USD	SUPERMARKETS	WHOLEFDS ARL 100	2016-08-21 14:41:29	NULL	NULL	NULL	NULL
2	Amtrak	pass.com.amtrak.rider.pnr	NULL	NULL	NULL	NULL	NULL	40.750327	-73.994459	Your departure station is nearby.	2016-07-03 15:00:00
3	Amtrak	pass.com.amtrak.rider.pnr	NULL	NULL	NULL	NULL	NULL	38.896993	-77.006422	Your departure station is nearby.	2016-07-02 06:17:00
4	Lufthansa	pass.com.lufthansa.mbp	NULL	NULL	NULL	NULL	NULL	48.3539	11.7861	Lufthansa LH414 Boarding 15:15	2014-07-26 09:15:00
5	Lufthansa	pass.com.lufthansa.mbp	NULL	NULL	NULL	NULL	NULL	59.6519	17.9186	Lufthansa LH2415 Boarding 11:25	2014-07-26 05:25:00
6	Lufthansa	pass.com.lufthansa.mbp	NULL	NULL	NULL	NULL	NULL	38.9444	-77.4558	Lufthansa LH419 Boarding 17:30	2014-07-17 17:30:00
7	Hilton Worldwide	pass.com.hilton.hhonors.staypass	NULL	NULL	NULL	NULL	NULL	36.858906	-75.97771	Hilton Virginia Beach Oceanfront	NULL

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



**SANS DFIR**

DIGITAL FORENSICS & INCIDENT RESPONSE

---

## Section 4: Part 10

### Photos

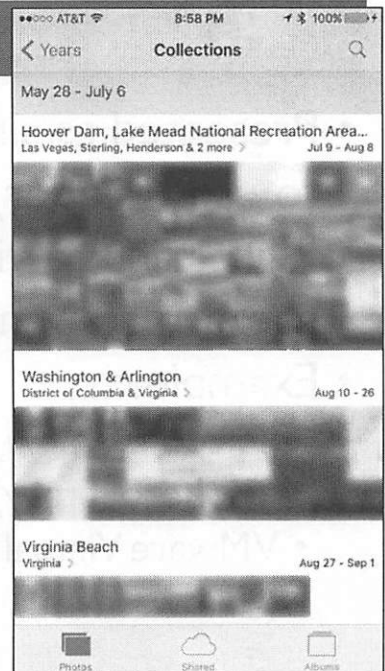
---

**SANS | DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 156

This page intentionally left blank.

## Photos: Photo Album Application



SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 157

The new Photos application changed everything under the hood from its predecessor iPhoto.

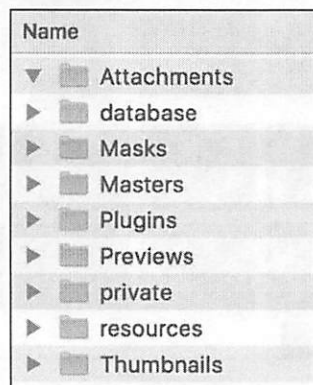
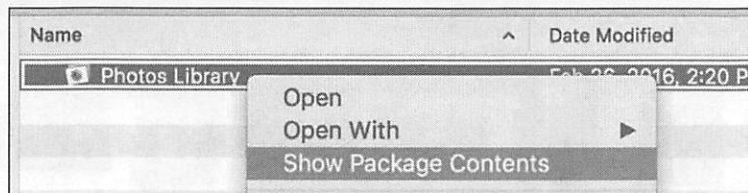
The User's macOS Photo library can be found in a package file: `~/Pictures/Photos Library.photoslibrary/`

Where local photos and iCloud photos were separate before, Photos introduces an integrated format.



## macOS Photos Library: Packages “File” Format

- Presented as single file to user (Finder)
- Right-click to view package contents
- Examples:
  - iPhoto Library (\*.iphotolibrary)
  - VMware Virtual Machines (\*.vmwarevm)



Packages are directories that are presented to the user as a single file in Finder. To view the contents of a package, right-click and choose “Show Package Contents” or view it in a terminal window.

Some of the more well-known packages are the iTunes Library and VMware Fusion virtual machines.

### Reference:

Apple Developer Website: About Bundles

[https://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/AboutBundles/AboutBundles.html#//apple\\_ref/doc/uid/10000123i-CH100-SW1](https://developer.apple.com/library/mac/#documentation/CoreFoundation/Conceptual/CFBundles/AboutBundles/AboutBundles.html#//apple_ref/doc/uid/10000123i-CH100-SW1)

## macOS Photos: Masters Directory

~/Pictures/Photos Library.photoslibrary/

- Masters Directory: The photos themselves
  - JPG: Photos
  - PNG: Screenshots
  - MOV: Movies
  - iOS 11: \*.HEIC (HEIF: High Efficiency Image Format): Open in Preview.app
- Timestamped File Paths
- Extended Attribute: `com.apple.quarantine` = `cloudphotosd, iCloud`

```
word:Masters oompa$ pwd
/Users/oompa/Pictures/Photos Library.photoslibrary/Masters
word:Masters oompa$ tree -L 4 2015/
2015/
├── 01
│   ├── 11
│   │   ├── 20150111-174805
│   │   │   └── IMG_1629.JPG
│   │   └── 20150111-174816
│   └── 24
│       ├── 20150124-130736
│       │   ├── IMG_1630.JPG
│       │   └── IMG_1631.JPG
│       └── 20150124-130748
├── 02
│   └── 07
│       ├── 20150207-095600
│       │   ├── IMG_1632.JPG
│       │   ├── IMG_1633.JPG
│       │   ├── IMG_1634.JPG
│       │   ├── IMG_1635.JPG
│       │   └── 20150207-095613
│       └── 11
│           ├── 20150211-205726
│           │   ├── IMG_1636.JPG
│           │   └── 20150211-205737
│           └── 14
│               ├── 20150214-235059
│               │   ├── IMG_1637.MOV
│               │   ├── IMG_1638.MOV
│               │   └── IMG_1639.MOV
└── 03
```

```
word:20150516-225727 oompa$ xattr -xl IMG_1916.PNG
com.apple.quarantine:
00000000 30 30 30 32 38 35 35 35 37 63 64 65 39 38 63 6C |0002;5557cde9;cl|
00000010 6F 75 64 70 68 6F 74 6F 73 64 3B |oudphotosd;|
0000001b
```

SANS | DFIR

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 159

The original photos can be found in the Masters directory. The photos (and other media) are stored in a timestamped file path.

Since local and iCloud photos are now stored together, it might be necessary to determine which are iCloud pictures. This can be shown in the extended attributes for each file. If the `com.apple.quarantine` attribute contains `cloudphotosd` as its Bundle ID, then it came from iCloud.

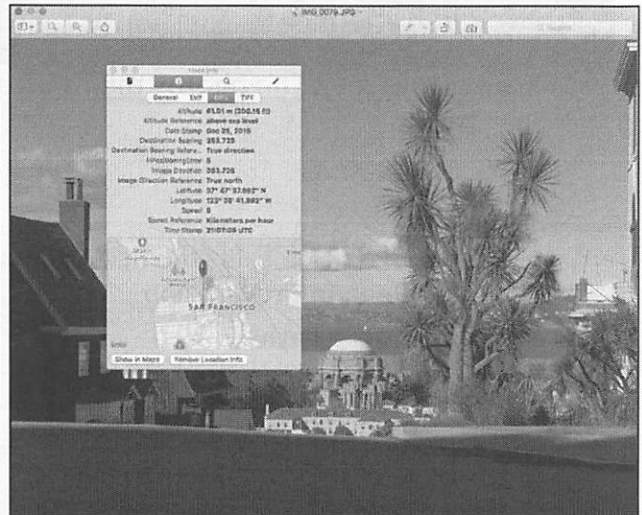
Pictures taken with devices running iOS 11 will use a new format for storage called High Efficiency Image Format.

Reference:

[https://en.wikipedia.org/wiki/High\\_Efficiency\\_Image\\_File\\_Format](https://en.wikipedia.org/wiki/High_Efficiency_Image_File_Format)

## Photo EXIF Data

```
byte:100APPLE oompas exiftool IMG_0079.JPG | grep GPS
GPS Latitude Ref      : North
GPS Longitude Ref    : West
GPS Altitude Ref     : Above Sea Level
GPS Time Stamp       : 21:07:06
GPS Speed Ref        : km/h
GPS Speed             : 0
GPS Img Direction Ref : True North
GPS Img Direction    : 353.7246377
GPS Dest Bearing Ref : True North
GPS Dest Bearing     : 353.7246377
GPS Date Stamp       : 2015:12:25
GPS Horizontal Positioning Error: 5 m
GPS Altitude         : 61 m Above Sea Level
GPS Date/Time        : 2015:12:25 21:07:06Z
GPS Latitude         : 37 deg 47' 37.96" N
GPS Longitude        : 122 deg 26' 41.89" W
GPS Position         : 37 deg 47' 37.96" N, 122 deg 26' 41.89" W
```



Photos found on iOS devices may contain GPS location coordinates if the user has authorized the application or Camera app to do so.

Two ways to show this information are shown; however, many applications and forensic software suites can provide similar output. On the left is an example of the output of “exiftool”, a command-line utility for parsing EXIF data, while on the right is a screenshot using the Preview applications Data Inspector to view the same data.

## Photo Databases

### macOS

- Older (10.10): ~/Pictures/Photos Library.photoslibrary/Databases/Library.apdb (Link to /apdb/Library.apdb)
- Newer (10.11+): ~/Pictures/Photos Library.photoslibrary/database/photos.db

### iOS:

- Physical: /private/var/mobile/Media/PhotoData/Photos.sqlite
- Backup: /mobile/Media/PhotoData/Photos.sqlite

The metadata for each photo (or other media) is found in the `Library.apdb/Photos.db/Photos.sqlite` databases. These databases contain nearly the same type of data; however, the table and column names may differ.

There are many other files in the `Photos.photoslibrary` package file on macOS and the `PhotoData` directory on iOS, including other databases, plist files, thumbnail photos, etc.

## macOS Photos: Photo Metadata

### macOS Photo Database: Photos.db (Photos.sqlite on iOS)

- Filename
- Timestamps (imageDate, Create, Export Image, Export Metadata)
- Height/Width/Rotation
- Associated Notes Flag
- Location Latitude/Longitude
- Time Zone
- Reversed Location Blob Data (Similar to Reverse IP Location)
- Albums/Moments/Faces/Places
- More!

macOS: ~/Pictures/Photos Library.photoslibrary/database/photos.db  
iOS Physical: /private/var/mobile/Media/PhotoData/Photos.sqlite  
iOS Backup: /mobile/Media/PhotoData/Photos.sqlite

An example from the Library.apdb database contains the metadata for each photo item. This can include a variety of different items such as file metadata, timestamps, time zone info, user comments, and location-based data, just to include some of what is available.



# iOS Photos: Photos.sqlite – Reverse Geo Location

The screenshot displays a database table named 'ZADDITIONALASSETATTRIBUTES'. The table has columns for various metadata fields, including 'ZREVERSELOCATIONDATA'. A BLOB is present in this column, which is expanded in a separate window to show a plist of geolocation data. An inset map shows the location in San Francisco, specifically the Upper Market, Castro & Pacific Heights area.

IGINALASSETSUA	ZORIGINALFILENAME	ZORIGINALPATH	PUBLICGLOBALULI	ZTIMEZONE	ZTITLE	ZFACE	ZORIGINALHASH	ZACEANNO	ZREVERSELOCATIONDATA
1	IMG_0079.JPG	DCIM100APPLE	F8E107B3-7410...	NULL	NULL	NULL	NULL	NULL	BLOB

```

0000 62 70 4c 89 73 74 30 30 d5 01 02 03 04 05 06 07  bplist00.....
0010 08 09 0a 0b 0c 0d 0e 0f 10 11 12 13 14 15 16 17  ...:Vishnuprovi
0020 64 65 72 59 64 5b 70 72 6f 76 69 64 65 72 58 65  de:providerVe
0030 72 5e 67 55 65 50 6c 63 65 52 65 73 75 6c 74  r"geolocateResult
0040 57 76 45 72 73 69 6f 6e 08 34 37 36 21 38 10 09  Wessim..79818..
0050 4f 11 03 41 0a 0d 02 38 39 22 11 32 37 35 39 20  C.A....9".2759"
0060 42 72 6f 64 65 72 69 63 6b 20 53 74 32 64 01 5a  Broderick St..t
0070 11 32 37 35 38 20 42 72 65 64 65 72 69 63 6b 20  .2759 Broderick
0080 53 74 5a 18 53 61 6e 20 46 72 61 6e 83 69 73 63  St.San Francisc
0090 61 2c 20 43 41 20 20 39 34 31 22 33 3a 0d 35 6e  C. CA #1123K.un
0100 69 74 85 84 20 53 74 61 74 65 72 78 83 01 0a 0d  lid States...
0110 25 6e 69 74 65 64 20 53 74 61 74 65 73 12 02 55  United States..C
0120 53 7a 0a 43 61 6c 59 66 6f 72 6e 69 61 22 02 43  S..California".C
0130 41 2a 0d 53 61 6e 20 46 72 61 6e 83 69 73 63 6f  A".San Francisc
0140 02 0d 53 61 6e 20 46 72 61 6e 83 69 73 63 6f 3a  2.San Francisco:
0150 05 39 34 31 32 33 42 6f 61 63 69 66 65 63 20  #9113B.Pacific
0160 48 65 69 67 68 74 73 32 6f 42 72 6f 64 65 72 69  HeightR.Broderi
0170 63 6b 20 53 74 3a 04 25 35 39 62 11 32 37 35  C# St#.2759b.275
0180 39 20 42 72 65 64 65 63 68 8b 20 53 74 6a 04  # Broderick St.
0190 33 38 30 33 8a 01 02 03 04 05 06 07 08 09 0a 0b  3803...Pacific H
    
```

macOS: ~/Pictures/Photos Library.photoslibrary/database/photos.db  
 iOS Physical: /private/var/mobile/Media/PhotoData/Photos.sqlite  
 iOS Backup: /mobile/Media/PhotoData/Photos.sqlite

The Photos.sqlite database file contains many tables. In the screenshot above, the ZADDITIONALASSETATTRIBUTES table is shown. This table contains the metadata for each photo in the database to include filename, file size, height/width, and reverse location data.

If the photo has location data embedded in it, Photos will show a BLOB in the ZREVERSELOCATIONDATA table. This BLOB contains a plist showing human-readable reversed geolocation data.

Table: ZADDITIONALASSETATTRIBUTES New Record

IGINALASSETSUL	ZORIGINALFILENAME	ZORIGINALPATH	PUBLICGLOBALUUI	ZTIMEZONENAME	ZTITLE	ZFACEREGIONS	ZORIGINALHASH	ACEANNOTATIOND	ZREVERSELOCATIONDATA
tor	IMG_0079	Filter	Filter	Filter	Filter	Filter	Filter	Filter	0
1 NULL	IMG_0079.JPG	DCIM/100APPLE	F8E1D7B3-7410-...	NULL	NULL	NULL	NULL	NULL	BLOB

Edit database cell

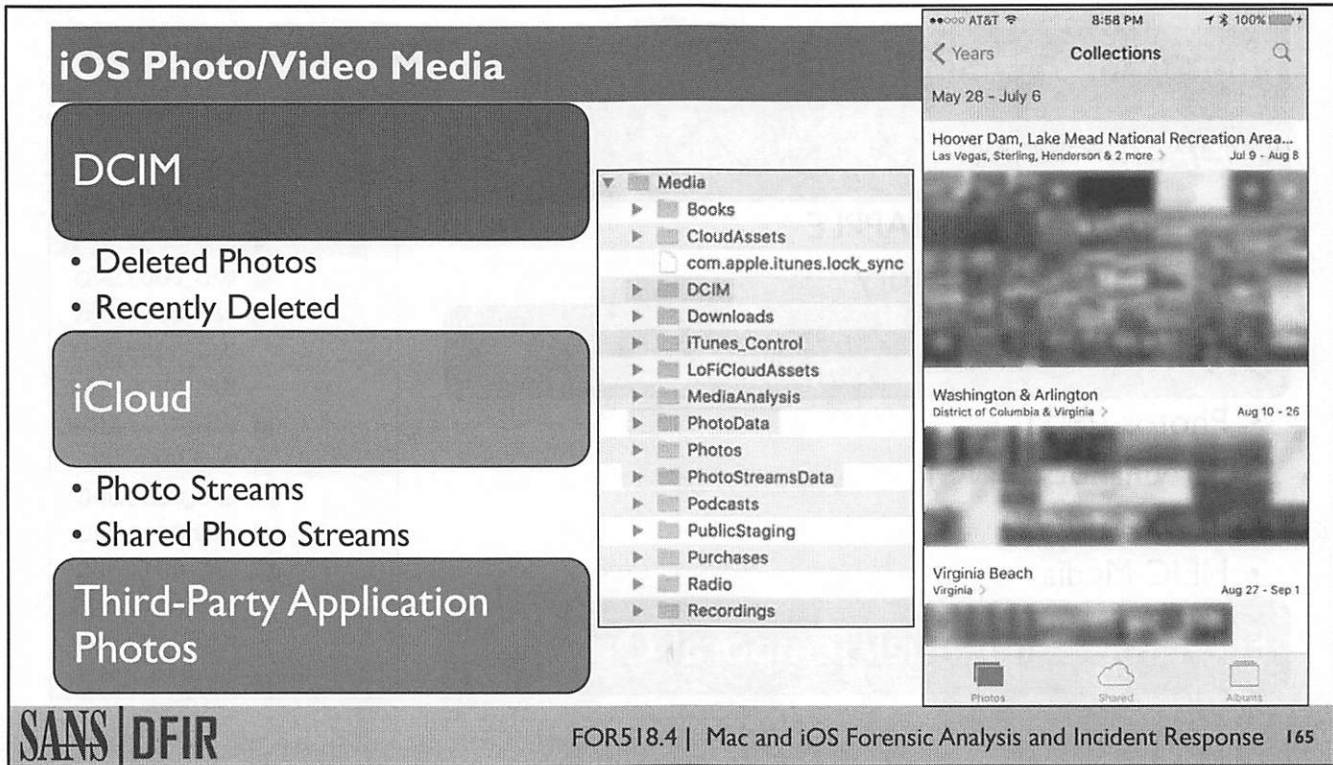
Binary  Clear

Import    Export

```

0000 62 70 6c 69 73 74 30 30 d5 01 02 03 04 05 06 07 bplist00.....
0010 08 09 0a 56 69 73 48 6f 6d 65 5a 70 72 6f 76 69 ...VisHome2provi
0020 64 65 72 49 64 5b 70 72 6f 76 69 64 65 72 56 65 derId(providerVe
0030 72 5e 67 65 6f 50 6c 61 63 65 52 65 73 75 6c 74 r"geoPlaceResult
0040 57 76 65 72 73 69 6f 6e 08 54 37 36 31 38 10 09 Wversion.T7618..
0050 4f 11 03 41 0a ad 02 18 39 22 11 32 37 35 39 20 O..A....9".2759
0060 42 72 6f 64 65 72 69 63 6b 20 53 74 32 e4 01 5a Broderick St2..Z
0070 11 32 37 35 39 20 42 72 6f 64 65 72 69 63 6b 20 .2759 Broderick
0080 53 74 5a 18 53 61 6e 20 46 72 61 6e 63 69 73 63 StZ.San Francisc
0090 6f 2c 20 43 41 20 20 39 34 31 32 33 5a 0d 55 6e o, CA 94123Z.Un
00a0 69 74 65 64 20 53 74 61 74 65 73 7a a5 01 0a 0d ited Statesz....
00b0 55 6e 69 74 65 64 20 53 74 61 74 65 73 12 02 55 United States..U
00c0 53 1a 0a 43 61 6c 69 66 6f 72 6e 69 61 22 02 43 S..California".C
00d0 41 2a 0d 53 61 6e 20 46 72 61 6e 63 69 73 63 6f A".San Francisco
00e0 32 0d 53 61 6e 20 46 72 61 6e 63 69 73 63 6f 3a 2.San Francisco:
00f0 05 39 34 31 32 33 42 0f 50 61 63 69 66 69 63 20 .94123B.Pacific
0100 48 65 69 67 68 74 73 52 0c 42 72 6f 64 65 72 69 HeightsR.Broderi
0110 63 6b 20 53 74 5a 04 32 37 35 39 62 11 32 37 35 ck StZ.2759b.275
0120 39 20 42 72 6f 64 65 72 69 63 6b 20 53 74 6a 04 9 Broderick Stj.
0130 33 38 30 33 8a 01 0f 50 61 63 69 66 69 63 20 48 3803...Pacific H

```



Photos, videos, and screenshots can be strewn about a device in many directories, depending on how the user uses their device.

These directories include the DCIM, PhotoData, and PhotoStreamsData directories.

The native Photos application is shown to the right.

## Photo/Video Media: DCIM

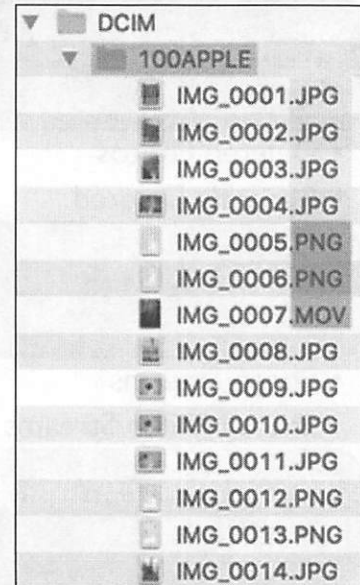
### 100APPLE

- ... 101APPLE ... 102APPLE ...
- 999 items per directory

### IMG\_####.EXT

- Photos (JPG)
- Screenshots (PNG)
- Movies (MOV)
- HEIC Media

### Filename Sequential/Temporal Order



Physical: iOS 6+: /private/var/mobile/Media/DCIM/  
Backup/FS: iOS 6+: /mobile/Media/DCIM/

Photos, videos, and screenshots that are taken with the device will be stored in the DCIM directory. This directory works just like it does on other camera devices. This directory may contain multiple other directories named 100APPLE, 101APPLE, and so on. Each of these may contain up to 999 photos, videos, and/or screenshots.

Each image is named sequentially starting with IMG\_0001.EXT, where EXT can be JPG for a photo, MOV for a movie, or PNG for a screenshot.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



---

## Section 4: Part II

### Maps

---

This page intentionally left blank.

## Maps: Files and Directories

### iOS Physical

- [iOS 8+]  
/private/var/mobile/Containers/Data/<GUID>/Library/Maps/

### iOS Backup/File System

- /mobile/Applications/com.apple.Maps/

### macOS

- ~/Library/Containers/com.apple.Maps/



The native mapping application is Apple Maps. The Maps application is available for both iOS and macOS, and the files are very similar. Maps data can be synced to all devices using iCloud.



## Maps: History and Searches (\*.mapsdata, \*.plist)

### [iOS 8+/macOS] GeoHistory.mapsdata / MSPFailedSearches.mapsdata

- Also copies in Apple Watch directories
- Binary Plist or just Data BLOB
  - Protobuf Data BLOB Format
- Item 0 (Most Recent)
- Item n (Least Recent)

### Map Bookmarks

- [iOS 8+/macOS]: GeoBookmarks.plist

```
43 39 45 41 ...$5BDEA8A2-4A9D-483E-9F7A-C9EA
40 0A CF 40 176ED341....P..A!...P..A:.0..0
C0 28 01 12 .....pC@.7Hl..FS.(.
EA 70 43 40 N.L".Current Location*$)....pC@
12 09 C0 E2 17Hl..FS.9....pC@A7Hl..FS.J....
48 6C 80 B1 ...pC@.7Hl..FS.....pC@.7Hl..
12 09 72 03 FS...?.....M...r.
74 61 74 65 ~-4nC@.....ES,"...United State
6F 6E 32 09 s..US..Virginia".VA*.Arlington2.
69 68 65 52 Arlington:.22204B.Columbia PikeR
65 62 65 20 .S Glebe RdZ.1014b.1014 S Glebe
20 58 69 68 Rd...Douglas Park...Columbia Pik
65 65 32 0F e*.Sugar Shack Donuts & Coffee2.
56 41 20 20 1014 S Glebe Rd2.Arlington, VA
CA F3 D1 F8 222042.United States.==.....
20 C0 D1 02 .....(.".....
20 C0 D1 02 (.0.".....(.".....
6D 2E 79 65 (.0.".....(.0.:...com.ye
79 65 6C 70 lp..7cic-y9jf10nE2ldkZtd4g"%yelp
5A 74 64 34 5.3://biz/7cic-y9jf10nE2ldkZtd4
45 32 6C 64 g"%yelp4://biz/7cic-y9jf10nE2ld
66 31 4F 6E kZtd4g"%yelp://biz/7cic-y9jf10n
62 69 7A 2F E2ldkZtd4g"*http://yelp.com/biz/
08 02 08 03 7cic-y9jf10nE2ldkZtd4gB.J.....
10 E0 EA 04 .....B.J.....
6C 70 12 16 ". .....(.0.:...com.yelp..
35 2E 33 3A 7cic-y9jf10nE2ldkZtd4g"%yelp5.3:
67 22 23 79 //biz/7cic-y9jf10nE2ldkZtd4g"#y
```

iOS 8+/macOS: GeoHistory.mapsdata, MSPFailedSearches.mapsdata  
iOS 7-: History.mapsdata/Directions.mapsdata

In iOS 7 and older devices, the maps data is stored in a proprietary format file. These files may still exist when a user upgrades to newer operating systems. The ASCII location strings can be viewed either in a hex editor or using the “strings” command. The format of these files is not documented.

On newer iOS 8+ devices, the \*.mapsdata files are stored in a slightly easier-to-read plist file but still contain BLOB location data that is hard to parse. The BLOB data can be extracted and viewed in a hex editor to determine the location data.

It is worth noting that non-ASCII Unicode strings used for other languages may be harder to read using a hex editor or the “strings” command.

# Maps: GeoHistory.mapsdata (plist file) [iOS 8+/macOS]

Key	Type	Value
Root	Dictionary	(2 Items)
MSPHistoryVersion	Number	1
MSPHistory	Array	(20 Items)
Item 0	Data	<08021224 44364145 353
Item 1	Data	<08031224 45433633 374
Item 2	Data	<08021224 37454642 444
Item 3	Data	<08011224 32423934 343
Item 4	Data	<08021224 43304546 423
Item 5	Data	<08031224 37323539 313
Item 6	Data	<08031224 42313534 343
Item 7	Data	<08011224 46373544 343
Item 8	Data	<08031224 45363236 383
Item 9	Data	<08021224 30304544 323
Item 10	Data	<08031224 45343837 344
Item 11	Data	<08031224 44453131 353
Item 12	Data	<08021224 35424445 413
Item 13	Data	<08031224 44303144 433
Item 14	Data	<08031224 35433430 324
Item 15	Data	<08021224 31313534 313
Item 16	Data	<08021224 30363642 333
Item 17	Data	<08031224 39464334 393
Item 18	Data	<08021224 34333130 413
Item 19	Data	<08011224 33364439 373

```
...$DE115720-59A8-4928-8476-6296
F2ED7692.....A1.....AB.....
.....[.....].....(.0:..
.....com.yelp..Fs_XnZbSIRywMaPAiAX
hNQ"yelp5.3:////biz/Fs_XnZbSIRyw
MaPAiAXhNQ"yelp4:////biz/Fs_XnZb
SIRywMaPAiAXhNQ"yelp:////biz/Fs_
XnZbSIRywMaPAiAXhNQ"htp://yelp
.com/biz/Fs_XnZbSIRywMaPAiAXhNQ@
@.....District Taco..Mexican...
@.....?.....@*..Arlington0...
.....(.0:.....).....
.....com.yelp..Fs_XnZbSIRywMaP
AiAXhNQ"yelp5.3:////biz/Fs_XnZbS
IRywMaPAiAXhNQ"yelp4:////biz/Fs_
XnZbSIRywMaPAiAXhNQ"yelp:////bi
z/Fs_XnZbSIRywMaPAiAXhNQ"htp://
yelp.com/biz/Fs_XnZbSIRywMaPAiAX
hNQ@.2.....@B.2...
(.0:.....).....@.....
.....com.yelp..Fs_XnZbSIR
ywMaPAiAXhNQ"yelp5.3:////biz/Fs_
XnZbSIRywMaPAiAXhNQ"yelp4:////bi
z/Fs_XnZbSIRywMaPAiAXhNQ"yelp:
////biz/Fs_XnZbSIRywMaPAiAXhNQ"ht
tp://yelp.com/biz/Fs_XnZbSIRywMa
PAiAXhNQ.....+170323712042.ht
tp://www.districttaco.comR.....en-
```

The GeoHistory.mapsdata plist file contains individual history items but in the proprietary BLOB format. Each item may contain a start and finish route, possible Yelp information, and/or reverse geolocation data.

Each item contains a GUID in the first few bytes of the BLOB that can be used to potentially correlate to an on-disk file with a matching GUID, as shown in future slides.

## Maps: Searches and Directions [1]

- /Search, /GraphDirections, and /Directions directories
  - Contains GUID-named files
  - Data BLOBs
  - Appears empty on iOS backups

- Match GUID from GeoHistory.mapsdata [iOS 8+]

- Item 0 (most recent)
- Data BLOB
  - Contains GUID
  - “Current Location”
  - Where exactly?
  - (Next slide)

GUID	Content	Dictionary	Number	Value
45424233	704E5860	Root	Dictionary	(2 items)
350AF235	4, v_0A! 4, v_0A:15 05	MSPHistoryVersion	Number	1
C0280112	A i &jsC@ π @y0BS2(C	▼ MSPHistory	Array	(20 items)
6A734340	N L" Current Location*\$)i &jsC@	Item 0	Data	<08021224 30313631 41
1209EB13	1π @y0BS29i &jsC@π @y0BS2j i	Item 1	Data	<08031224 43354542 3
1340D8AF	&jsC@ π @y0BS2 i &jsC@ π @y0	Item 2	Data	<08021224 30313631 41
1A120936	BS2 i4 " • A /AEif#U AM 6		Data	<08031224 413433 3
53746174	L~'esC@ =00epBS2"X United Stat		Data	<08011224 39373946 4
22024443	es US District of Columbia" DC		Data	<08031224 41313545 3
696E6774	* District of Columbia2 Washingt		Data	<08011224 31334535 3
0F313134	on: 20036R 17th St NWZ 1145b 114		Data	<08031224 38373543 3
61706869	5 17th St NWr National Geographi		Data	<08011224 31413637 44
6E746F77	c Societyä Washingtonä Downtow		Data	<08011224 34423537 41
6D320F31	n* National Geographic Museum2 1		Data	<08011224 39453245 31
44432032	145 17th St NW2 Washington, DC 2		Data	<08021224 41314438 41
DACBE892	00362 United States ¶2 £2 A /AEi		Data	<08011224 35344646 31
765FBD41	f#U " z- C 0 ` A}03= v_0A		Data	<08021224 30333044 4
08181000	" z- C 0 ` A}03= v_0A"		Data	<08011224 35344646 31
D1022800	z- C 0 ` A}03= v_0A" z- C		Data	<08021224 30333044 4
013ABE01	0 ` A}03= v_0A"1 z- C 0 :æ		Data	<08011224 35344646 31
49586146	com.yelp HI35IBk0HRV28TQbIXaF		Data	<08021224 30333044 4

The GUID in each GeoHistory.mapsdata item can be matched to files in the /Search, /GraphDirections, or /Direction directories to find additional related information.

These files do not appear to get backed up in iTunes backups.



## Maps: Searches and Directions [2]

```

45424233 SED8D1194-EFCE-4BFF-9C3E-EBB3
350AF235 704E5860 4, v_QA! 4, v_QA:15 Ú5
C0280112 A i &jsC@ π @y0BSzC
6A734340 N L" Current Location"i &jsC@
1209EB13 1π @y0BSz9i &jsC@Am @y0BSzJ i
1340D8AF &jsC@ π @y0BSz i &jsC@ π @y0
1A120936 BSz i4 ™ • A./ÆEifú A. 6
53746174 L'esC@ =00epBSzA United States
22024443 es US District of Columbia" DC
696E6774 * District of Columbia2 Washingto
0F313134 on: 20036R 17th St NW2 1145b 114
61706869 5 17th St NWr National Geographi
6E746F77 c Societyã Washingtonã Downtow
60320F31 n* National Geographic Museum2 1
44432032 145 17th St NW2 Washington, DC 2
DACBE892 00362 United States 12 £2 A./ÆEi
765FBD41 fú " z- ( 0 ` A}Ü3= v_QA
08181000 " z- ( 0 ` A}Ü3= v_QA"
D1022800 z- ( 0 ` A}Ü3= v_QA" z- (
013ABE01 0 ` A}Ü3= v_QA"i z- ( 0 :æ
49586146 com.yelp HI35IBk0HRV28TQbIXaf
    
```

```

▼ ReportAProblem
  ► Directions
  ▼ GraphDirections
    030DEE89-EE79-48B0-9A99-D670CC1F1687
    0161A5E2-CBA7-456F-A90D-78B2754A69D2
    368C78A9-8599-4034-8E25-1D4829742FD3
    A1D8A3FD-54DB-40B1-BFB4-81F1FEE8816D
    ED8D1194-EFCE-4BFF-9C3E-EBB3704E5860
  ▼ Search
    1A67D286-9E36-4BB0-9022-BB9BC62D6B3C
    4B57B2EA-2827-45C2-A064-6A7631E767C4
    9E2E962F-B3C6-44D5-90DE-6EEE31F2A7D4
    13E5937D-FBFB-460F-932E-E79907992A28
    
```

```

en-US ° Z 1835 I St NWZ Washington, DC 20006Z United Sta
tes20 United States US District of Columbia" DC2 Washington:
20006B DowntownR I St NWZ 1835 I St NWj 5402ã Downtowñ
20006-5402 E en-US < \tn=address\1835 \tn=normal\ /+'a&I_'stR
+it_nOR+T.'wEst /+ ð en-US é /+'ju.na&I.r6Id_'ste&Its /+
/+'ju.na&I.r6Id_'ste&Its /+ District of Columbia" District of C
olumbia2 /+'wA.SInK.tSn /+B DowntowR /+'a&I_'stR+it_nOR+T.'wE
st /+Z \tn=address\1835 \tn=normal\b< \tn=address\1835 \tn=norm
al\ /+'a&I_'stR+it_nOR+T.'wEst /+ã Downtow J appleJ revgeoJ
USP ` "# z- ( 0 J appleJ revgeoJ USP ` "# z- ( 0 J apple
J revgeoJ USP ` "# z- ( 0 J appleJ revgeoJ USP ` (-; z * en-
US* zh-Hant-US* zh-Hans-US2 US: en-USb
    
```

**SANS** **DFIR**

FOR518.4 | Mac and iOS Forensic Analysis and Incident Response 173

These GUID files can contain route data in the /Directions and /GraphDirections directories and possibly Yelp and other search data in the /Search directory.

In the example above, this shows a map route from “Current Location” to “National Geographic Museum” in Washington, DC. Toward the end of this file, there will be reverse geolocation data (look for “revgeo” strings) that may be able to show you the start location of “Current Location”. In the example, this is “1835 I Street” in “Washington, DC”.

```

▼ ReportAProblem
  ► Directions
  ▼ GraphDirections
    030DEE89-EE79-48B0-9A99-D670CC1F1687
    0161A5E2-CBA7-456F-A90D-78B2754A69D2
    368C78A9-8599-4034-8E25-1D4829742FD3
    A1D8A3FD-54DB-40B1-BFB4-81F1FEE8816D
    ED8D1194-EFCE-4BFF-9C3E-EBB3704E5860
  ▼ Search
    1A67D286-9E36-4BB0-9022-BB9BC62D6B3C
    4B57B2EA-2827-45C2-A064-6A7631E767C4
    9E2E962F-B3C6-44D5-90DE-6EEE31F2A7D4
    13E5937D-FBFB-460F-932E-E79907992A28
    54FE98A4-2355-458F-A443-F9EF1BD0DC31
    98FC3AAD-E9E0-4AFA-B117-C2A57258116E
    979FF99C-4491-474C-8807-E7EC9AFC0057
    39016100-FA7E-4649-A0D9-97018D852B94
    A70031DB-C812-402D-B8BA-32838F239731
    E16A09BD-0A20-41CB-AA31-6C1C9EB055FA
    
```

## Decoding Map Protobuf BLOBs [1]

▼ 458893F1-3FOA-43DA-8755-3D54AF8EFF46	Dictionary	(3 items)
contentsTimestamp	Data	<04aa2eab 2b164879 b07a08e7 a74d25ee 00000004>
modificationDate	Date	Dec 14, 2018 at 6:07:57 PM
contents	Data	<08021224 34353838 39334631 2d334630 412d3433 4441

```

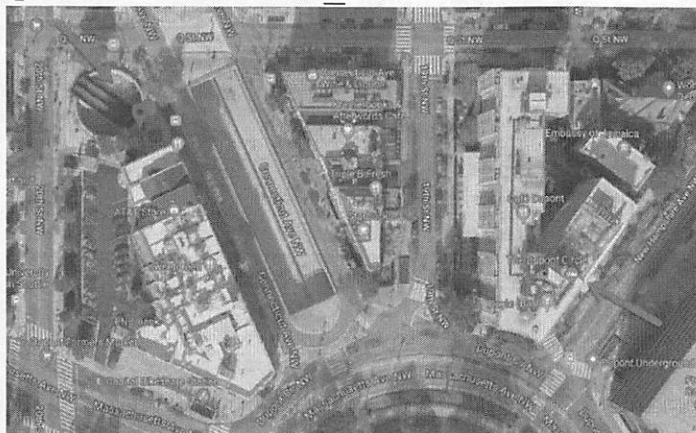
44412D38 3735352D 33443534 $458893F1-3FOA-43DA-8755-3D54
0AB96C0A E3190A91 01080422 AF8EFF46 c..7..A:.l .l . . "
1AD94253 C0A00601 100119E5 . . . .-...tC@ .. .BS. . .
00000000 00084039 00000020 +.7..A ) H@1 @9
00F0BF9A 0651AE52 6ADB8AFB .6.7I . . . .-...tC@ .. .Q.Rj
4340111F A8D71AD9 4253C0A0 ?. . . .-...tC@ .. .Q.Rj
1000225B 08021000 20C0D102 . . . .-...tC@ .. .Q.Rj
1FA8D71A D94253C0 22120A10 ( 0 B, * ..-...tC@ .. .BS."
6170706C 654A0672 65766765 America/New_York0 J appleJ revge
A9010801 100020C0 D1022800 oJ USP ` .}.p..7..A" . . . (
35323820 436F6E6E 65637469 ( 0 B) ' 9R en-US 1528 Connecti
4A067265 7667656F 4A025553 cut Ave NW. J appleJ revgeoJ US
68414844 6D4E7662 53356863 P Z0 MplaceRequest=ChAKDmNvb55hc
41456946 416F5343 6263747A HBsZSSNYXBz0ARCGiYGAeiFAoSbctz
02817D88 708BA437 E2C04122 byWdENAER+oixrZQ1PA' }.p..7..A"
0ADE010A 05656E2D 55531AD4 . . . ( 0 B, * . . en-US .
41766520 4E575A15 57617368 Z 1528 Connecticut Ave NWZ Wash
6E697465 64205374 61746573 ington, DC 20036Z United States
55531A14 44697374 72696374 z. United States US District
7368696E 67746F6E 3A053230 of Columbia" DC2 Washington: 20
436F6E6E 65637469 63757420 036B DuPont CircleR Connecticut
6F6E6E65 63746963 75742041 Ave NWZ 1528b 1528 Connecticut A
6512630A 05656E2D 55531A5A ve NW. DuPont Circle c en-US Z
5C746E3D 6E6F726D 616C5C20 \tn=address\ 1528 \tn=normal\
    
```

The protobuf BLOBs may be embedded in plist files or just files themselves. They look like the example above; some may be more data, some may be less.

The first step is to extract the BLOB (if it needs to be extracted). Save this data to a file.

## Decoding Map Protobuf BLOBs [2]

- Use 'protoc' utility to decode
  - brew install protobuf
  - protoc -decode\_raw < [BLOB]



```
Sarahs-Air:Edits ompa$ protoc --decode_raw < mapblob.txt
1: 2
2: *458993f1-3f8a-430a-8755-3d54af8eff46*
3: 0x41c0e237a4cd313
7 {
  1 {
    1 {
      1: 4
      4 {
        1 {
          1: 0x48437496bccd2db7
          2: 0xc05342d91ad7a81f
          100: 1
        }
        2: 1
        3: 0x41c0e237a42b09e5
        4: 31
        5: 0x4040000000000001
        6: 0x4000000000000000
        7: 0x3fa03c0a20000000
        8: 0xbff0000000000000
        9: 0xbff0000000000000
        100: 0xbff0000000000000
        101: 0x3ffbbadba52ae51
        102: 0
        103: 0
        105: 1
        106 {
          1: 0x48437496bccd2db7
          2: 0xc05342d91ad7a81f
          100: 1
        }
        107: 0xbff0000000000000
      }
    }
  }
  5: 1
}
2 {
  1 {
    2: 0
    4 {
```

Starting Lat/Long

Timestamp

Protobuf data can be decoded by using the protoc utility from Google's Protobuf utilities.

Shown on the right is a partial example of the JSON-like output. There are many pieces of data stored, including originating coordinates (Dupont Metro in DC) and the location search timestamp. The output may be quite verbose; it is advisable to save this output to another file for analysis.

The coordinates were extracted and changed to decimal format and mapped. The Pin on the left shows where the search was originally started from to find directions to the Dupont Circle Hotel.

## Section 4: Agenda

Part 1: Application Fundamentals

Part 7: Contacts

Part 2: Introduction to SQLite Queries

Part 8: Notes

Part 3: Safari Browser

Part 9: Apple Pay, Wallet, Passes

Part 4: Apple Mail

Part 10: Photos

Part 5: Communication

Part 11: Maps

Part 6: Calendar and Reminders

Part 12: Apple Watch

This page intentionally left blank.



---

## Section 4: Part 12

### Apple Watch

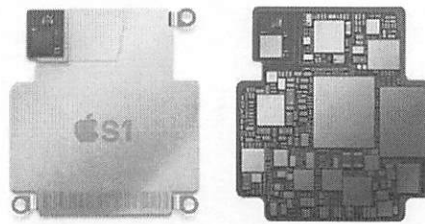
---

This page intentionally left blank.



## Apple Watch

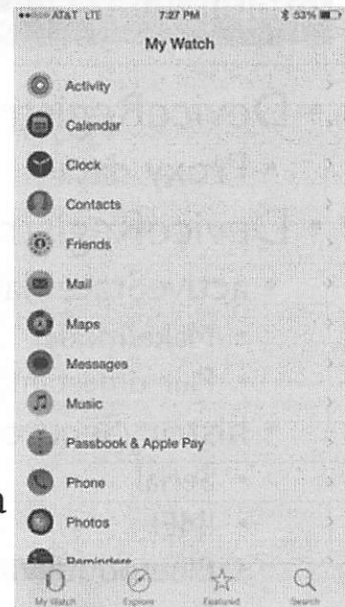
- Apple's wearable technology
  - "Smart Watch"
- Must be paired with iPhone (iOS 8.2, iPhone 5, or newer)
- Wi-Fi, GPS, NFC, Bluetooth, cellular
- "System in Package"
- Proxy device



The Apple Watch is Apple's new wearable technology, or "Smart Watch". It comes in two sizes and must be paired with an iPhone (not an iPod or iPad).

## Software

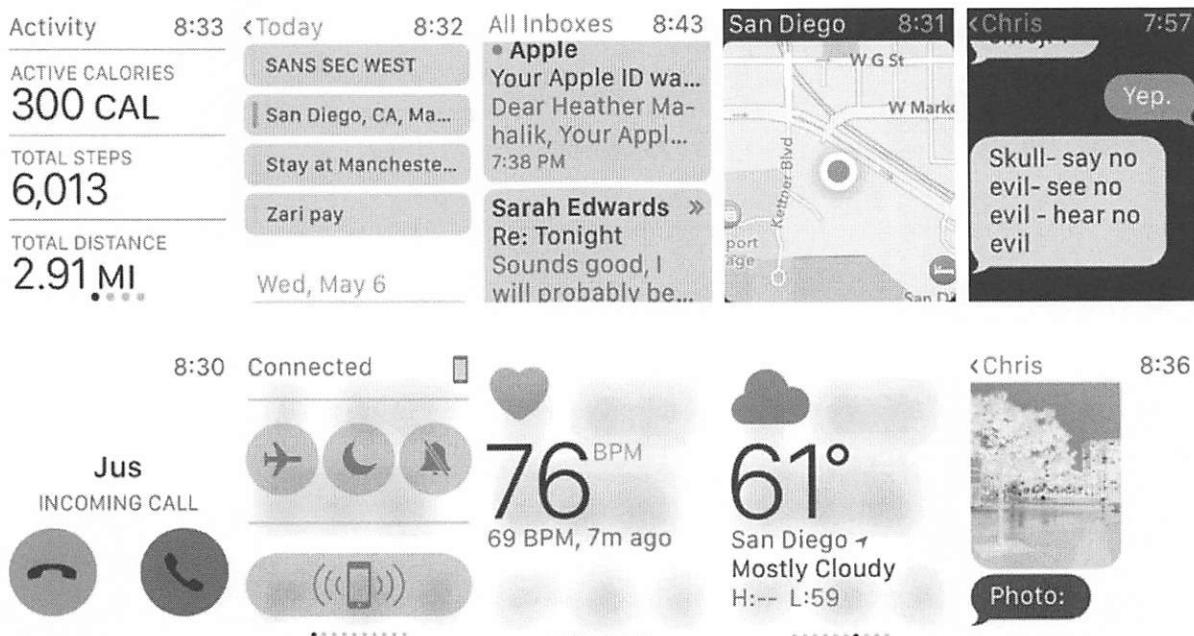
- Software
  - WatchOS: iOS-Based
  - Carousel: The Interface
    - i.e., Springboard/Finder
- Apps:
  - Native: Activity, Calendar, Clock, Contacts, Friends, Mail, Maps, Messages, Music, Passbook/Apple Pay, Phone, Photos, Reminders, Stocks, Weather, Workout
  - Third-party Apps!
- Use the Apple Watch App on iPhone to sync data



179

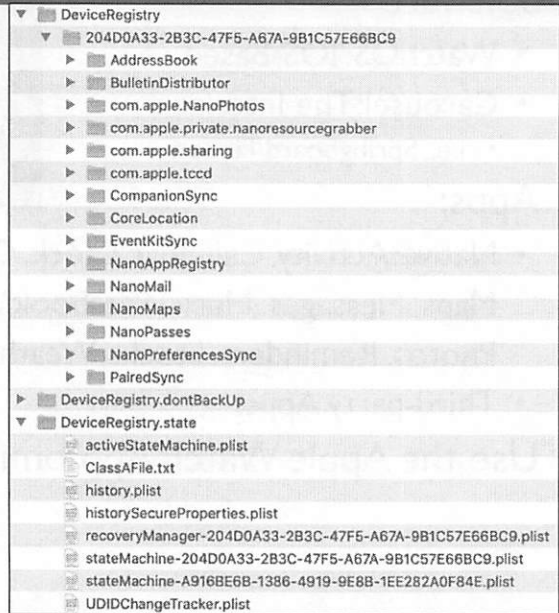
The Apple Watch runs WatchOS. Research shows this may be a slightly modified version of iOS.

Applications on the watch include many of those on the iPhone. The Watch can be thought of as an extension of the iPhone.



## Apple Watch: Data Directory /mobile/Library/DeviceRegistry/<GUID>

- DeviceRegistry
  - Proxy data
- DeviceRegistry.state/
  - activeStateMachine.plist
    - Make/model
    - Pair timestamp
  - historySecureProperties.plist
    - Serial
    - IMEI
    - Bluetooth/Wi-Fi MAC addresses
    - Etc.



180

Most of the data associated with the Apple Watch in a backup is stored in the /mobile/Library/DeviceRegistry/ directory. This data is mostly redundant data found on the device itself.

Identifying information can be found in the DeviceRegistry.state/ directory in a few plist files.

## Apple Watch: PassBook/Wallet /mobile/Library/DeviceRegistry/<GUID>/NanoPasses/ nanopasses.sqlite3

Table:  pass

	unique_id	type_id	encoded_pass
Filter	Filter		Filter
1	zqhdx--65l9...	pass.united.UnitedMobileBoardingPass	(BLOB)
2	HlmS+zJwkr...	pass.united.UnitedMobileBoardingPass	(BLOB)
3	tiS0+4d-CLZjeVfyza4...	pass.united.UnitedMobileBoardingPass	(BLOB)
4	egg2jjshjPfrg...	pass.united.UnitedMobileBoardingPass	(BLOB)
5	YpGV2LFtK...	pass.united.UnitedMobileBoardingPass	(BLOB)

Contains  
Pass Data!



181

The Apple Watch can be used to board flights using the Passbook application. This information is stored in its own SQLite database, `nanopasses.sqlite3`.

---

## Lab 4.3

# Applications: Part II

---

This page intentionally left blank.





# FOR518 Section 4: Application Data Analysis

© 2020 Sarah Edwards | All Rights Reserved | Version F01\_01

Author: Sarah Edwards  
oompa@csh.rit.edu  
mac4n6.com  
<http://twitter.com/iamevltwin>

<https://digital-forensics.sans.org/>  
<http://twitter.com/sansforensics>

*"As usual, SANS courses pay for themselves by Day 2. By Day 3, you are itching to get back to the office to use what you've learned."*

Ken Evans, Hewlett Packard Enterprise - Digital Investigation Services

**SANS Programs**  
[sans.org/programs](http://sans.org/programs)

GIAC Certifications  
Graduate Degree Programs  
NetWars & CyberCity Ranges  
Cyber Guardian  
Security Awareness Training  
CyberTalent Management  
Group/Enterprise Purchase Arrangements  
DoDD 8140  
Community of Interest for NetSec  
Cybersecurity Innovation Awards



Search SANSInstitute

**SANS Free Resources**  
[sans.org/security-resources](http://sans.org/security-resources)

- E-Newsletters
  - NewsBites: Bi-weekly digest of top news
  - OUCH!: Monthly security awareness newsletter
  - @RISK: Weekly summary of threats & mitigations
- Internet Storm Center
- CIS Critical Security Controls
- Blogs
- Security Posters
- Webcasts
- InfoSec Reading Room
- Top 25 Software Errors
- Security Policies
- Intrusion Detection FAQ
- Tip of the Day
- 20 Coolest Careers
- Security Glossary

**SANS Institute**

8120 Woodmont Avenue | Suite 310  
Bethesda, MD 20814  
301.654.SANS(7267)  
[info@sans.org](mailto:info@sans.org)