SANS

www.sans.org

FORENSICS 518

MAC FORENSIC ANALYSIS

518.2

User Domain File Analysis

The right security training for your staff, at the right time, in the right location.

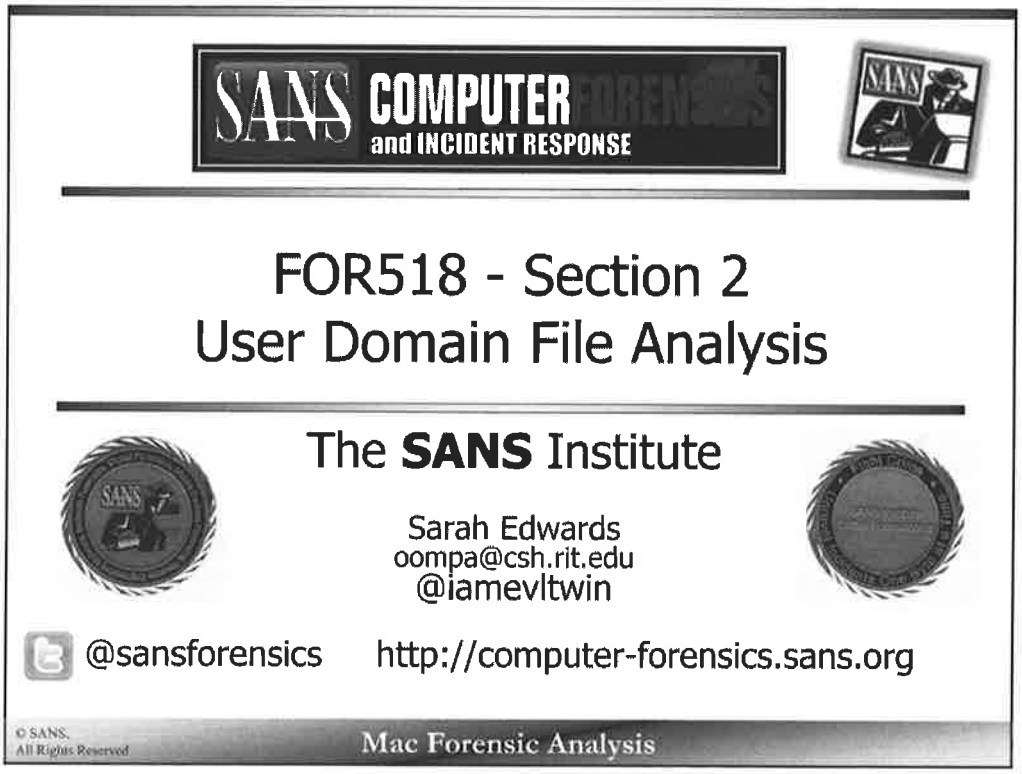
IMPORTANT-READ CAREFULLY:

This Courseware License Agreement ("CLA") is a legal agreement between you (either an individual or a single entity; henceforth User) and the SANS Institute for the personal, non-transferable use of this courseware. User agrees that the CLA is the complete and exclusive statement of agreement between The 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. 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. BY ACCEPTING THIS COURSEWARE YOU AGREE TO BE BOUND BY THE TERMS OF THIS CLA. IF YOU DO NOT AGREE YOU MAY RETURN IT TO THE SANS INSTITUTE FOR A FULL REFUND, IF APPLICABLE. The SANS Institute hereby grants User a non-exclusive license to use the material contained in this courseware subject to the terms of this agreement. User may not copy, reproduce, re-publish, distribute, display, modify or create derivative works based upon all or any portion of this publication in any medium whether printed, electronic or otherwise, for any purpose without the express written consent of the 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 the SANS Institute.

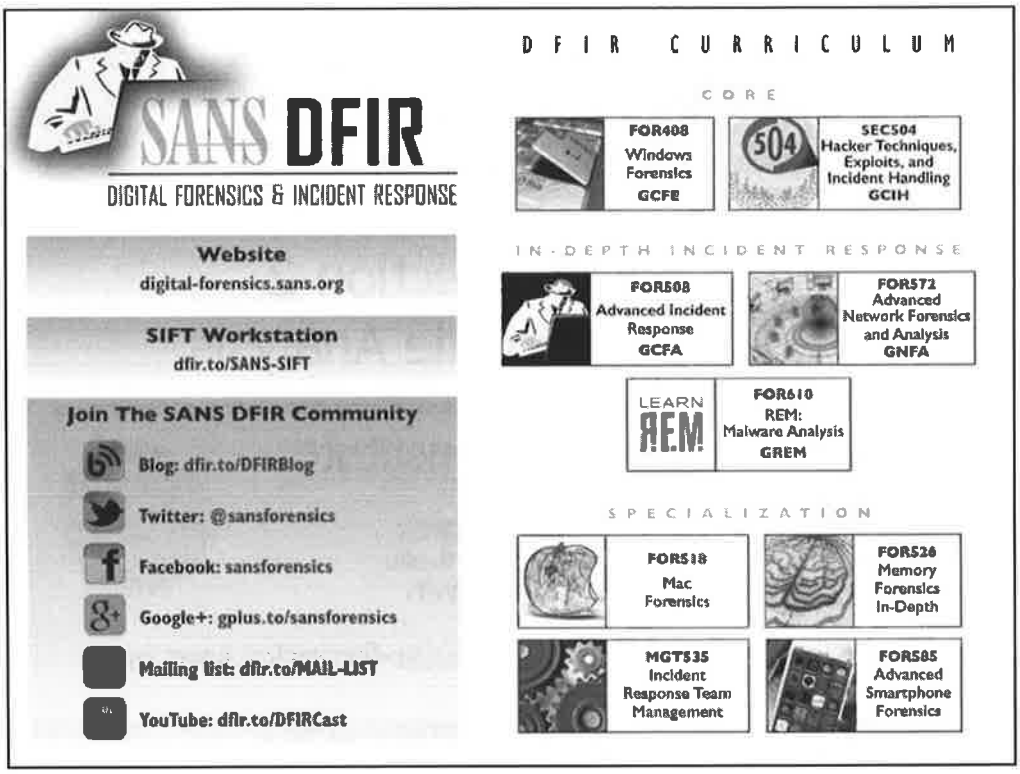
The SANS Institute reserves the right to terminate the above lease at any time. Upon termination of the lease, user is obligated to return all materials covered by the lease within a reasonable amount of time.

SANS acknowledges that any and all software and/or tools presented in this courseware are the sole property of their respective trademark/registered/copyright owners.

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.



Author: Sarah Edwards
oompa@csh.rit.edu
http://twitter.com/iamevltwin
http://twitter.com/sansforensics



This page intentionally left blank.

SANS DFIR

DIGITAL FORENSICS & INCIDENT RESPONSE

Website
digital-forensics.sans.org

SIFT Workstation
dfir.to/SANS-SIFT

Join The SANS DFIR Community

- Blog: dfir.to/DFIRBlog
- Twitter: @sansforensics
- Facebook: sansforensics
- Google+: gplus.to/sansforensics
- Mailing list: dfir.to/MAIL-LIST
- YouTube: dfir.to/DFIRCast

DFIR CURRICULUM

CORE

FOR408
Windows Forensics
GCFE

SEC504
Hacker Techniques, Exploits, and Incident Handling
GCIH

IN-DEPTH INCIDENT RESPONSE

FOR508
Advanced Incident Response
GCFA

FOR572
Advanced Network Forensics and Analysis
GNFA

LEARN REM

FOR610
REM: Malware Analysis
GREM

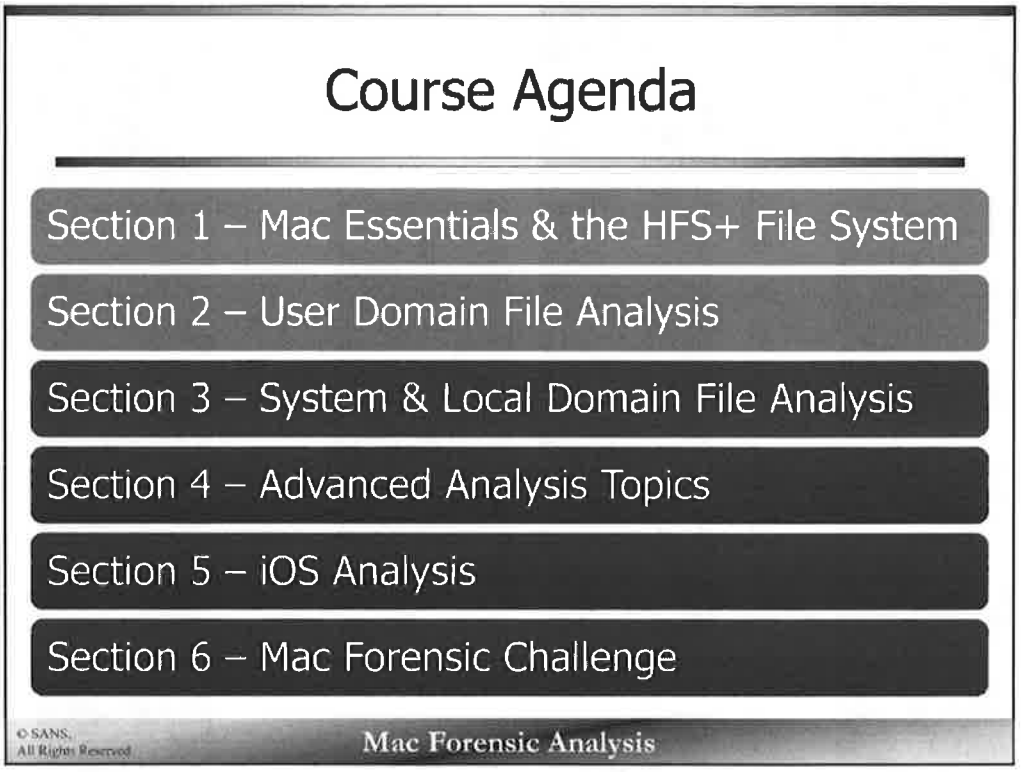
SPECIALIZATION

FOR518
Mac Forensics

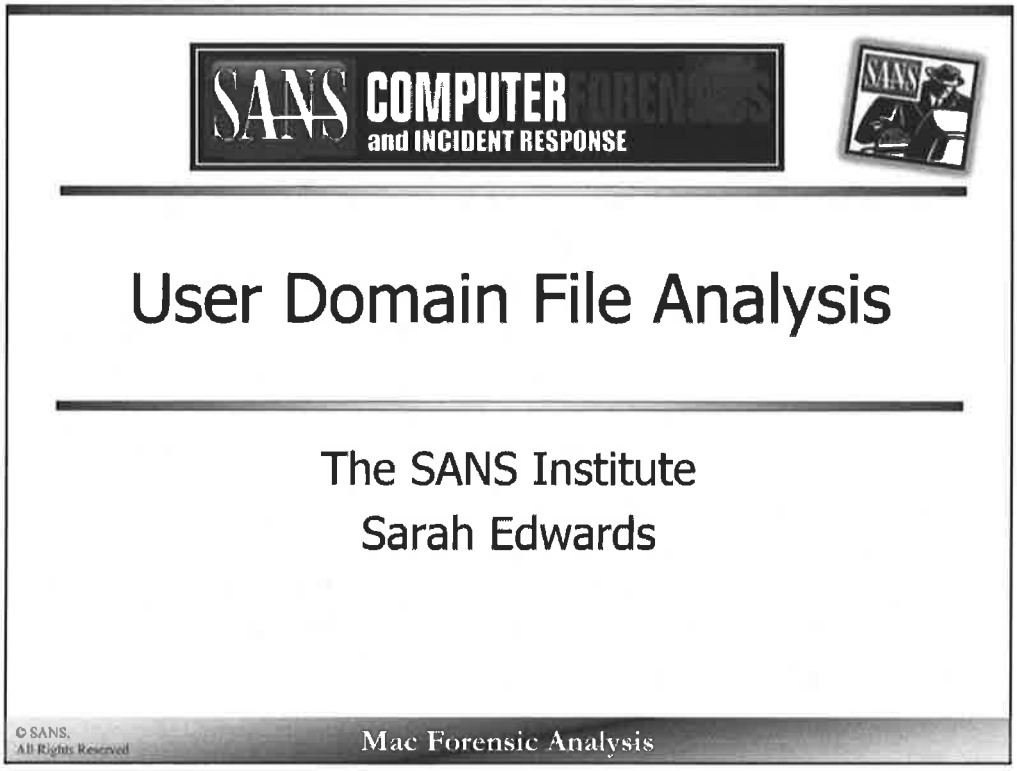
FOR526
Memory Forensics In-Depth

MGT535
Incident Response Team Management

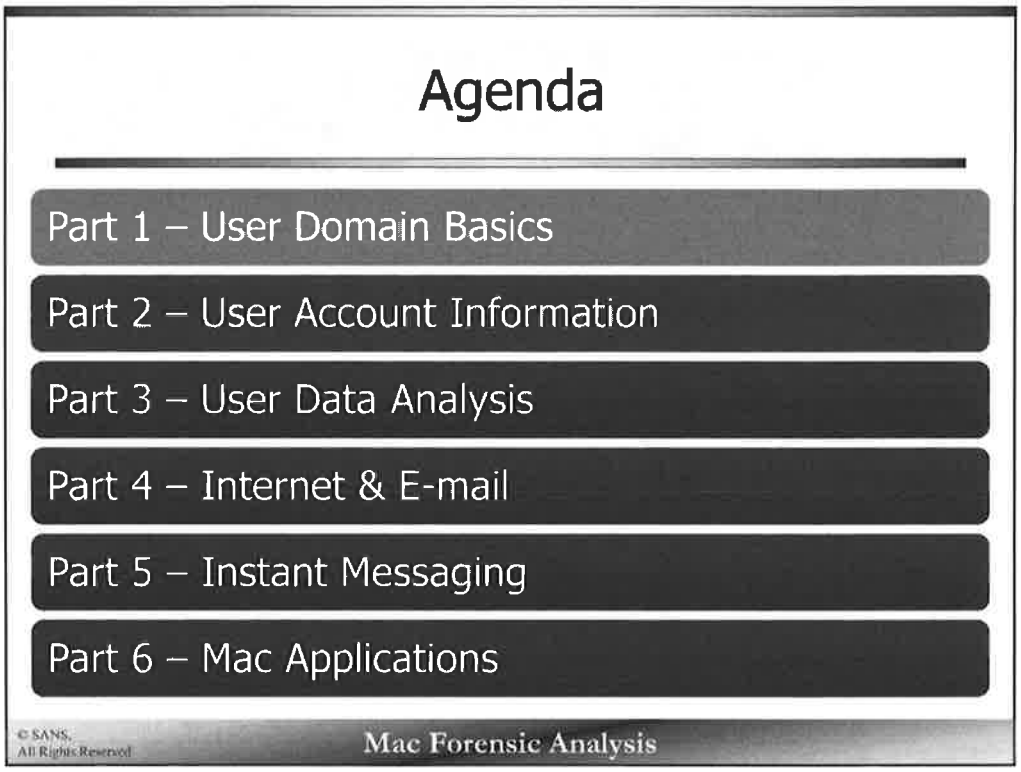
FOR585
Advanced Smartphone Forensics



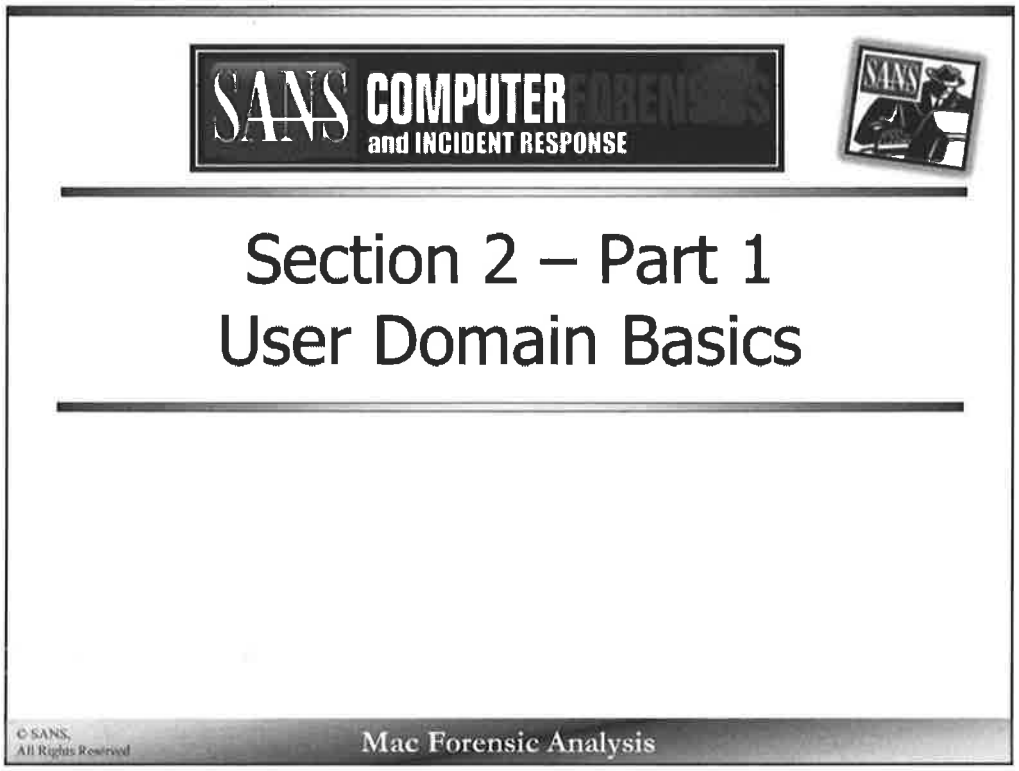
This page intentionally left blank.



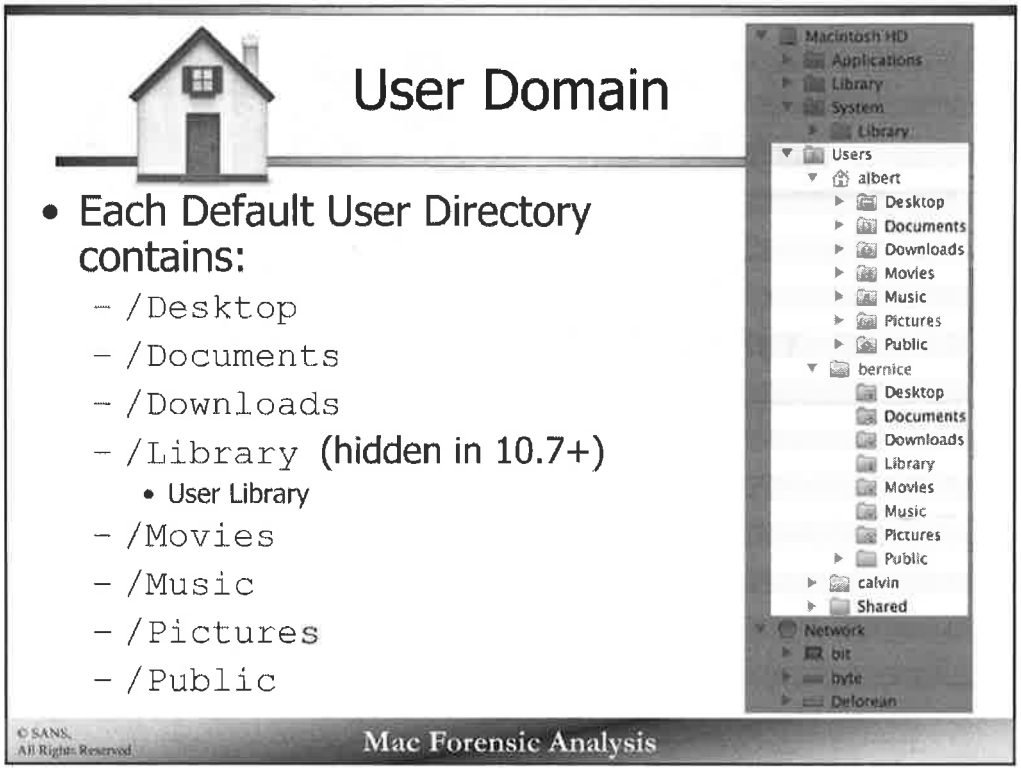
This page intentionally left blank.



This page intentionally left blank.



This page intentionally left blank.



The User Domain consists of the all the user files. Their documents, pictures, music, etc.

Each user has their own home directory in the `/Users/` folder, marked by a house icon in the screenshot above. If a user is logged on, default permissions will not allow them to view the files in another user's directory noted by the red circle with the line icon.

Most of the directories are self explanatory in terms of information they may contain. The public folder may contain data the user wants to share with other users. The user's Library directory contains app-specific data.

This domain may also contain "Sites" directory if web sharing is enabled; this would contain the user's personal web site.

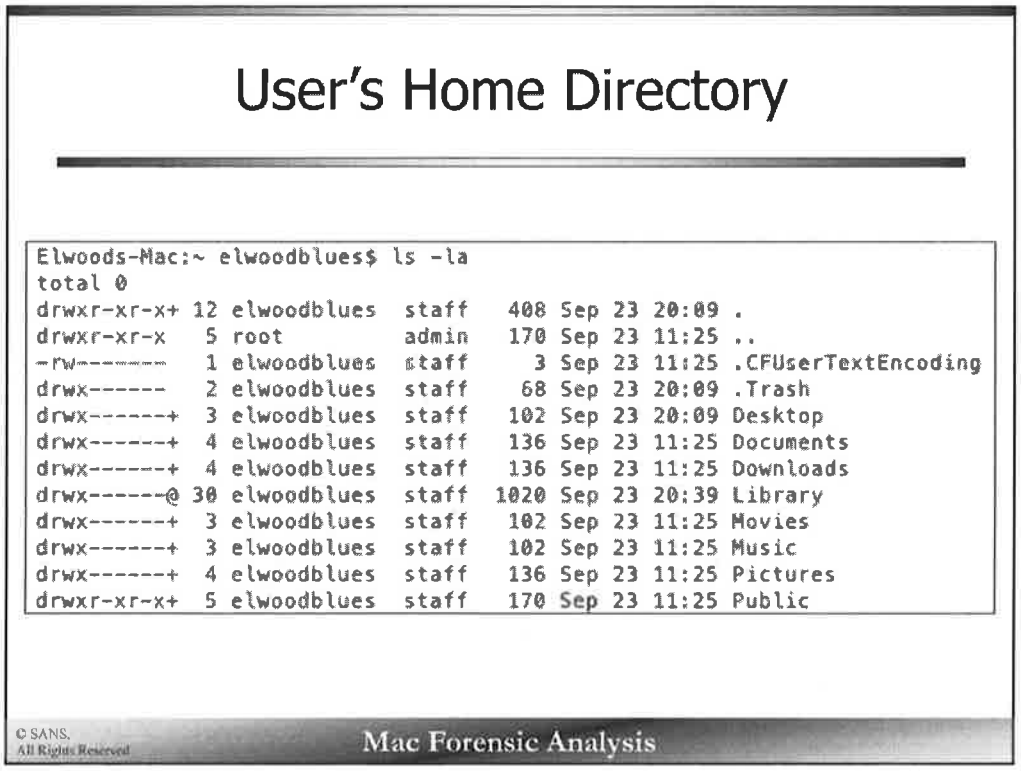
Starting with Mac OS X Lion, the user's `Library` folder was hidden. Some users may prefer to be able to view the files easily. The author finds that it helps forensic research to have the Library directory to be viewable. The command to permanently change this for a specific account is below. It may also be accessed by holding down the "option" key when accessing it via the 'Go' menu in the Finder Toolbar.

```
chflags nohidden /Users/<username>/Library
```

Reference:

File System Programming Guide – File System Basics

[https://developer.apple.com/library/mac/DOCUMENTATION/FileManagement/Conceptual/FileSystemProgrammingGUide/FileSystemOverview/FileSystemOverview.html]



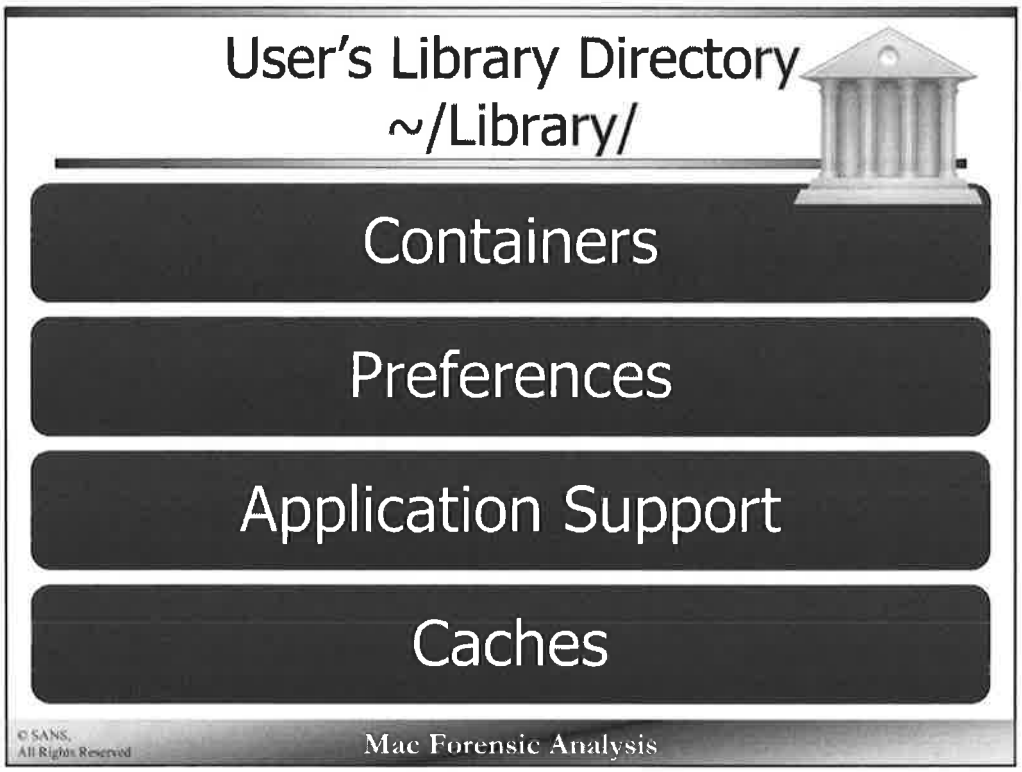
Similar to Windows systems, the user has folders for specific types of data (i.e., My Documents, My Music, My Pictures, etc.). The `Movies`, `Music`, and `Pictures` directories should (but not required) contain items related to their respective directories.

The `.Trash` contains the items the user has chosen to "delete", similar to the Windows Recycle Bin.

The `Downloads` directory is the default folder for downloads from web browsers and other applications. This directory may contain years worth of user downloads!

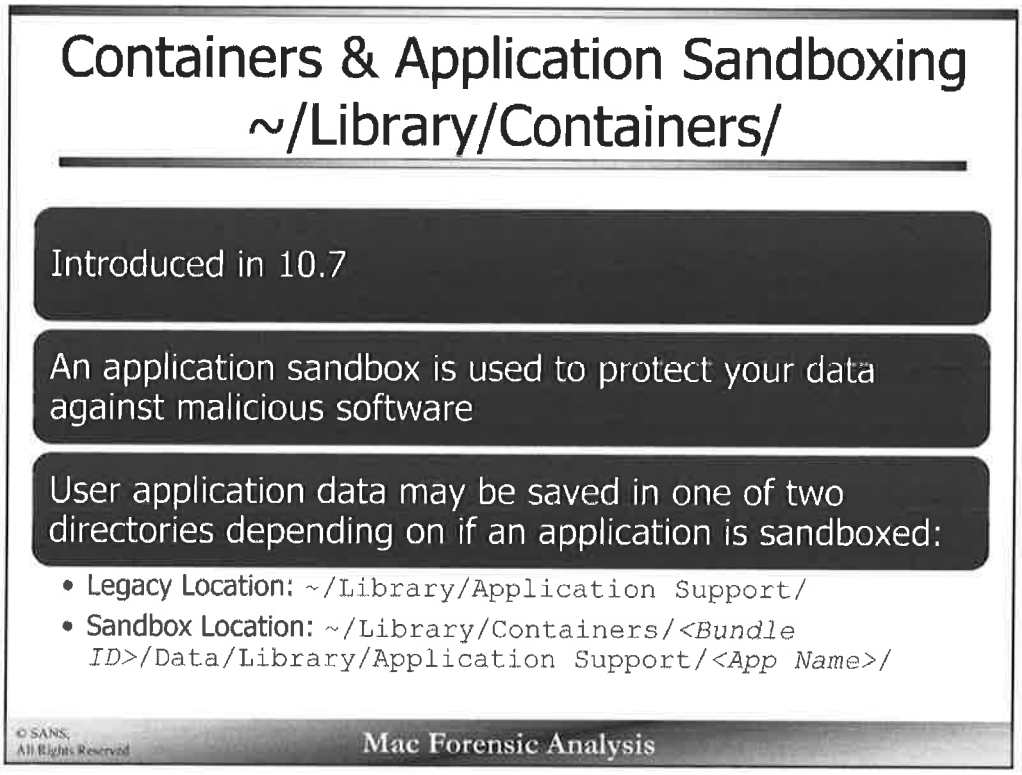
The `Public` directory is used for items the user has chosen to share with other users.

The `Library` [User Library] contains many items related to the specific user, such as preferences and application data. This `Library` directory is different from the System Library and Local Library directories. These files are specific to a user account.



The user's `Library` directory contains many sub-directories of interest. The few listed here will be mentioned over and over in this section. A forensic analyst may find lots of good forensic tidbits in this directory that they can tie to a specific user account. Each user account will have their own user Library directory.

It is worth noting here the tilde '~' is used as a terminal shortcut to the current user's home directory.



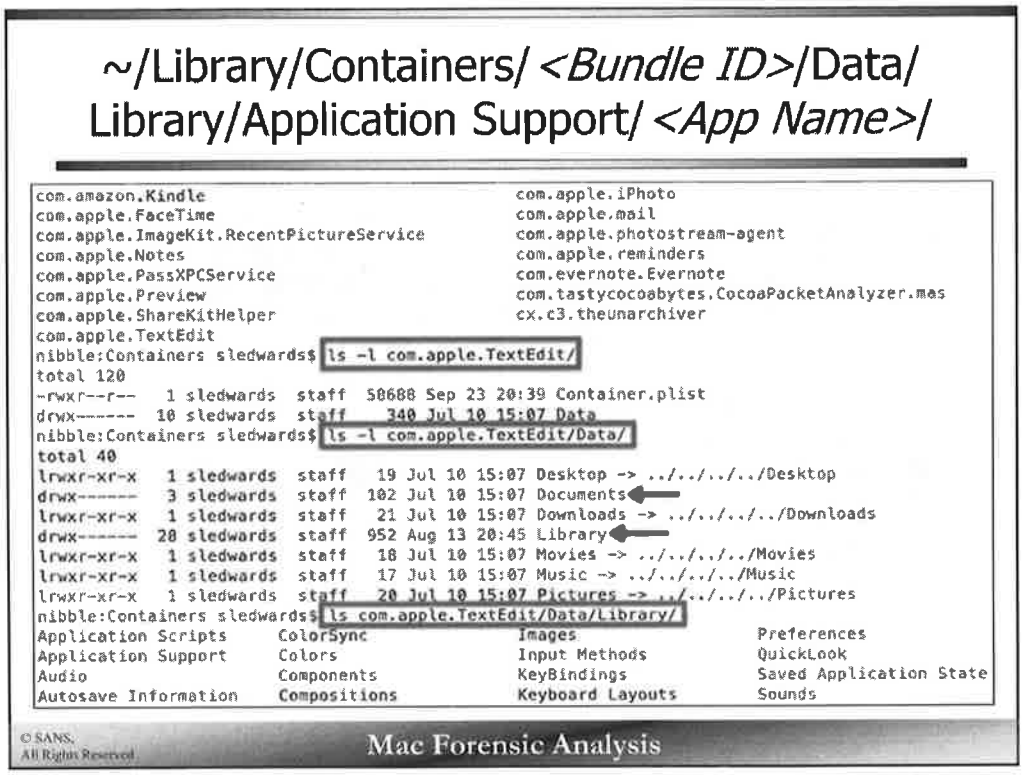
The `Containers` directory, introduced in 10.7, contains data that is sandboxed. This data will be similar to that found in `~/Library/Application Support/` directory for those applications that do not implement sandboxing.

References:

Apple Developer Documentation – App Sandbox Design Guide

http://developer.apple.com/library/mac/#documentation/Security/Conceptual/AppSandboxDesignGuide/AboutAppSandbox/AboutAppSandbox.html#//apple_ref/doc/uid/TP40011183-CH1-SW1

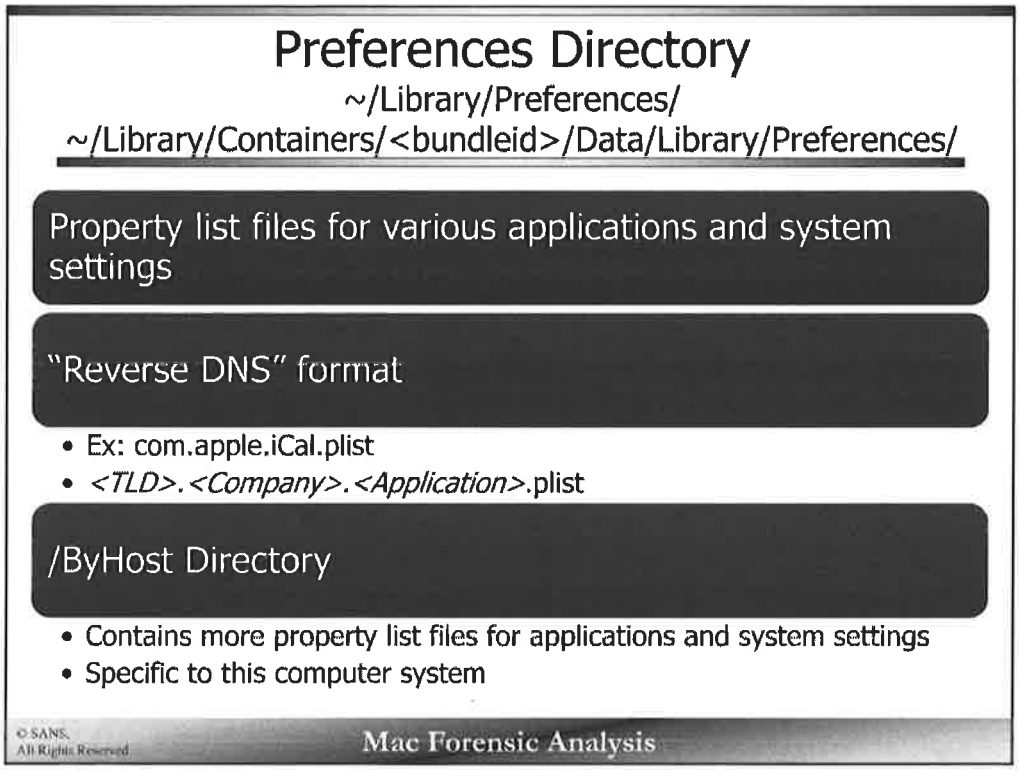
http://developer.apple.com/library/mac/#documentation/Security/Conceptual/AppSandboxDesignGuide/MigratingALegacyApp/MigratingAnAppToASandbox.html



Each "Container" in the `Containers` directory is named in the reverse DNS format. Each directory contains a `Container.plist` file and a Data directory.

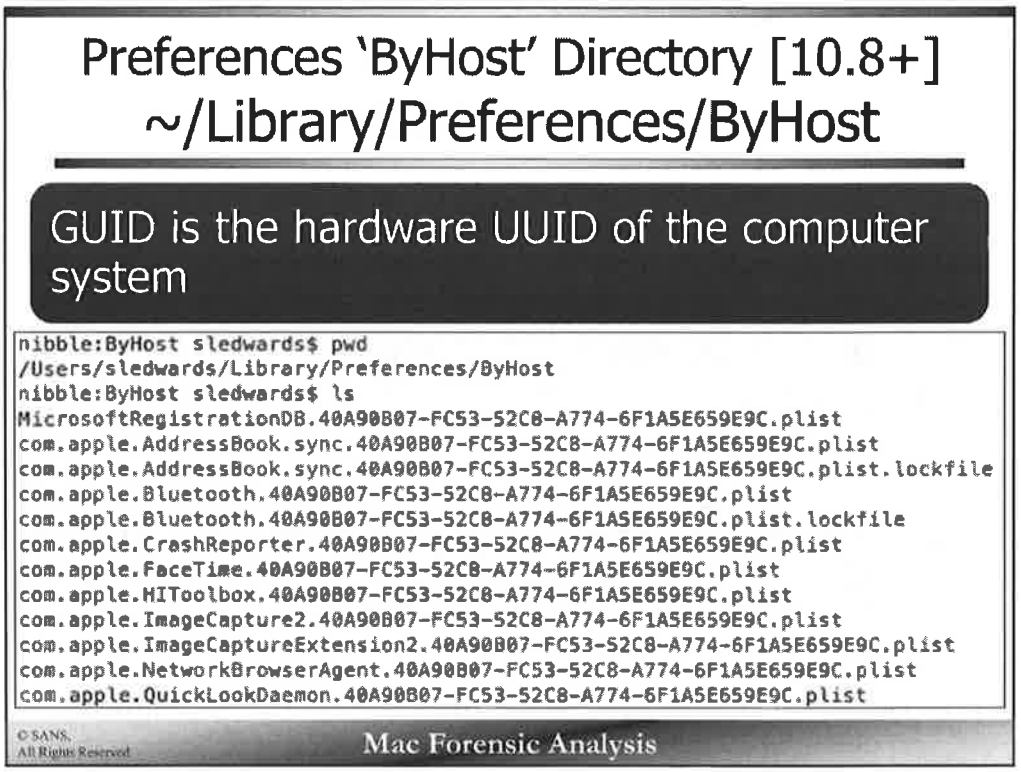
The `Container.plist` file contains information about the sandbox application.

The `Data` directory contains a similar layout to the user folder with symbolic links to various directories. The important directories here are those that are **not** links. In the screenshot above, the `Documents` and `Library` directories contain the data of interest. While the nested directories may also contain linked data, the sandboxed data will not be linked.



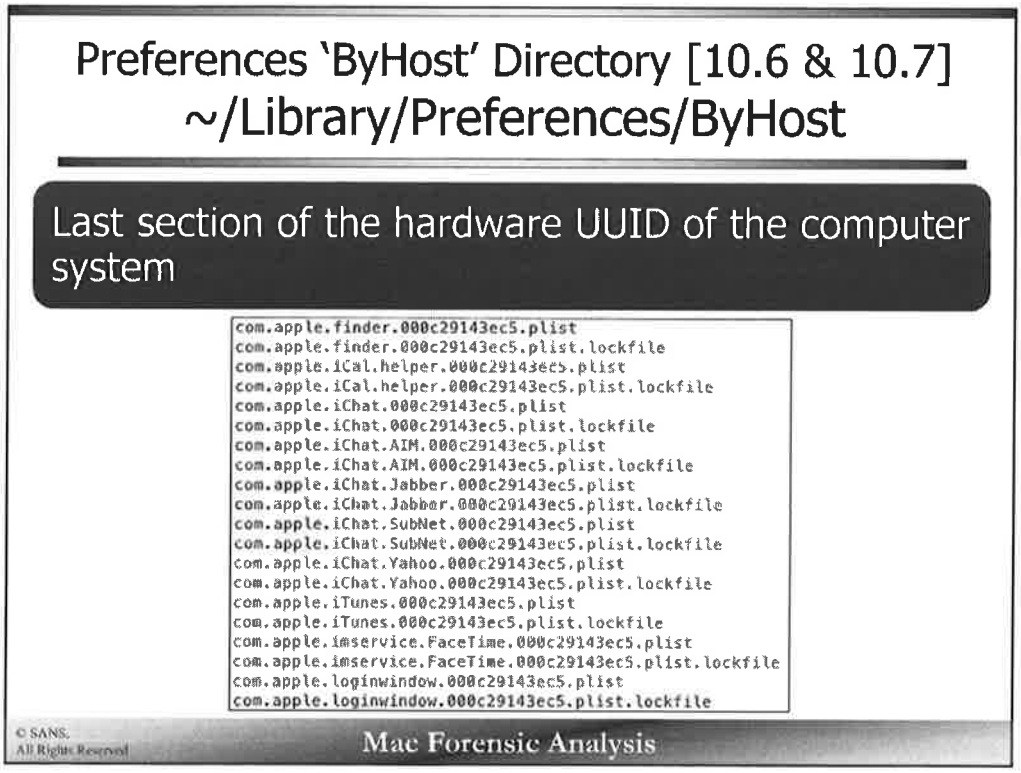
The User `Library` contains the user's preferences, usually in the form of a Property List file (`.plist`), in what is sometimes called "reverse DNS format". Reverse DNS format starts with the top level domain of the company such as ".com" or ".org", followed by the company name (i.e.,: "apple" or "microsoft"), followed by the name of the application with a ".plist" file extension.

The `ByHost` subdirectory in the `Preferences` directory contains more preference property list files that are specific to the computer system.

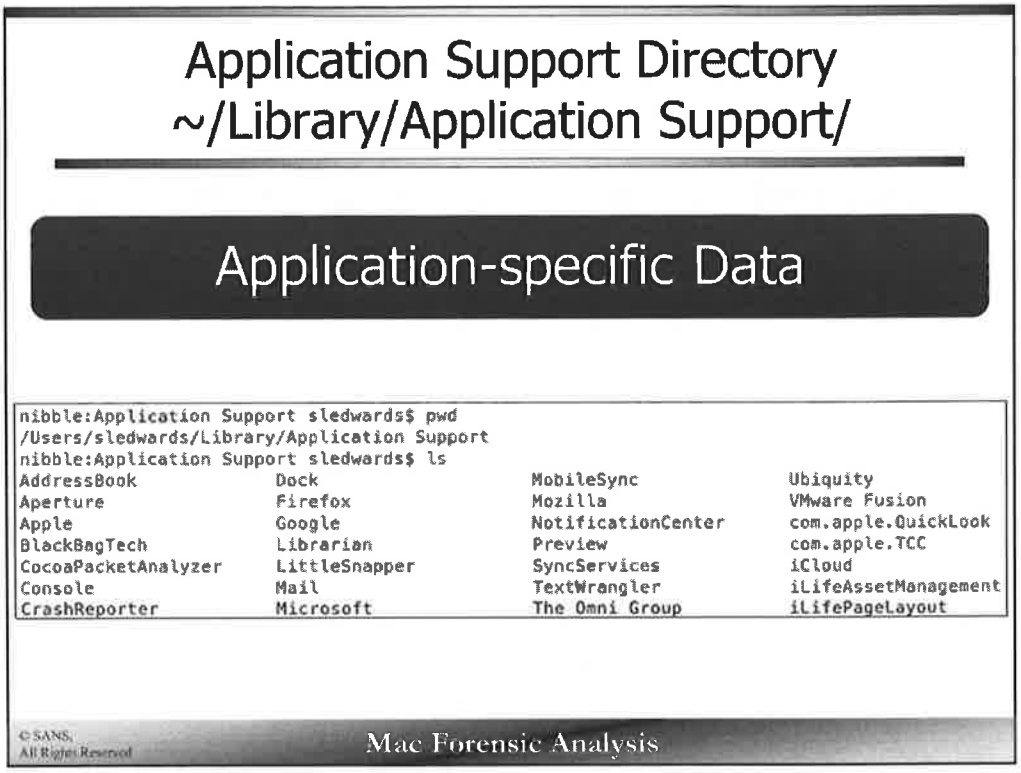


The `ByHost` directory contains preferences files with an additional GUID at the end of the filename. On 10.8+ this GUID is the hardware UUID of the system.

The hardware UUID can be found on a live system by using the System Information application and viewing the top-level of the "Hardware" section.

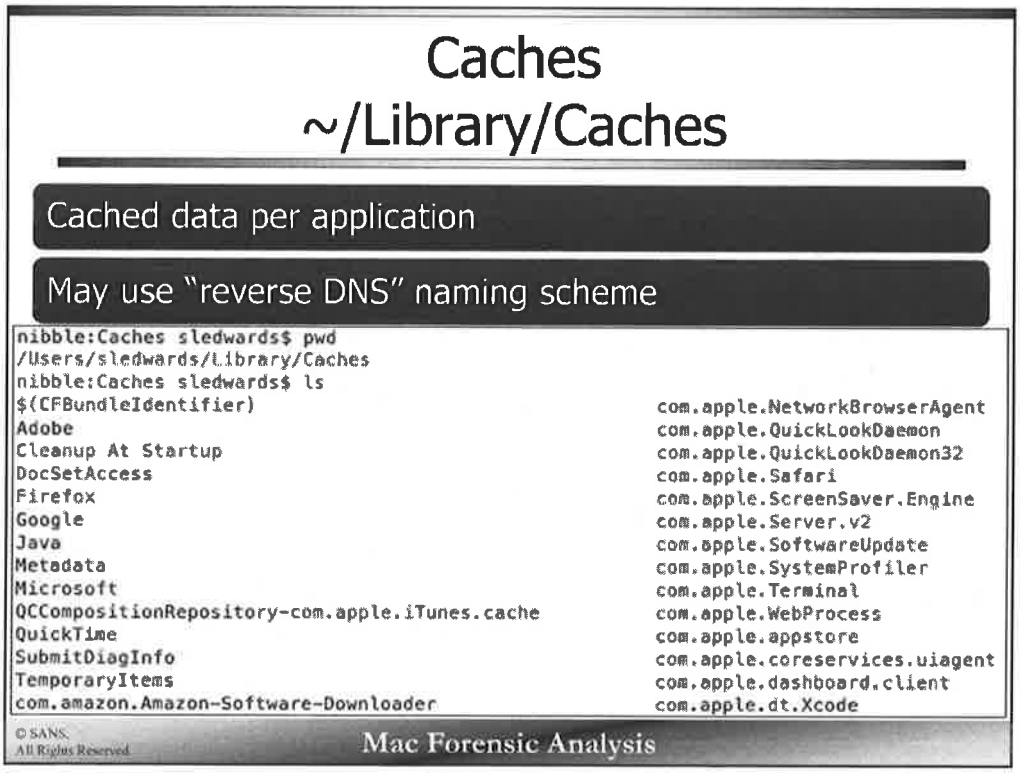


Slightly different from 10.8+ systems, on 10.6 and 10.7 the preference files in the ByHost directory contain only the MAC address of the system.

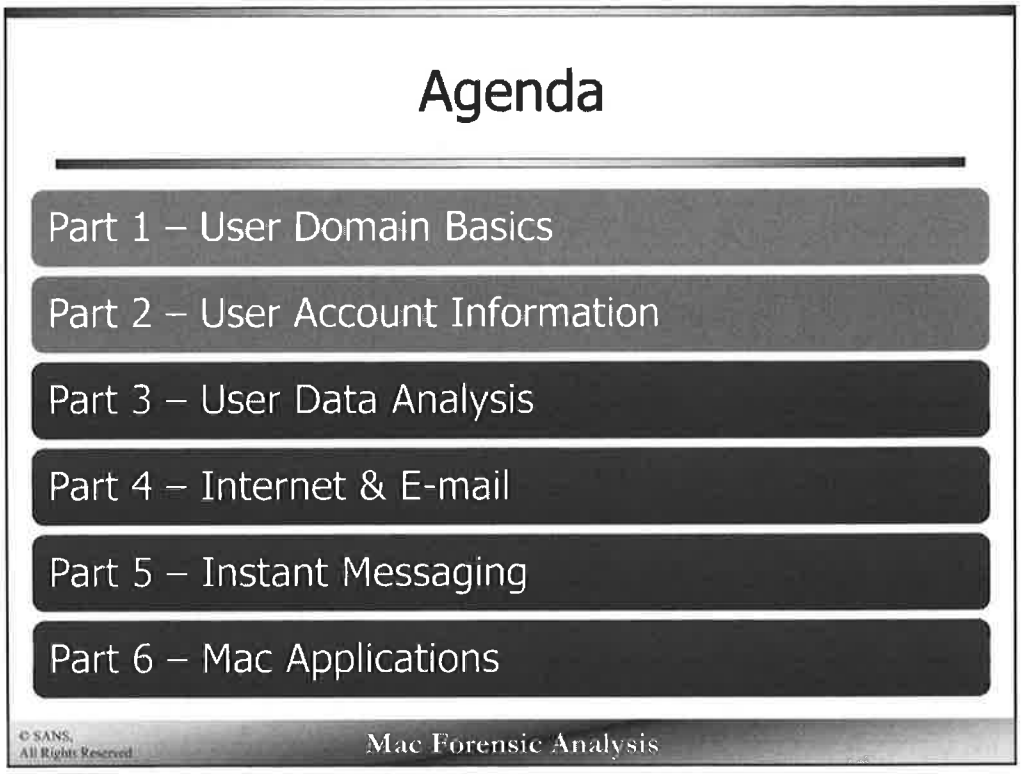


The `Application Support` directory contains data specific to various applications. This is similar to the `AppData` directory on Windows systems.

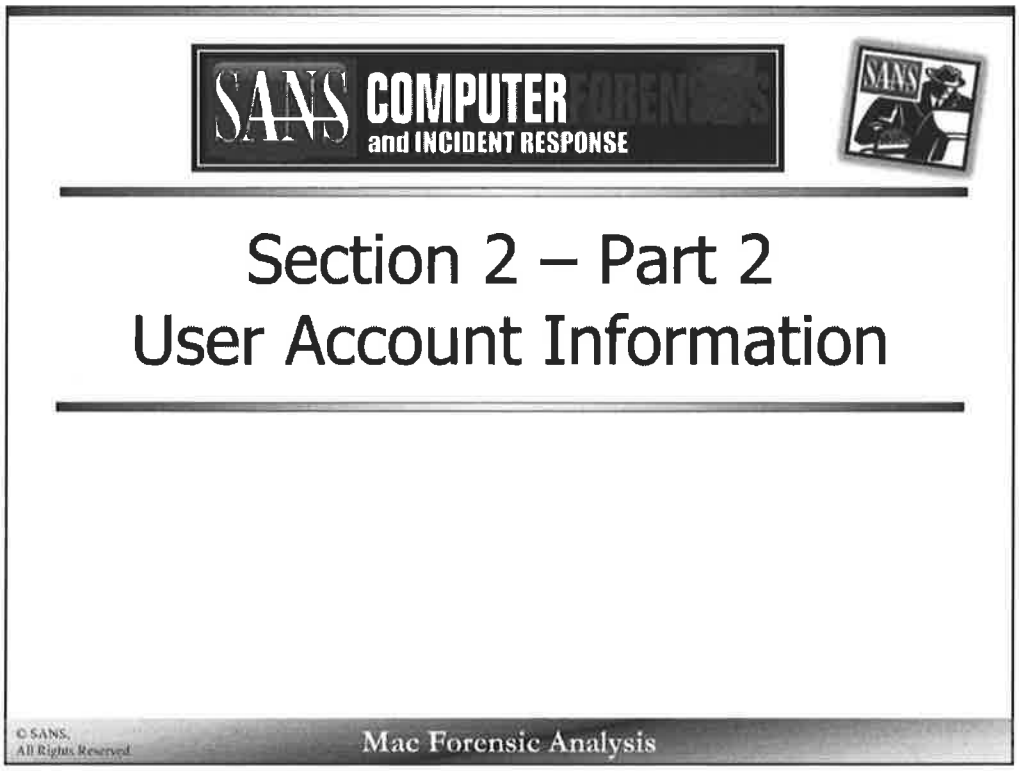
Each application directory may contain databases, property list files, or other proprietary data files. The method in which each application stores its data is up to the developer.



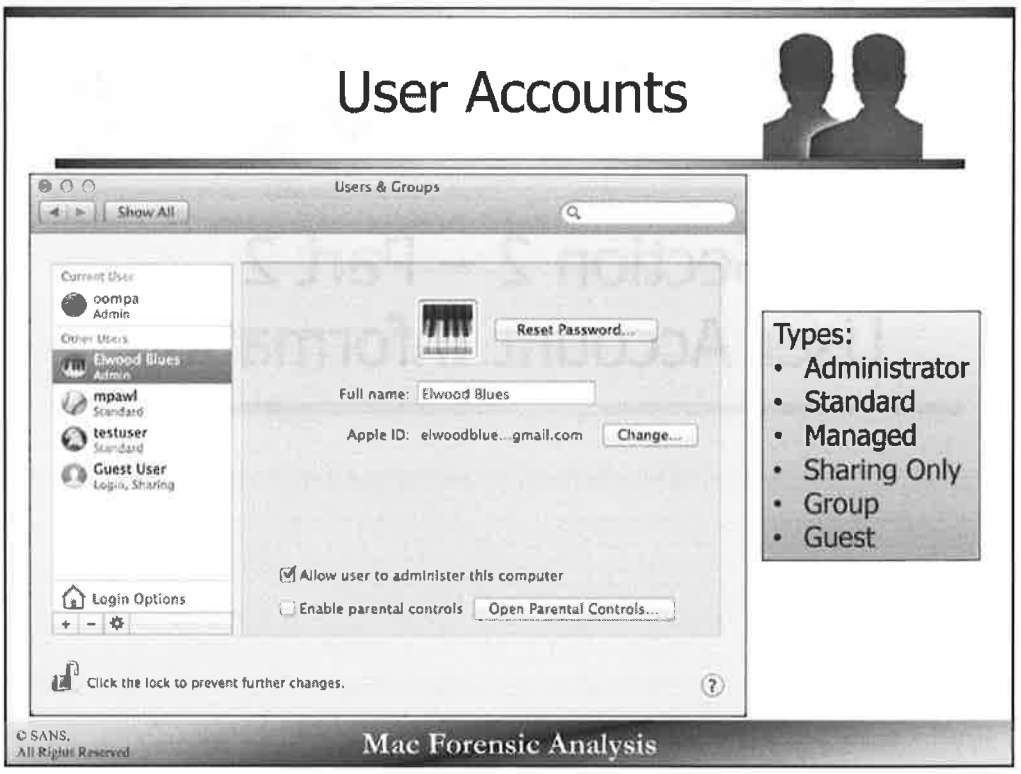
The `Caches` directory stores cached data in sub-directories that are specific to an application. It should be noted that the file names may be in "reverse DNS format" (i.e., com.app.Terminal) or as the application or company name (Adobe). It is normal to see a company name with various application directories nested underneath. For example, the `Google` directory may contain a `Chrome` folder – while the `Microsoft` directory may contain an `Office` folder.



This page intentionally left blank.



This page intentionally left blank.



Information about the user accounts is useful when trying to determine which user did what. The Users & Groups preference panel shown above allows a user to add or remove users, change passwords, change their information, enable parental controls and other administrative functions.

Each user is associated with a particular account type:
- Administrator
- Standard
- Managed with Parental Controls
- Sharing Only
- Group

The **Administrator** has full administrative access to the system while a **Standard** user has limited administrative privileges to the system; they can install software and change their own account preferences.

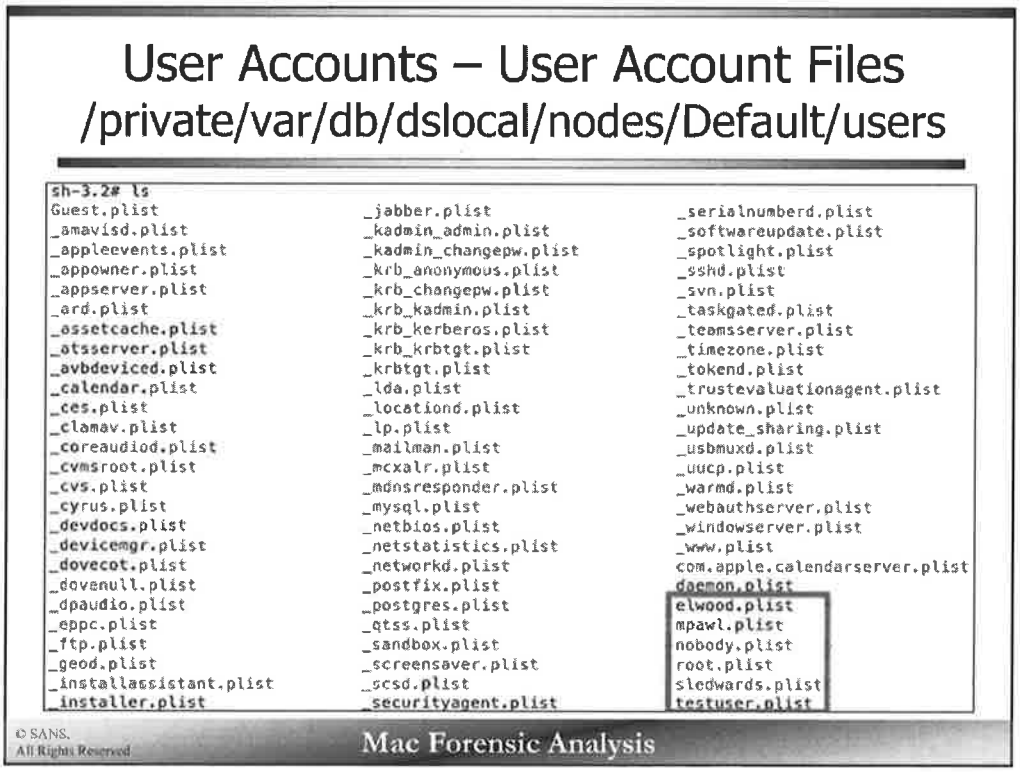
An account that is **Managed with Parental Controls**, can be restricted from using certain applications, have limited exposure to inappropriate content, or may have time usage restrictions.

A **Sharing Only** account can be used by networked users to access shared files.

A **Group** can be created to keep track of more complex user types such as in an enterprise environment.

A **Guest User** can login (without a password) and use the computer temporarily. If configured, a Guest user may be able to connect to shared folders or have parental controls enabled. If the system uses FileVault, the Guest user will only be able to access Safari. After a Guest user logs out, all data is the Guest home directory is deleted. Guest users on 10.7 and below are similar. One exception using 10.7, when FileVault is enabled, guest users cannot log on.

References:
Apple Knowledge Base Article: OS X Mavericks: Set up users on your Mac - http://support.apple.com/kb/PH14411
Apple Knowledge Base Article: OS X Mountain Lion: Set up guest users - http://support.apple.com/kb/PH11321



Each user and group has a property list file containing data about their respective user. The users are located in the `/private/var/db/dslocal/nodes/Default/users` directory while the groups are detailed in the `/private/var/db/dslocal/nodes/Default/groups` directory.

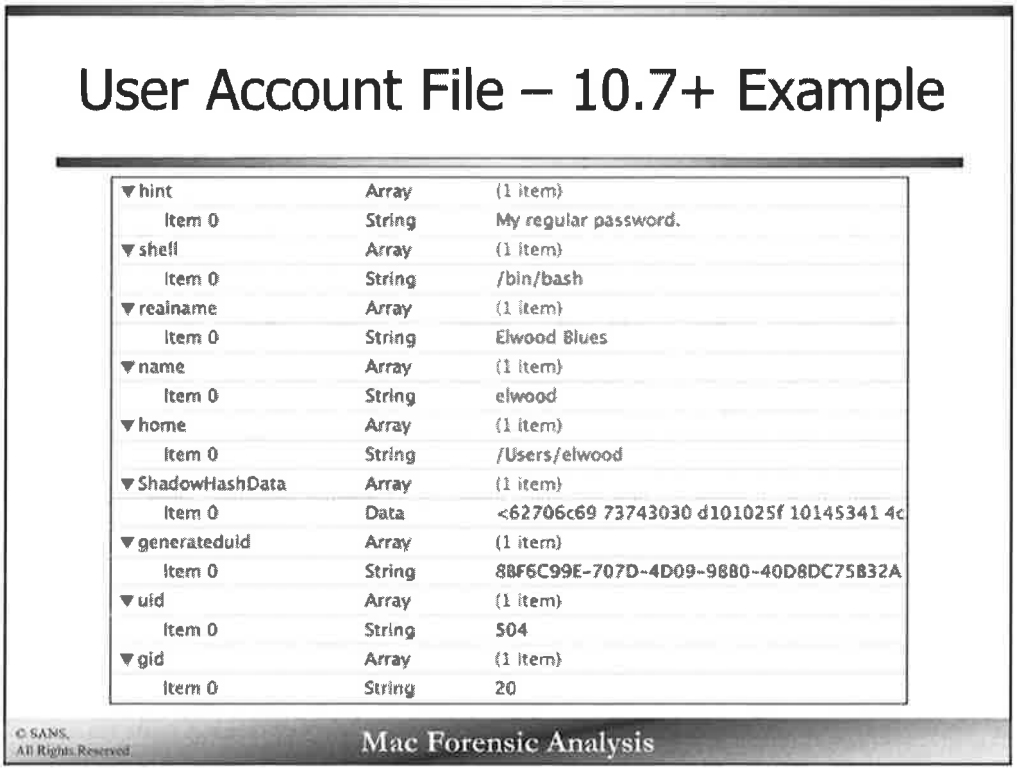
The property list files may be binary or XML depending on the OS version.

- 10.6 – XML
- 10.7+ – Binary

Access to this directory requires root privileges.

It is worth noting that users that use Open Directory (similar to Active Directory) will not have a user plist in this directory.

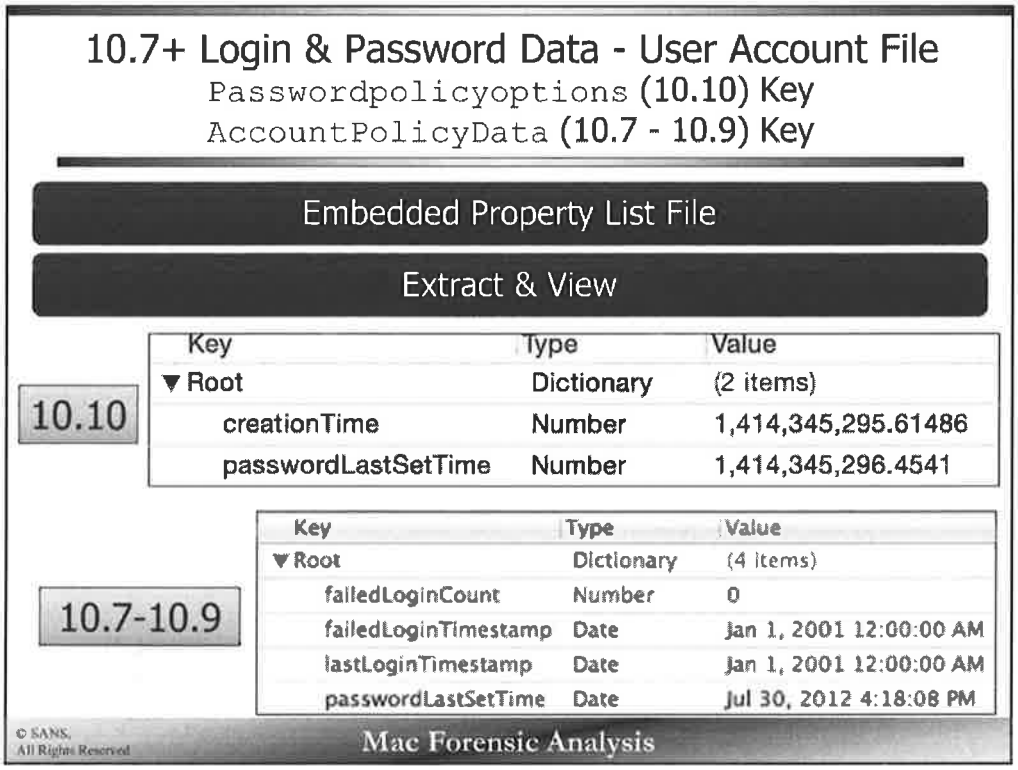
```
sh-3.2# ls
Guest.plist                     _jabber.plist                   _serialnumberd.plist
_amavisd.plist                  _kadmin_admin.plist             _softwareupdate.plist
_appleevents.plist              _kadmin_changepw.plist          _spotlight.plist
_appowner.plist                 _krb_anonymous.plist            _sshd.plist
_appserver.plist                _krb_changepw.plist             _svn.plist
_ard.plist                      _krb_kadmin.plist               _taskgated.plist
_assetcache.plist               _krb_kerberos.plist             _teamsserver.plist
_atsserver.plist                _krb_krbtgt.plist               _timezone.plist
_avbdeviced.plist               _krbtgt.plist                   _tokend.plist
_calendar.plist                 _lda.plist                      _trustevaluationagent.plist
_ces.plist                      _locationd.plist                _unknown.plist
_clamav.plist                   _lp.plist                       _update_sharing.plist
_coreaudiod.plist               _mailman.plist                  _usbmuxd.plist
_cvmsroot.plist                 _mcxalr.plist                   _uucp.plist
_cvs.plist                      _mdnsresponder.plist            _warmd.plist
_cyrus.plist                    _mysql.plist                    _webauthserver.plist
_devdocs.plist                  _netbios.plist                  _windowserver.plist
_devicemgr.plist                _netstatistics.plist            _www.plist
_dovecot.plist                  _networkd.plist                 com.apple.calendarserver.plist
_dovenull.plist                 _postfix.plist                  daemon.plist
_dpaudio.plist                  _postgres.plist                 elwood.plist
_eppc.plist                     _qtss.plist                     mpawl.plist
_ftp.plist                      _sandbox.plist                  nobody.plist
_geod.plist                     _screensaver.plist              root.plist
_installassistant.plist         _scsd.plist                     sledwards.plist
_installer.plist                _securityagent.plist            testuser.plist
```



Each user property list contains data about the specific user account. The information found on systems running 10.7+ may include the keys found on the next page. The second table shows a few keys that are different in systems running 10.6. Of note, 10.6 does not include the `ShadowHashData` key instead it uses a `.state` file, described later.

Key	Contents
jpegphoto	User Account Picture
passwordpolicyoptions	XML property list containing number of failed logons, last failed logon timestamp, last logon timestamp, password set timestamp. Default timestamps will show as 01/01/2001 00:00:00 UTC
uid	User ID (500, 501, 502, etc.)
gid	Group ID
generateduid	Generated Unique ID (GUID format)
passwd	Asterisks (not necessarily the number of characters in the password)
realname	Full name if given in account setup
name	Account username
hint	Password hint
shell	Default shell path (OS X default is /bin/bash)
home	User home directory
LinkedIdentity	XML property List containing linked identities accounts. May contain iCloud or Apple ID accounts
ShadowHashData	Binary property list containing password salted-hash data
authentication_authority	Authentication Data
kerberosKeys	Kerberos Keys
_writers_picture	Username
_writers_realname	Username
_writers_UserCertificate	Username
_writers_passwd	Username
_writers_hint	Username
_writers_jpegphoto	Username

Different Keys in 10.6	Contents
home_loc	Link to home directory sparse bundle if user has enabled FileVault
picture	Link to picture used for account

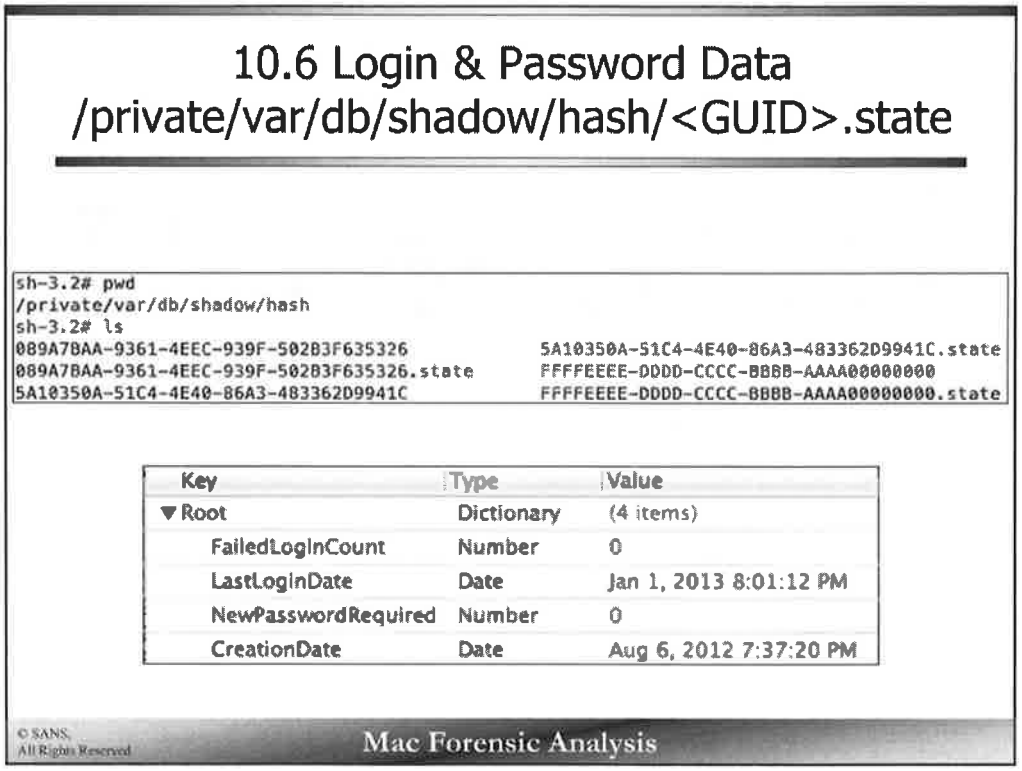


In 10.7 and newer systems, the account policy information is an embedded property list file that can be extracted, saved as a plist file and viewed.

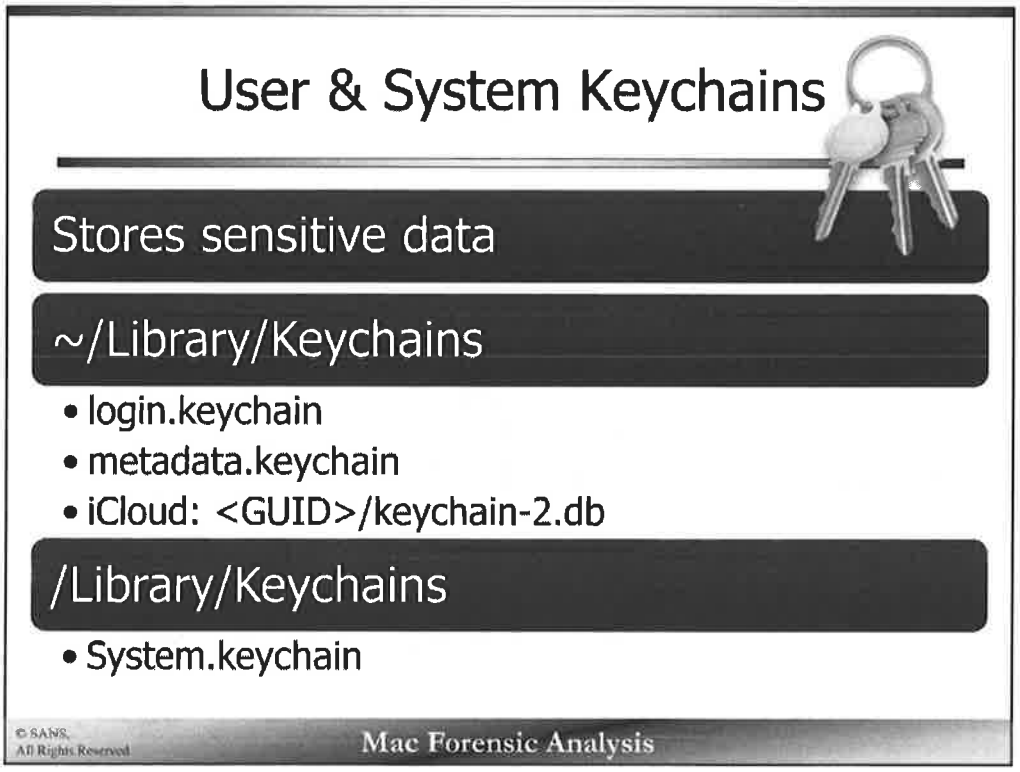
The top screenshot shows the account policy data from the users account file. On 10.10 systems this plist contains the account creation time and when the password was last set in Unix epoch time.

The bottom screenshot shows an extracted property list file from a user account file from a 10.8 system. All the timestamps are in UTC.

- Failed Login Count – How many logins were attempted but failed
- Failed Login Timestamp –Time of last failed login.(10.7 systems do not appear to update this, but do update the failed login count.)
- Last Login Timestamp – Last Login Time, does not appear to get used with default settings.
- Password Last Set Time (10.7 – `passwordTimestamp`) – When the password was last changed



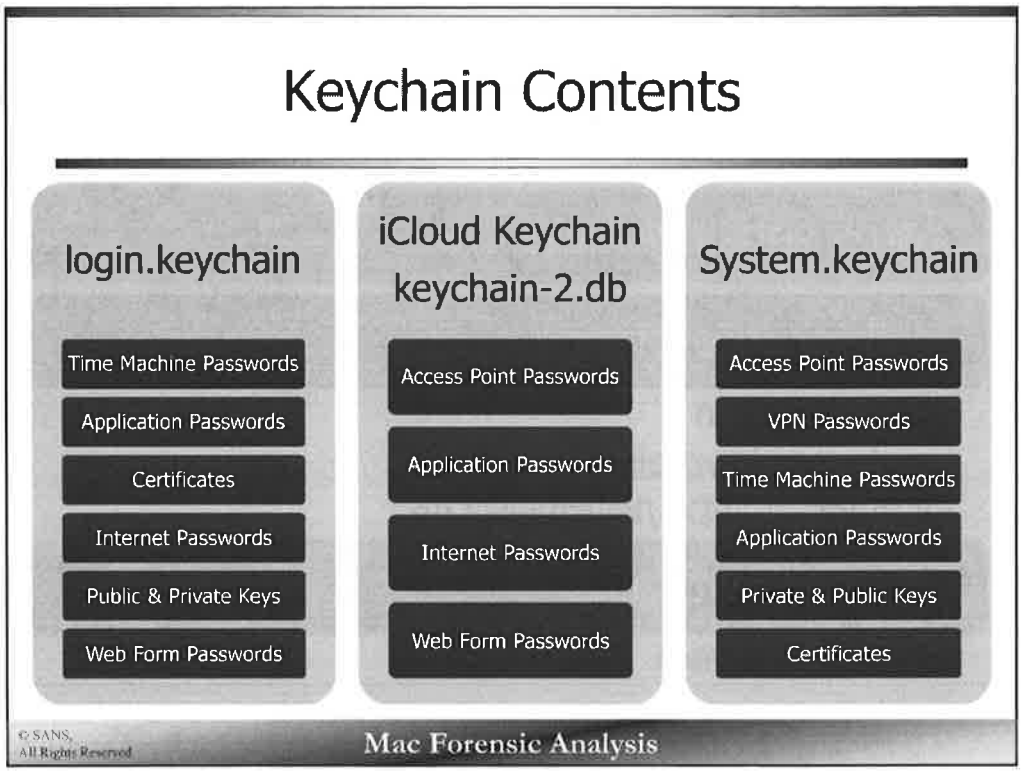
Some of the same information can be found in a binary property list file in the `/private/var/db/shadow/hash` directory with the filename `<GUID>.state` on a 10.6 system. The GUID matches that of the user whose information you are interested in. The key, `NewPasswordRequired` was changed to the `PasswordLastSetTime` key in newer versions of OS X.



The Keychains on a system are used to store sensitive data such as usernames, passwords, public and private keys. There are at least three keychains on a system. Two user keychains, `login.keychain` and `metadata.keychain` which hold user-sensitive items. The `System.keychain` which holds sensitive system-wide information. Other keychains may exist on the system, for example keychains with specific certificates may exist in the System Library.

- `/System/Library/Keychains/SystemCACertificates.keychain`
- `/System/Library/Keychains/SystemRootCertificates.keychain`

If enabled, the iCloud keychain will be stored in another GUID labeled directory in the `keychain-2.db` SQLite database file.



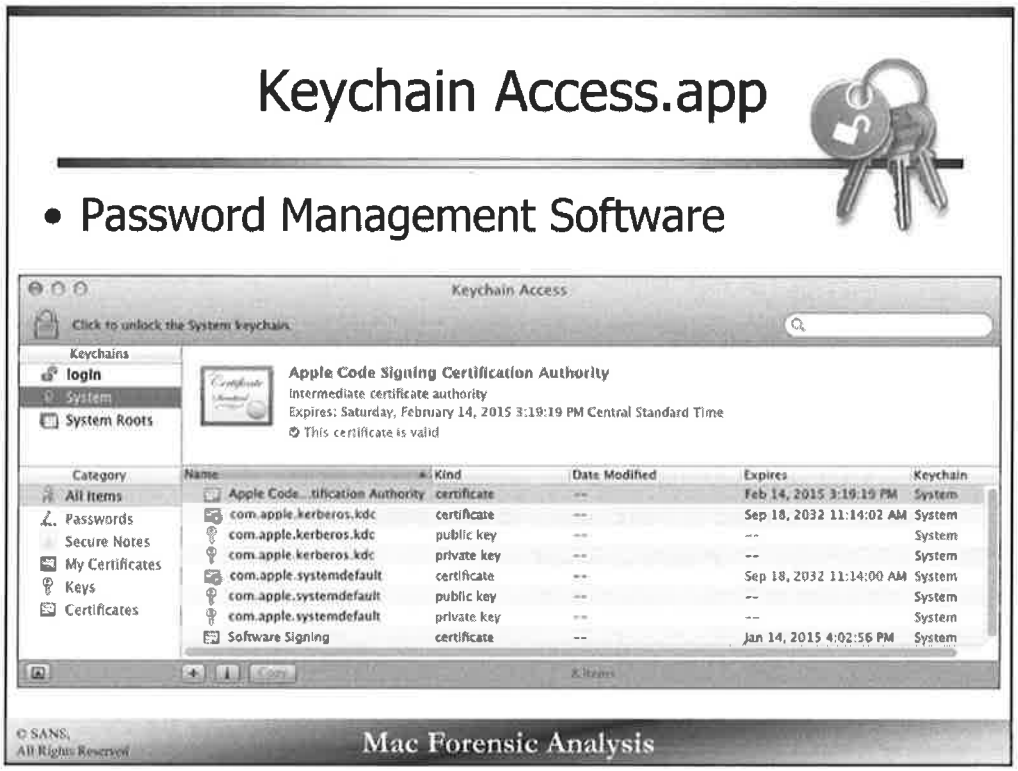
The two most important keychains are the user's `login.keychain` and the `System.keychain`.

The `login.keychain` may contain the users' passwords for Access Points, Time Machine, application and websites. It also stores the user's certificates, and public and private keys. Amongst all of this information there may also be notes if the user chooses to create them here. The default `login.keychain` password is the user's account password.

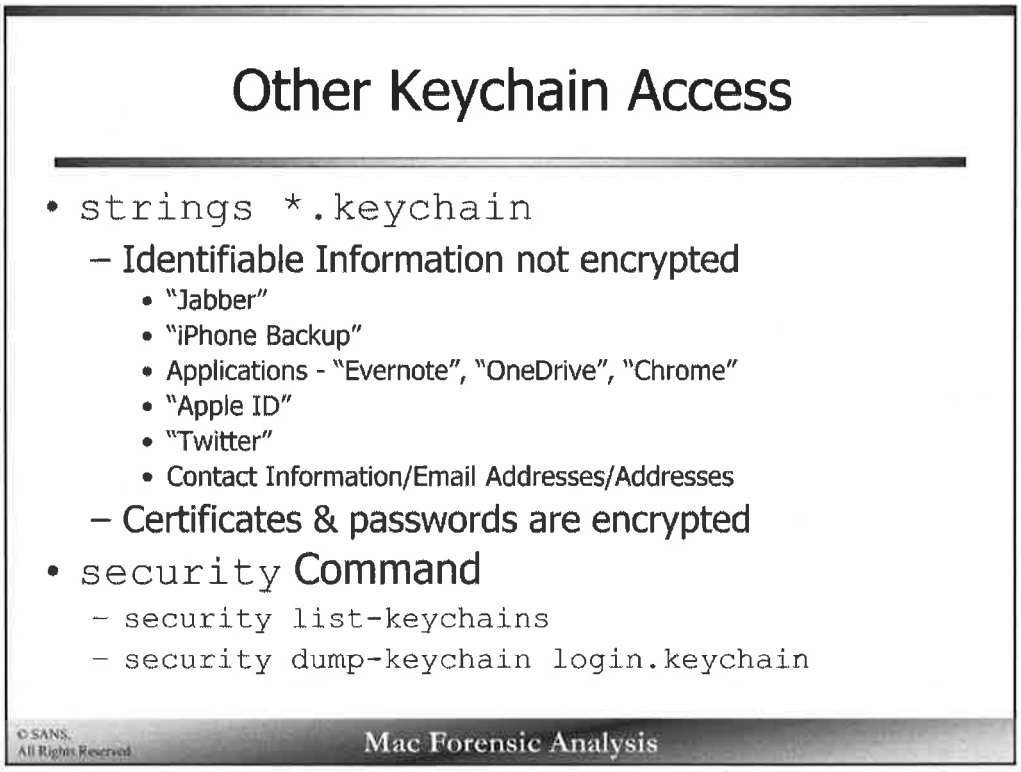
The iCloud keychain stores information that may not only be used on OS X but on iDevices as well – an access point used on the suspects iPhone may be synced to the iCloud keychain.

The `System.keychain` also contains passwords for VPNs, Access Points, Time Machine and applications. It may also contain private and public keys, and certificates.

The `metadata.keychain`, if you are curious, contains data related to the user's spotlight metadata index.



The Keychain Access application can be used to interact with the keychains. This application works like any other password management system. Click on the keychain in the upper right and the contents will be shown in the main screen. Be sure to choose "All Items" in the Category listing, otherwise you may not see all the data stored.



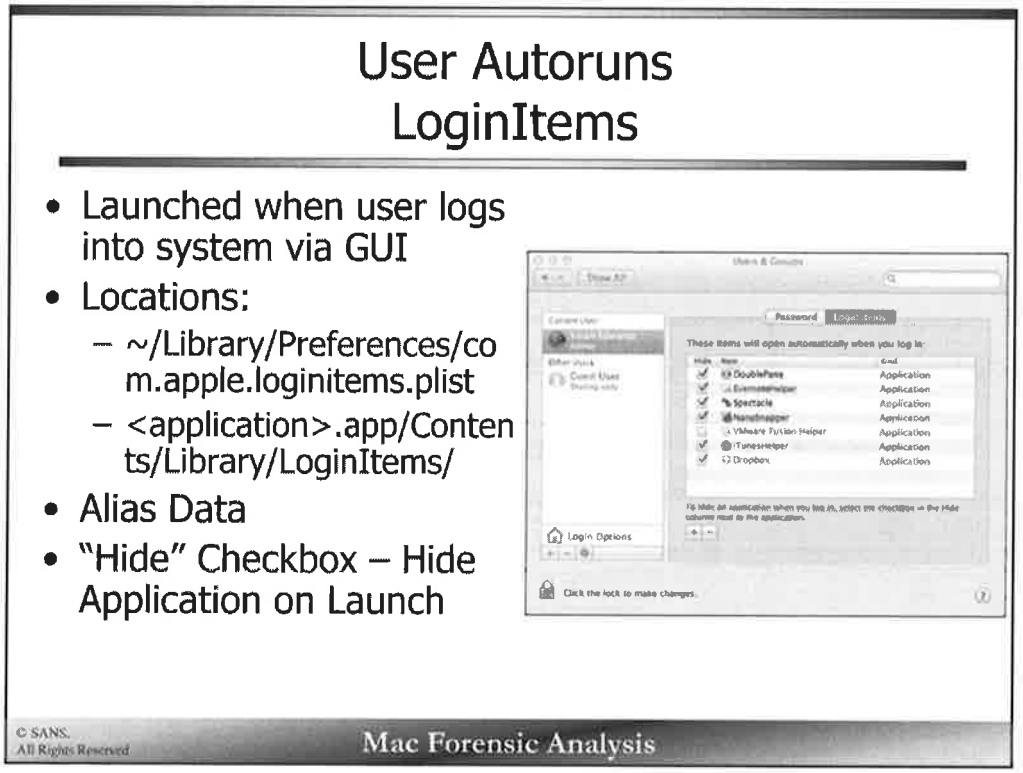
The `strings` command can be used to identify interesting keywords in a keychain that are not encrypted. This can give you an idea of what might be stored in the keychain if you are unable to access it with the user's password, (remember a user does not have to use their login password, but it is used by default).

Strings like the following may suggest certain items are saved in the keychain:

- "`imagent`" – iChat
- "`iCloud`" – iCloud Account
- "`com.apple.account.google`" – Google Account
- "`Jabber: user@gmail.com`" - Google Chat Account
- "`AIM: <userhandle>`" – AIM Account
- "`FaceTime: <acct>`" - FaceTime Account
- "`<system>._rfb._tcp.local`" – Screen Sharing

The `security` command can be used to list keychains and dump keychain contents (though not the encrypted contents) on a live system. You'll see many of the same strings with more context.

The security dump-keychain command can be particularly useful when used on a live system – it can sometimes dump plaintext passwords that have been unlocked. If the user is logged in but will not give up their password, you may be able to run this command and find it yourself!



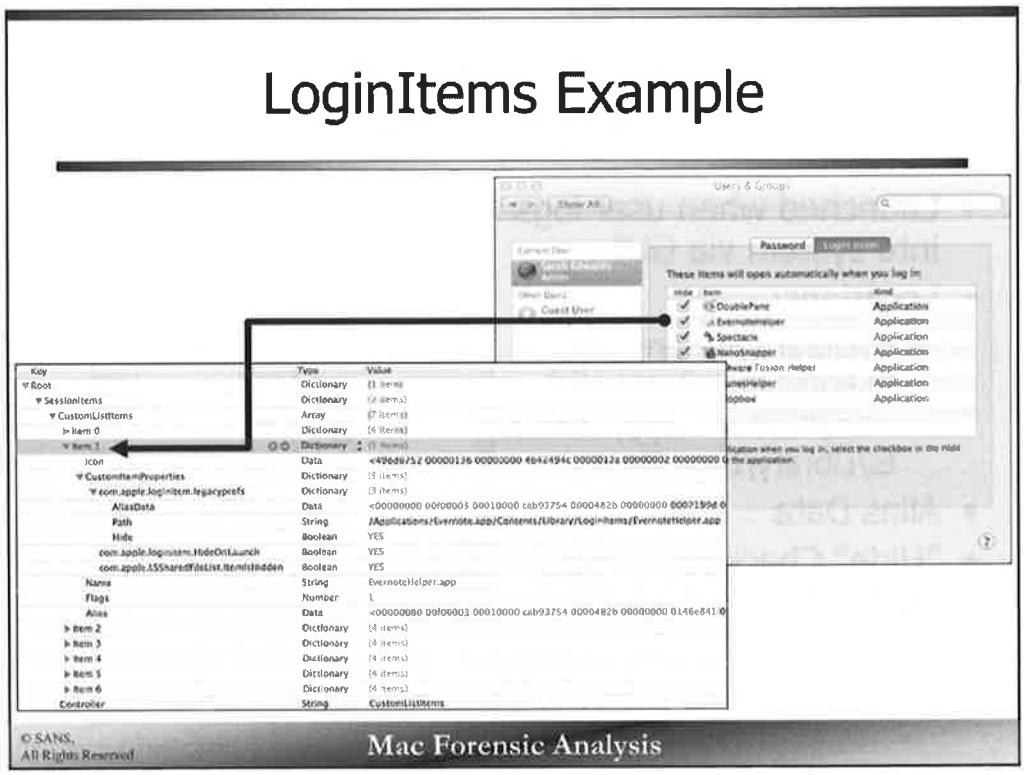
Each user may have a set of autoruns that get launched when a user logs in via the system GUI login.

The `com.apple.loginitems.plist` contains each login item. Each item has a `Alias` pointer to the `LoginItem` program located in the application bundle.

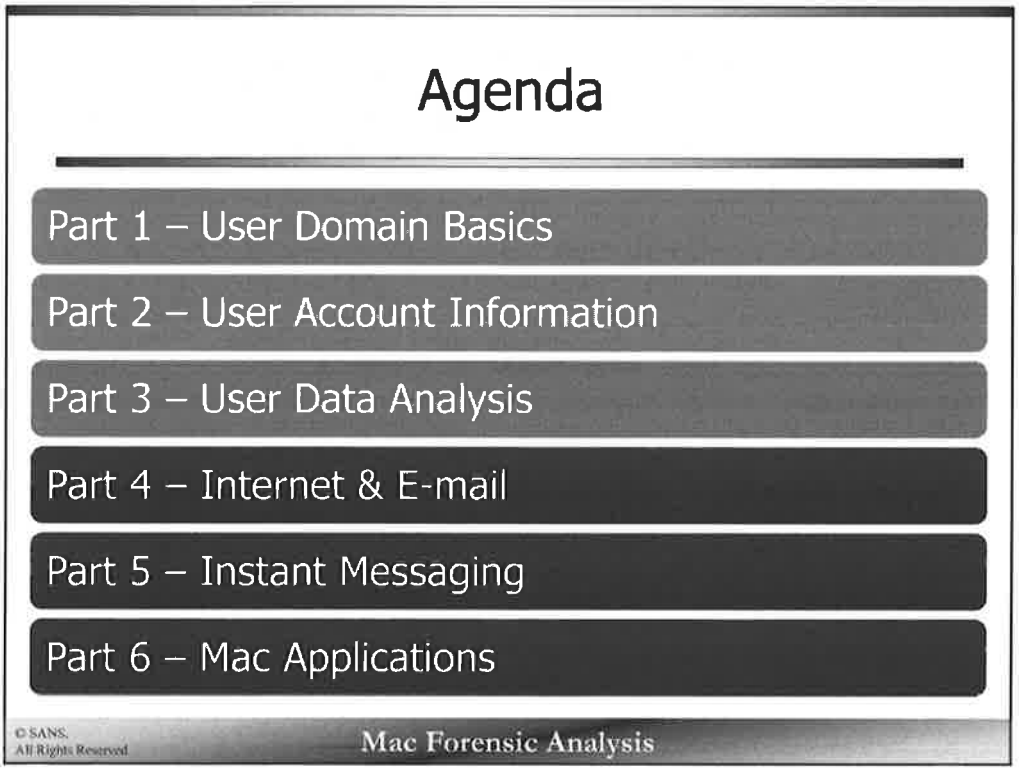
Each login item can be "hidden" on launch, this means the login item is hidden from view when it is starting up.

Another important autorun folder under the user context is the `~/Library/LaunchAgents/` directory. This will be covered in depth in Section 3.

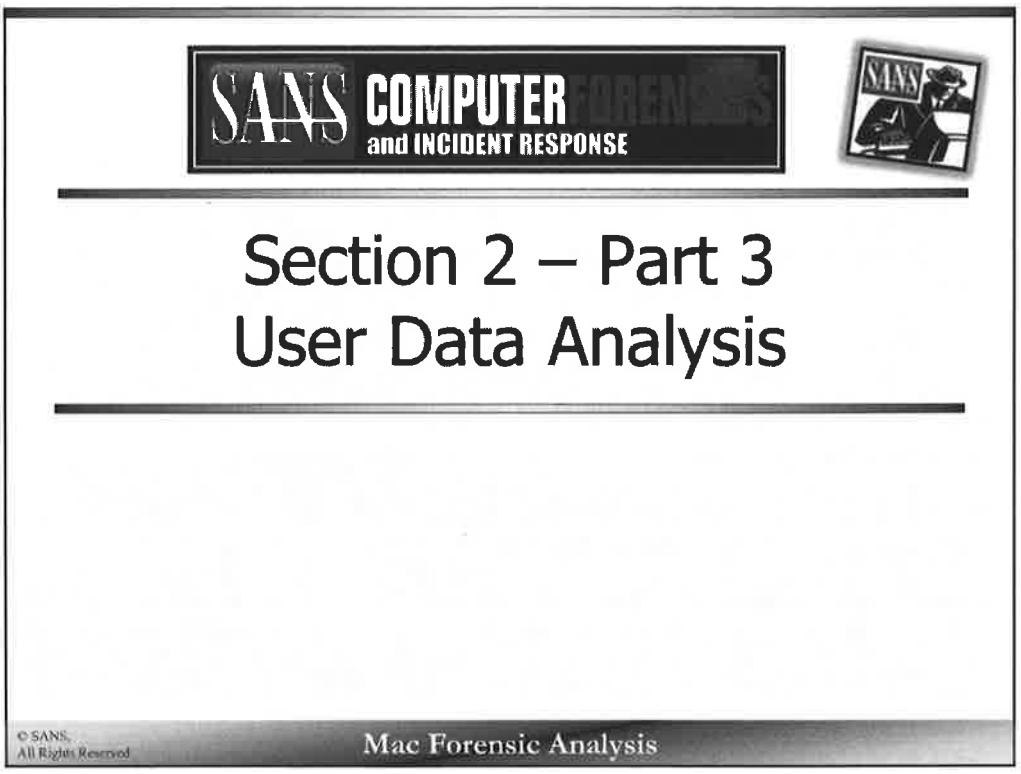
`LoginItems`, while usually legitimate, may be used as a malware persistence mechanism.



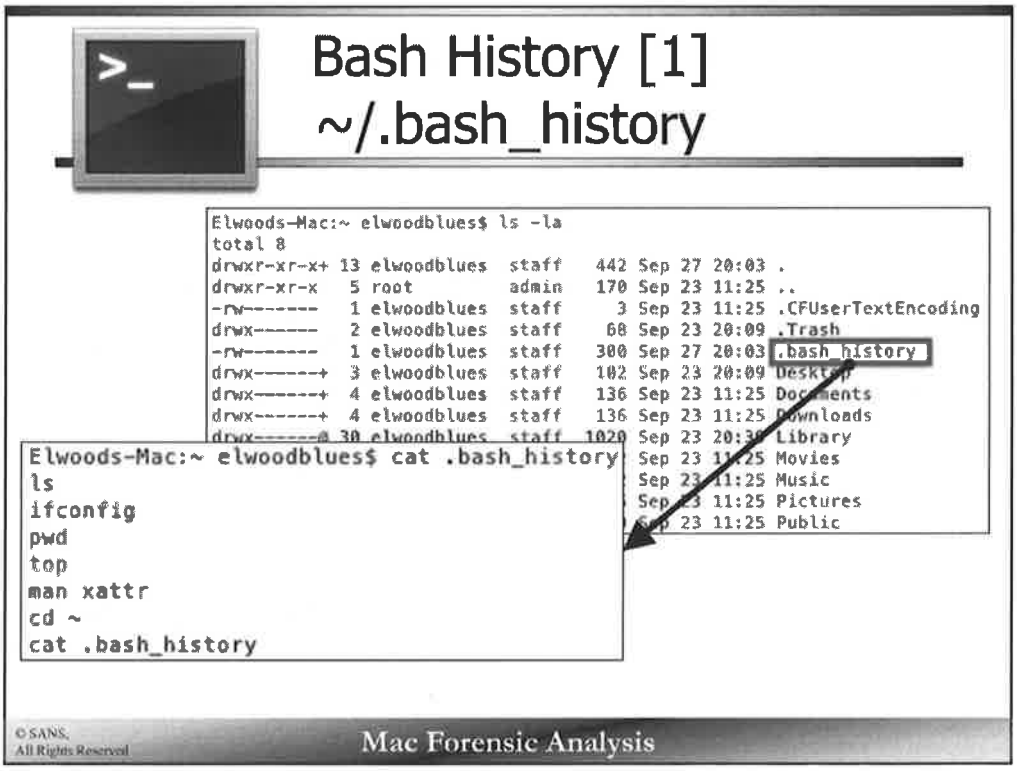
The `com.apple.loginitems.plist` contains each login item. Each item has an associated icon, alias data blob, the path to the login item, and a binary value showing if the item was "hidden" on launch.



This page intentionally left blank.



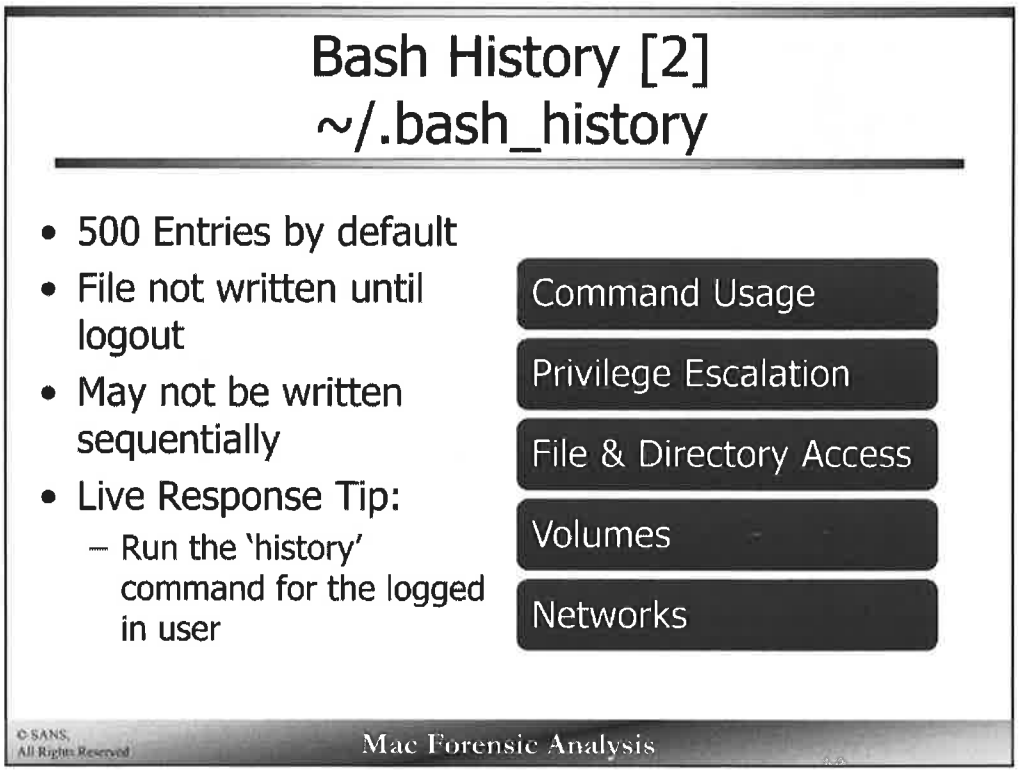
This page intentionally left blank.



The bash command history is located in the respective home directory for each user as a hidden file.

- `/Users/<user>/.bash_history`

This file is a plaintext file containing commands that were run in order of execution. This file will be created when the Terminal application has been used for the first time. If a user never used the Terminal application, this file would not be created.



This file contains the command history from the Bash shell. By default, the history file contains a maximum of 500 commands.

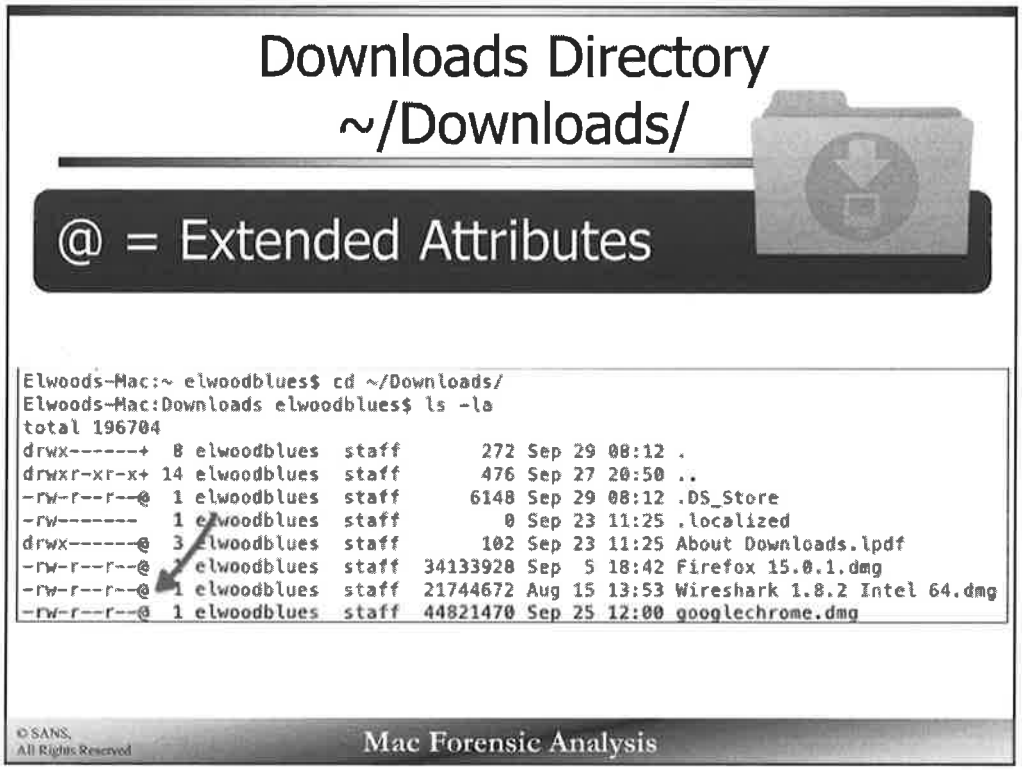
These commands can be useful in showing:

- What types of applications or programs the user was accessing
- Privileges were escalated to perform administrative functions or possible suspicious actions
- Files and directories accessed
- Mounted volumes
- Systems and networks that were accessed
- Etc.

Each command is shown respective to other commands that were previously entered. It should be noted that each command has no date or time context associated in this file.

From an incident response perspective, the `.bash_history` file is not updated until the user is logged out.

The history can be viewed on a live system by using the `history` command, this will show the history of the current user.

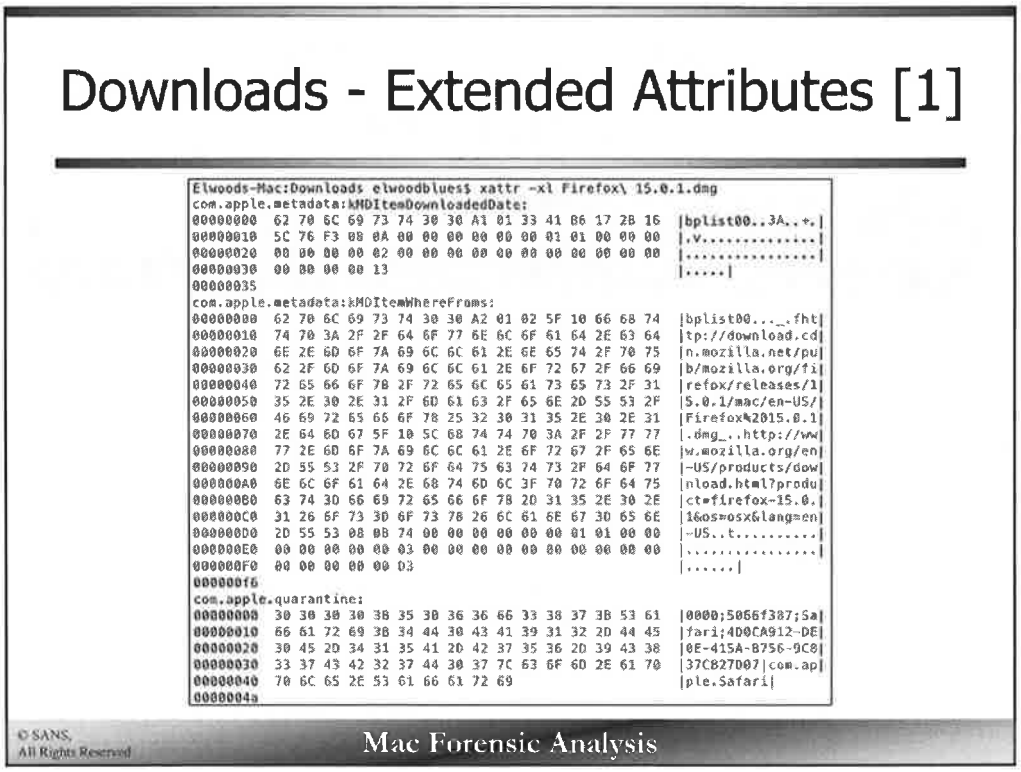


Each user has their own `Downloads` directory. This directory contains all the downloads from applications such as web browsers and e-mail that use this folder as their default download location. Most applications will have a preference setting that users can change; however, by default this directory will be used.

On a well used system this directory will contain months, even years worth of downloads.

Each item that was downloaded will likely have extended attributes, showing metadata of the download. To see if a file contains extended attributes you can run the command; `ls -la`

Files with extended attributes will have the "`@`" at the end of the permissions.



Extended attributes can contain a variety of information. The attributes mainly found on downloaded files are:

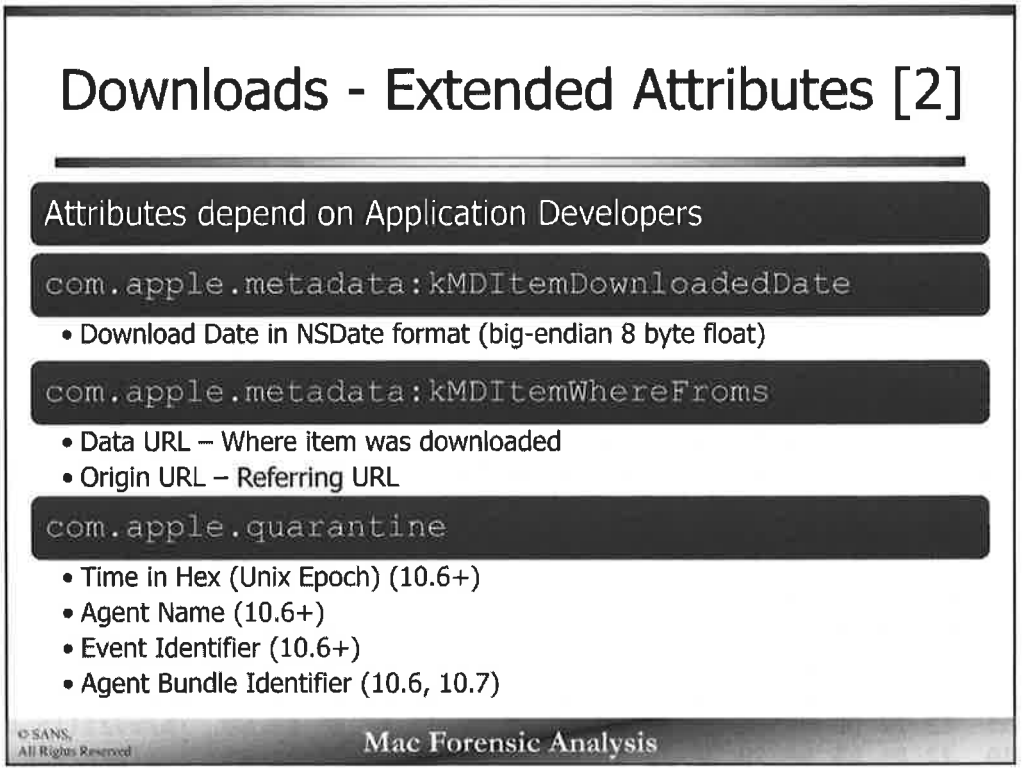
- `com.apple.metadata:kMDItemDownloadedDate`
- `com.apple.metadata:kMDItemWhereFroms`
- `com.apple.quarantine`

Extended attributes can be viewed by using the `xattr` command shown above.

The `xattr` command has the following output options:

- `-x` – View in hex
- `-l` – Outputs the attribute name and contents

```
Elwoods-Mac:Downloads elwoodblues$ xattr -xl Firefox\ 15.0.1.dmg
com.apple.metadata:kMDItemDownloadedDate:
00000000  62 70 6C 69 73 74 30 30 A1 01 33 41 B6 17 2B 16  |bplist00..3A..+.|
00000010  5C 76 F3 08 0A 00 00 00 00 00 00 01 01 00 00 00  |.v..............|
00000020  00 00 00 00 02 00 00 00 00 00 00 00 00 00 00 00  |................|
00000030  00 00 00 00 13                                   |.....|
00000035
com.apple.metadata:kMDItemWhereFroms:
00000000  62 70 6C 69 73 74 30 30 A2 01 02 5F 10 66 68 74  |bplist00..._.fht|
00000010  74 70 3A 2F 2F 64 6F 77 6E 6C 6F 61 64 2E 63 64  |tp://download.cd|
00000020  6E 2E 6D 6F 7A 69 6C 6C 61 2E 6E 65 74 2F 70 75  |n.mozilla.net/pu|
00000030  62 2F 6D 6F 7A 69 6C 6C 61 2E 6F 72 67 2F 66 69  |b/mozilla.org/fi|
00000040  72 65 66 6F 78 2F 72 65 6C 65 61 73 65 73 2F 31  |refox/releases/1|
00000050  35 2E 30 2E 31 2F 6D 61 63 2F 65 6E 2D 55 53 2F  |5.0.1/mac/en-US/|
00000060  46 69 72 65 66 6F 78 25 32 30 31 35 2E 30 2E 31  |Firefox%2015.0.1|
00000070  2E 64 6D 67 5F 10 5C 68 74 74 70 3A 2F 2F 77 77  |.dmg_..http://ww|
00000080  77 2E 6D 6F 7A 69 6C 6C 61 2E 6F 72 67 2F 65 6E  |w.mozilla.org/en|
00000090  2D 55 53 2F 70 72 6F 64 75 63 74 73 2F 64 6F 77  |-US/products/dow|
000000A0  6E 6C 6F 61 64 2E 68 74 6D 6C 3F 70 72 6F 64 75  |nload.html?produ|
000000B0  63 74 3D 66 69 72 65 66 6F 78 2D 31 35 2E 30 2E  |ct=firefox-15.0.|
000000C0  31 26 6F 73 3D 6F 73 78 26 6C 61 6E 67 3D 65 6E  |1&os=osx&lang=en|
000000D0  2D 55 53 08 0B 74 00 00 00 00 00 00 01 01 00 00  |-US..t..........|
000000E0  00 00 00 00 00 03 00 00 00 00 00 00 00 00 00 00  |................|
000000F0  00 00 00 00 00 D3                                |......|
000000f6
com.apple.quarantine:
00000000  30 30 30 30 3B 35 30 36 36 66 33 38 37 3B 53 61  |0000;5066f387;Sa|
00000010  66 61 72 69 3B 34 44 30 43 41 39 31 32 2D 44 45  |fari;4D0CA912-DE|
00000020  30 45 2D 34 31 35 41 2D 42 37 35 36 2D 39 43 38  |0E-415A-B756-9C8|
00000030  33 37 43 42 32 37 44 30 37 7C 63 6F 6D 2E 61 70  |37CB27D07|com.ap|
00000040  70 6C 65 2E 53 61 66 61 72 69                    |ple.Safari|
0000004a
```



The contents of extended attribute data can vary by operating system or application. Each browser saves different extended attributes:

Safari – All in slide

Chrome – Does not use `kMDItemDownloadedDate`

FireFox – Only `com.apple.quarantine`

The extended attributes can contain useful information such as:

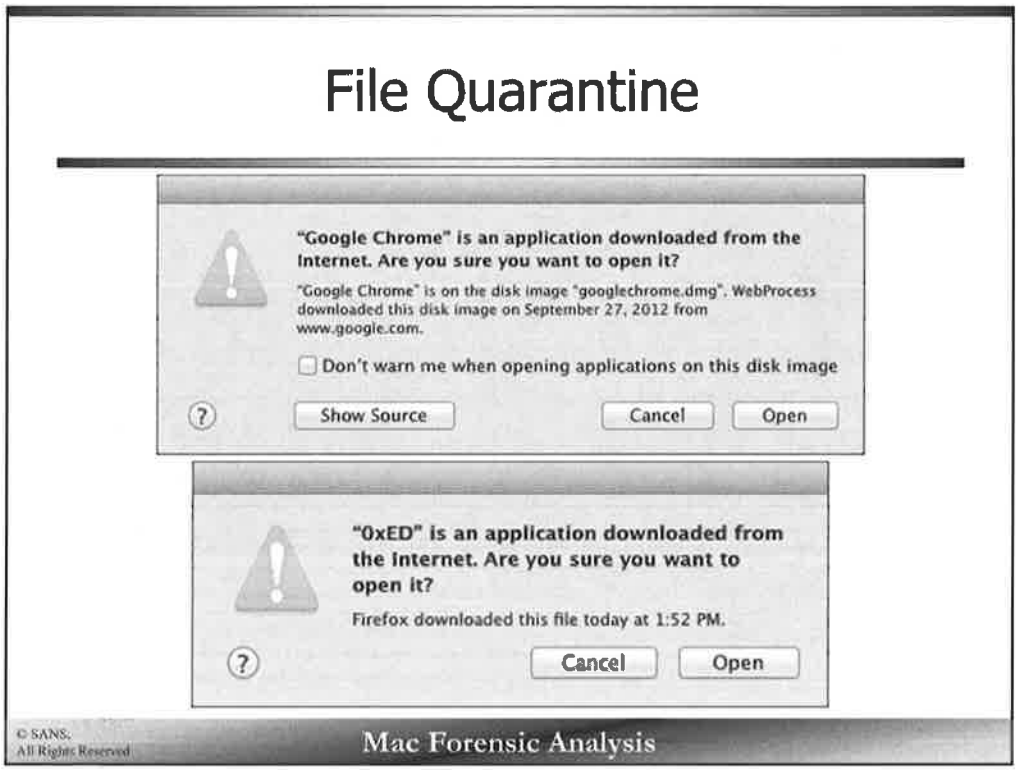
- Download dates
- URLS – The Data URL shows where the data was actually downloaded from, the Origin URL shows where the download was referred from.
- Quarantine Data – Contains information related to Apple's file quarantine system.

References:

NSDate

http://opensource.apple.com/source/CF/CF-368/NumberDate.subproj/CFDate.h

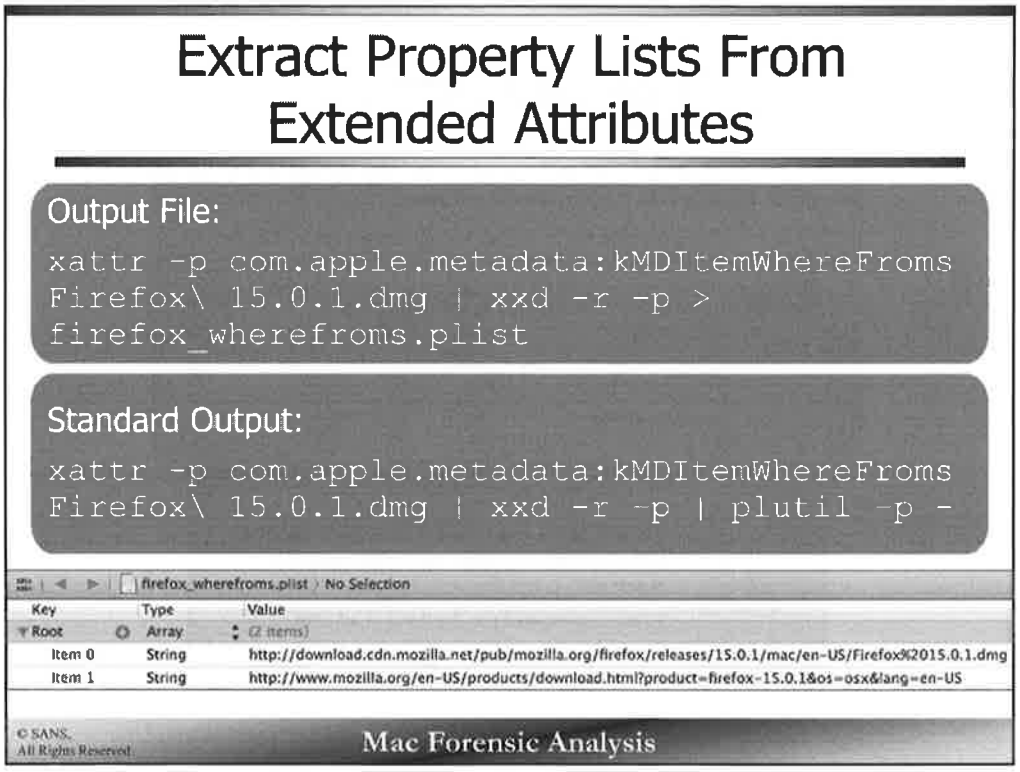
http://opensource.apple.com/source/CF/CF-550/CFBinaryPList.c



Apple uses file quarantine to protect the system by checking files against a malicious file database (more on this in the Advanced Analysis section). It is also used to inform the user where and when a file was downloaded from when the user attempts to execute the file.

In the slide above, the top screenshot shows that a disk image named `googlechrome.dmg`. This file was downloaded on September 27th from `google.com`.

The screenshot on the bottom shows the application '`0xED`" was downloaded at 1:52pm using Firefox.



Some extended attributes contain binary property lists. To extract these for more detailed analysis, we can use the `xattr` command with the `-p` option. This option "prints" the attribute data .

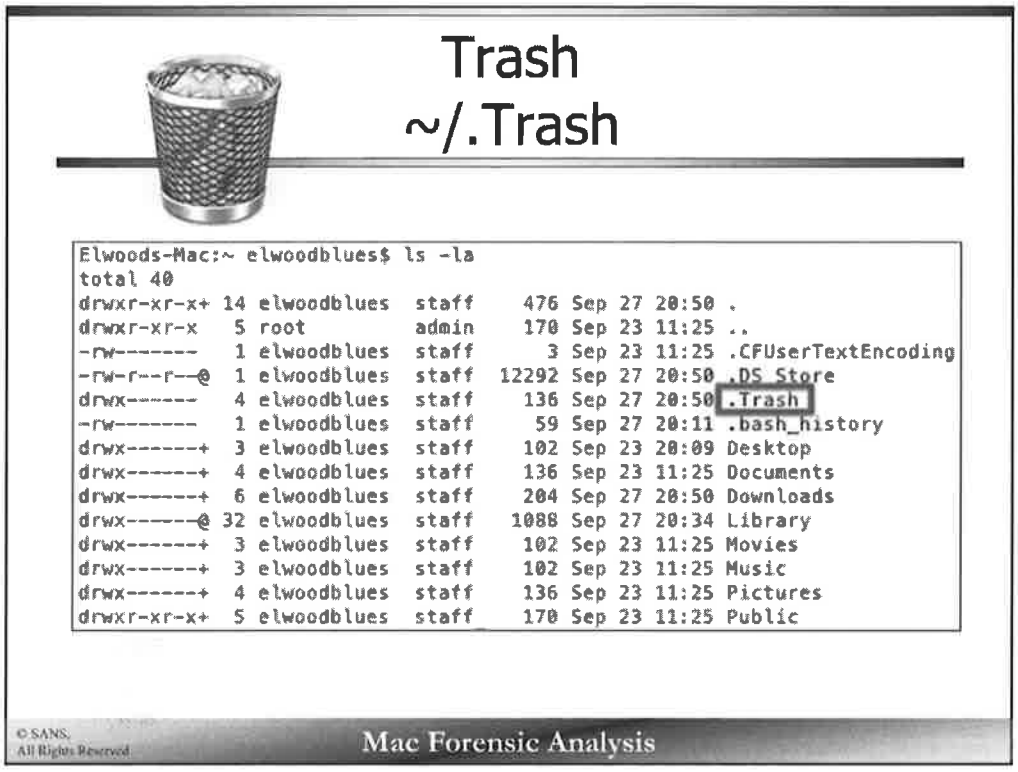
Using only the `-p` option, it will print the hex version of the data. To create a file we must use the `xxd` command to "revert to binary" (`-r`) and print to "plaintext" (`-p`). We can use the ">" symbol to redirect the output to a file.

Together the commands will look like this. This command prints the output of the `com.apple.metadata:kMDItemWhereFroms` attribute to a file named `firefox_wherefroms.plist`.

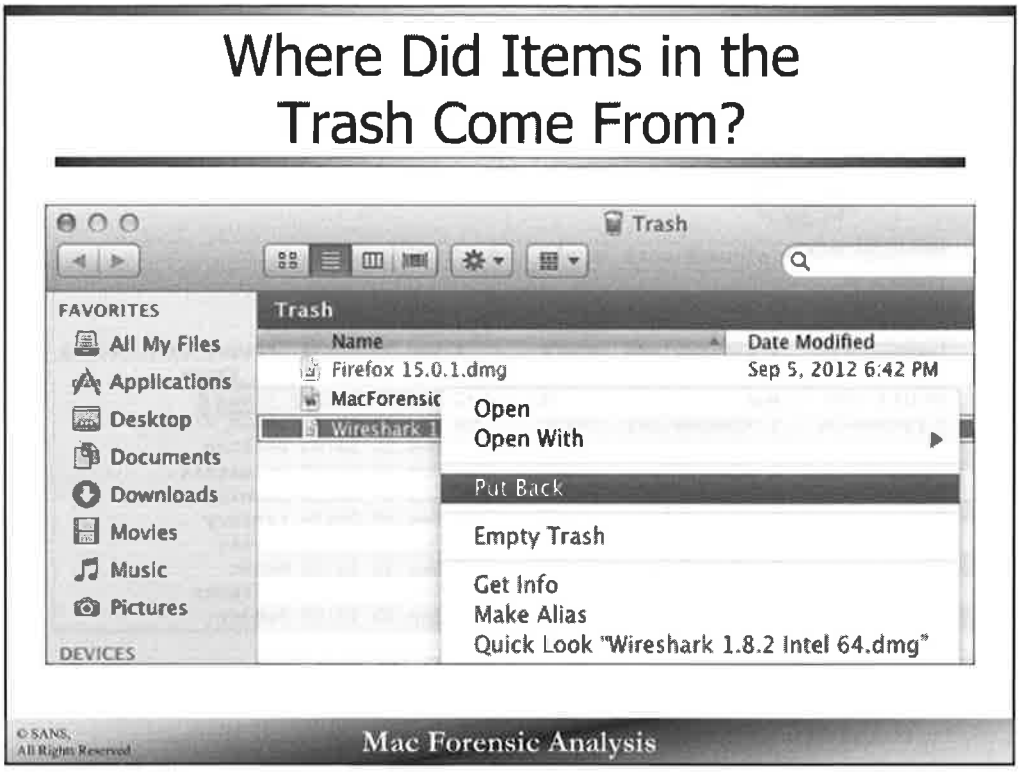
```
xattr -p com.apple.metadata:kMDItemWhereFroms Firefox\ 15.0.1.dmg | xxd -r
-p > firefox_wherefroms.plist
```

To print a binary plist directly to standard out on the Terminal use the following command format:

```
xattr -p <attribute name> <file> | xxd -r -p | plutil -p -
```

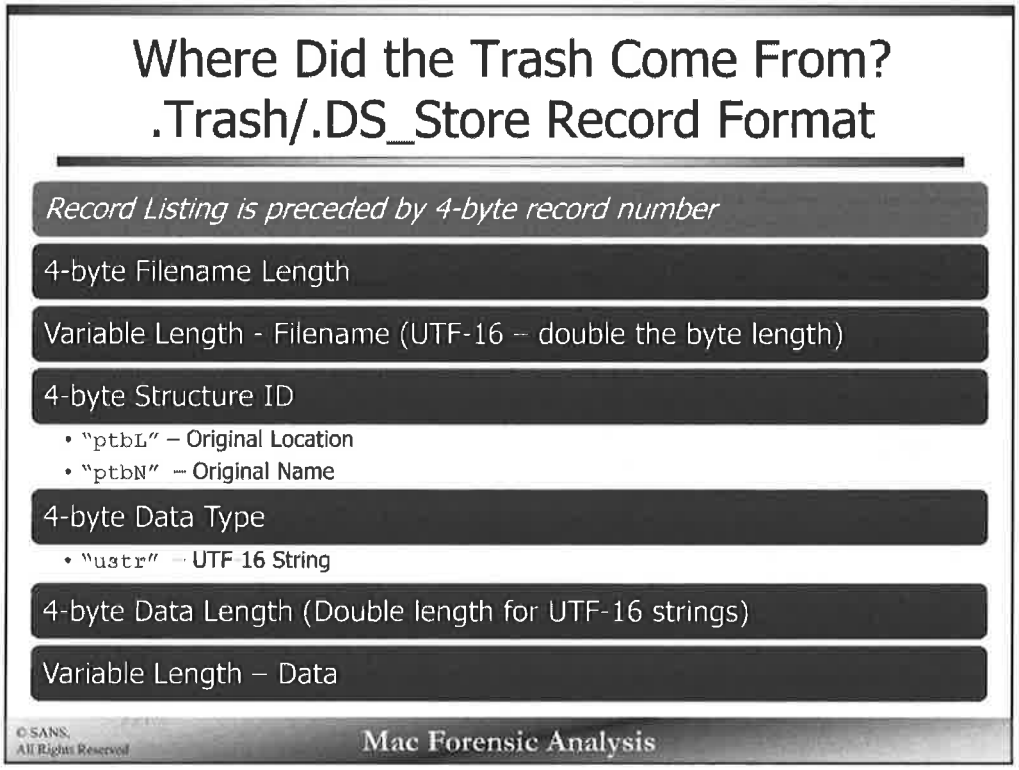


Each user has their own hidden `.Trash` directory. This is located in the root of their home directory.



The `.Trash` contains "deleted" items. Each trashed file can be restored using the "Put Back" option in the right-click context menu.

This "Put Back" data can be found in the hidden `.DS_Store` file in the `.Trash` directory.



Each record in the `.DS_store` file can be identified by a UTF-16 filename, just prior to this filename is a 4-byte filename length. These are followed by a 4-byte Structure ID, and 4-byte Data type. Some of these Structure IDs and Data types have been documented in the reference pages below. The final part of the record is the data size and data – this can be a variable size.

Each deleted file will have:

- Filename
- Original File Path

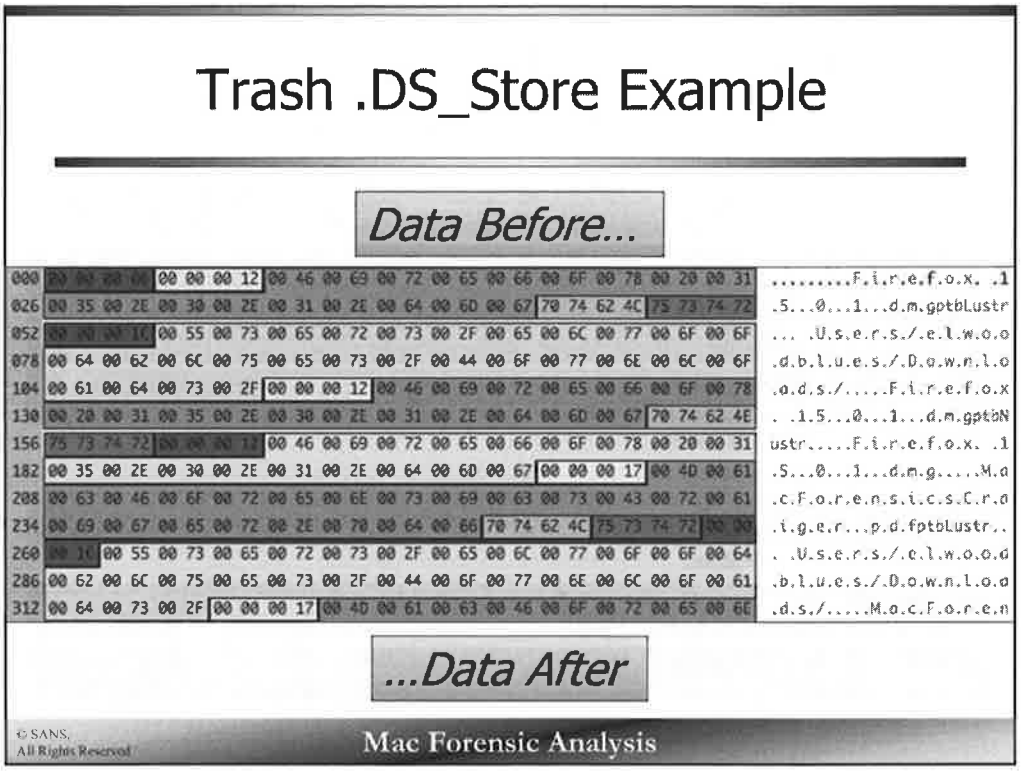
These records are preceded by a 4-byte record number. This will determine how many records are in this file.

References:

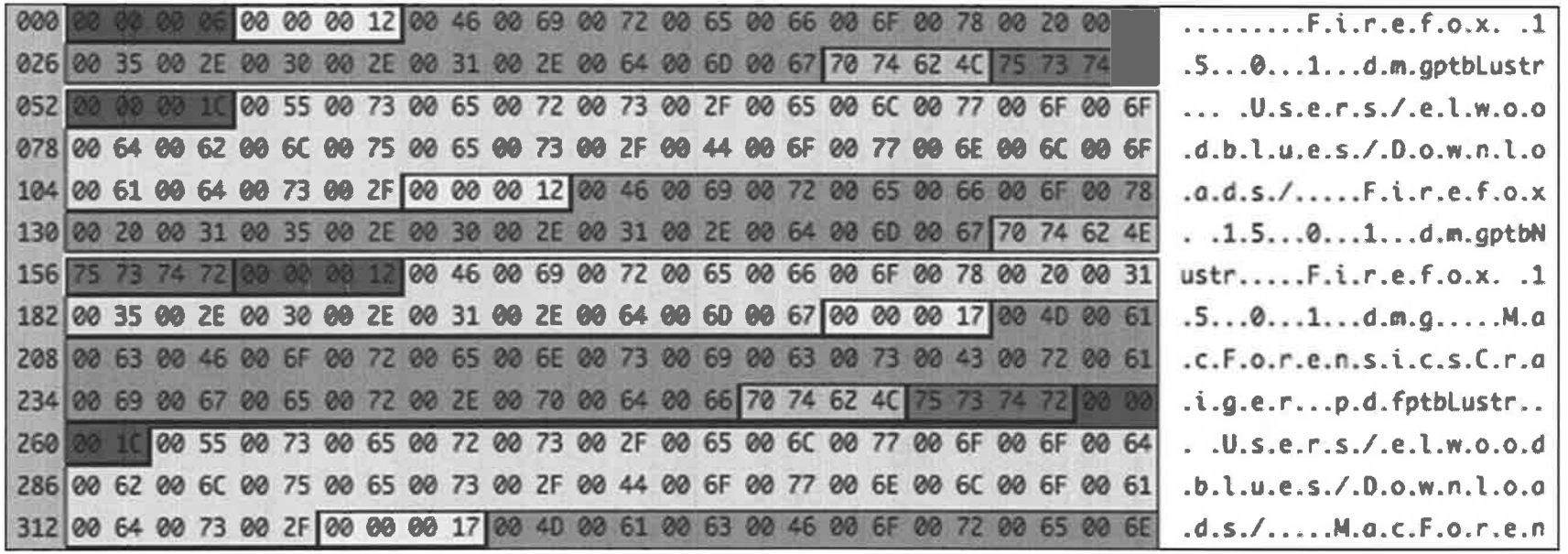
http://search.cpan.org/~wiml/Mac-Finder-DSStore-0.95/DSStoreFormat.pod

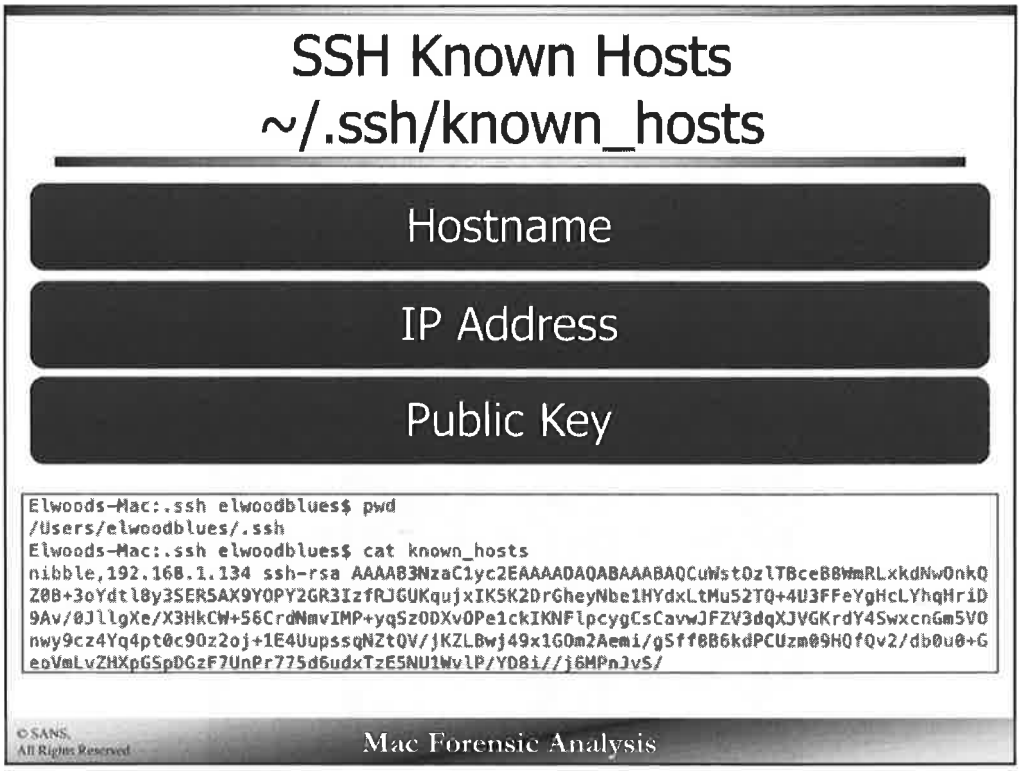
https://wiki.mozilla.org/DS_Store_File_Format

https://github.com/dscho/dsstore



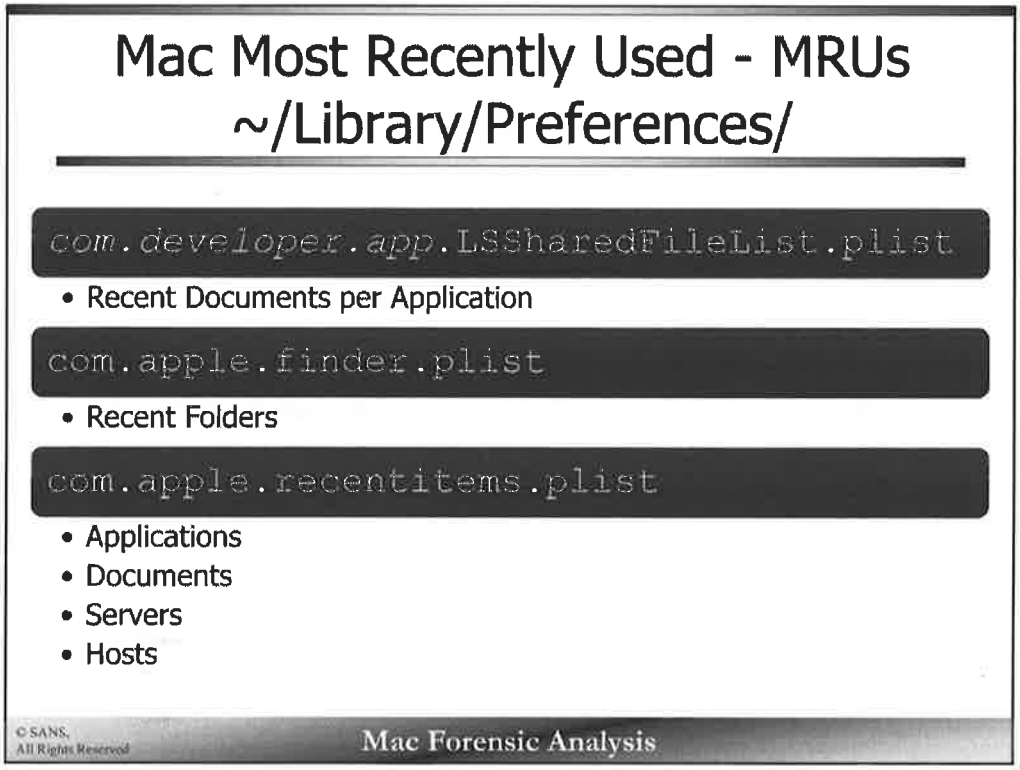
Field	Size (bytes)	Data
Number of Records	4	0x00000006 = 6 Records
Record 1 – Filename Size	4	0x00000012 = 18 (characters)
Record 1 - Filename	Variable	"Firefox 15.0.1.dmg" (UTF-16 – length doubled)
Record 1 – Structure ID	4	"ptbL"
Record 1 – Data Type	4	"ustr"
Record 1 – Data Size	4	0x0000001C = 28 (characters)
Record 1 - Data	Variable	"Users/elwoodblues/Downloads/" (UTF-16 – length doubled)
Record 2 – Filename Size	4	0x00000012 = 18 (characters)
Record 2 - Filename	Variable	"Firefox 15.0.1.dmg" (UTF-16 – length doubled)
Record 2 – Structure ID	4	"ptbN"
Record 2 – Data Type	4	"ustr"
Record 2 – Data Size	4	0x00000012 = 18 (characters)
Record 2 - Data	Variable	"Firefox 15.0.1.dmg" (UTF-16 – length doubled)


000 00 00 00 06 00 00 00 12 00 46 00 69 00 72 00 65 00 66 00 6F 00 78 00 20 00 .....F.i.r.e.f.o.x. .1
026 00 35 00 2E 00 30 00 2E 00 31 00 2E 00 64 00 6D 00 67 70 74 62 4C 75 73 74 .5...0...1...d.m.gptbLustr
052 00 00 00 1C 00 55 00 73 00 65 00 72 00 73 00 2F 00 65 00 6C 00 77 00 6F 00 6F ... .U.s.e.r.s./.e.l.w.o.o
078 00 64 00 62 00 6C 00 75 00 65 00 73 00 2F 00 44 00 6F 00 77 00 6E 00 6C 00 6F .d.b.l.u.e.s./.D.o.w.n.l.o
104 00 61 00 64 00 73 00 2F 00 00 00 12 00 46 00 69 00 72 00 65 00 66 00 6F 00 78 .a.d.s./.....F.i.r.e.f.o.x
130 00 20 00 31 00 35 00 2E 00 30 00 2E 00 31 00 2E 00 64 00 6D 00 67 70 74 62 4E . .1.5...0...1...d.m.gptbN
156 75 73 74 72 00 00 00 12 00 46 00 69 00 72 00 65 00 66 00 6F 00 78 00 20 00 31 ustr.....F.i.r.e.f.o.x. .1
182 00 35 00 2E 00 30 00 2E 00 31 00 2E 00 64 00 6D 00 67 00 00 00 17 00 4D 00 61 .5...0...1...d.m.g.....M.a
208 00 63 00 46 00 6F 00 72 00 65 00 6E 00 73 00 69 00 63 00 73 00 43 00 72 00 61 .c.F.o.r.e.n.s.i.c.s.C.r.a
234 00 69 00 67 00 65 00 72 00 2E 00 70 00 64 00 66 70 74 62 4C 75 73 74 72 00 00 .i.g.e.r...p.d.fptbLustr..
260 00 1C 00 55 00 73 00 65 00 72 00 73 00 2F 00 65 00 6C 00 77 00 6F 00 6F 00 64 . .U.s.e.r.s./.e.l.w.o.o.d
286 00 62 00 6C 00 75 00 65 00 73 00 2F 00 44 00 6F 00 77 00 6E 00 6C 00 6F 00 61 .b.l.u.e.s./.D.o.w.n.l.o.a
312 00 64 00 73 00 2F 00 00 00 17 00 4D 00 61 00 63 00 46 00 6F 00 72 00 65 00 6E .d.s./.....M.a.c.F.o.r.e.n



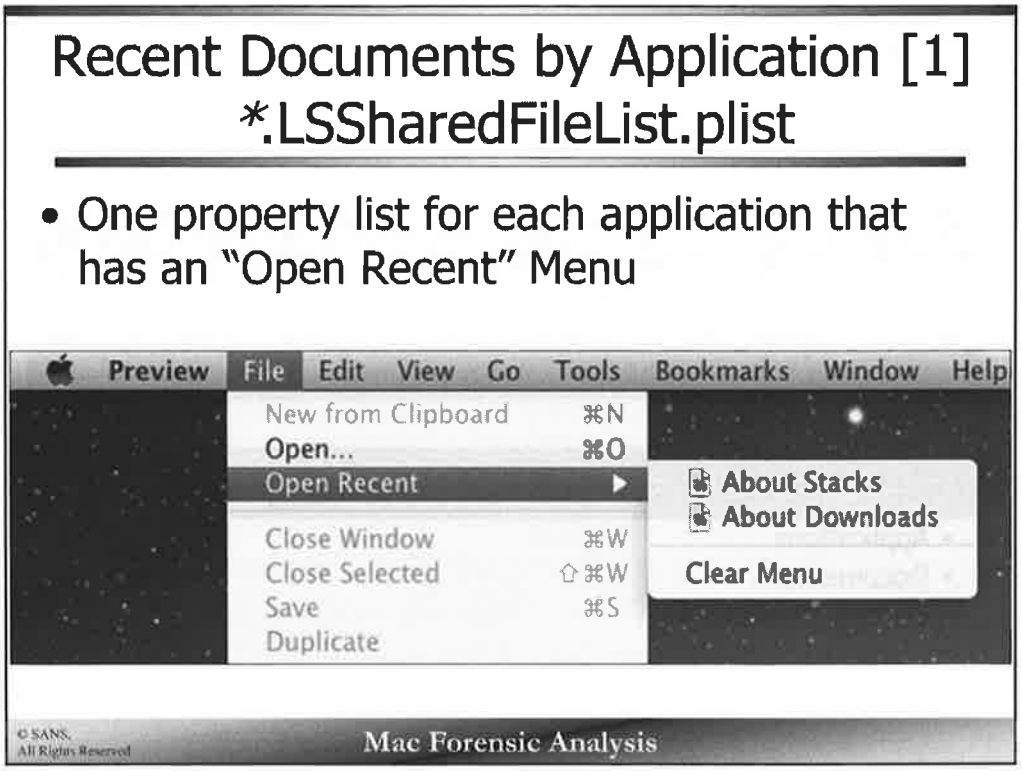
If a user is more computer savvy, they may use SSH to connect to other systems. The SSH `known_hosts` file contains a hostname and/or IP address and a public key. This file is not a definitive way to show all systems that a user connected to, but will contain those that have a saved public key.

By default the hostnames and IP address should be human readable. If the user has configured their `/etc/ssh/ssh_config` file to set `HashKnownHosts` to yes, this data will be hashed.



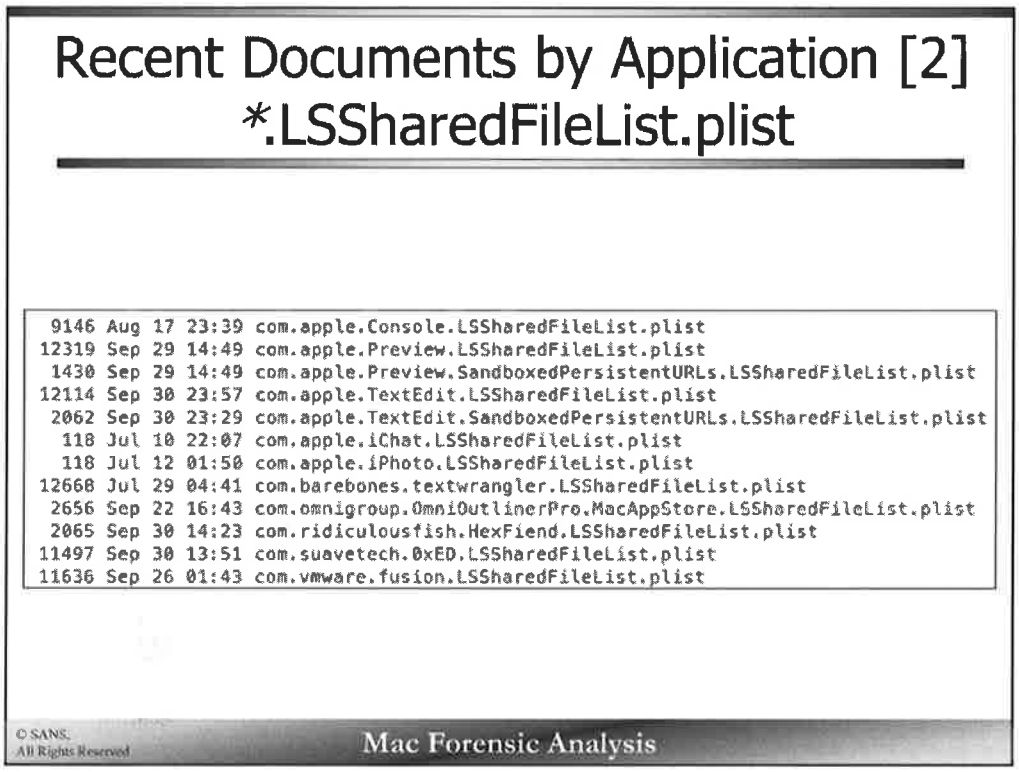
As with Windows OSs, Mac also has many Most Recently Used information.

- The `LSSharedFileList` property lists have the most recently used files per application.
- The `com.apple.finder.plist` has the most recent folders.
- The `com.apple.recentitems.plist` holds the most recent applications, documents, servers, and hosts.



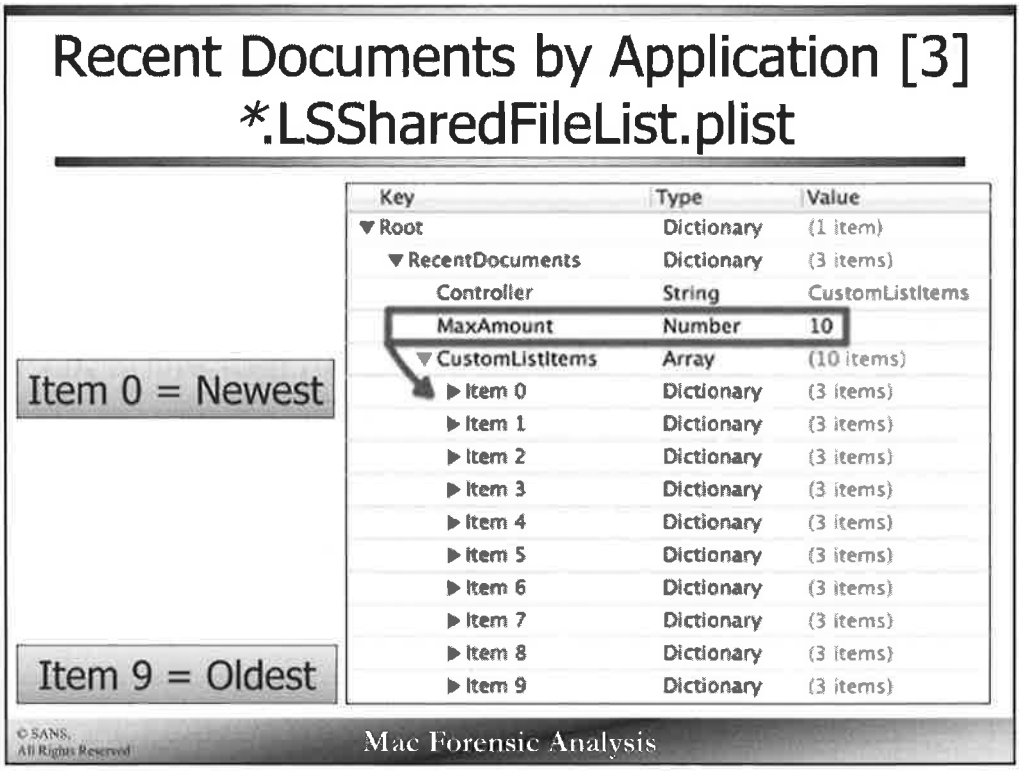
The `~/Library/Preferences` directory contains many `*.LSSharedFileLists.plist` property lists - each associated with an application.

These lists can be viewed in the `File | Open Recent` menu shown in the screenshot above for each specific application. The example shown above is for the Preview application.



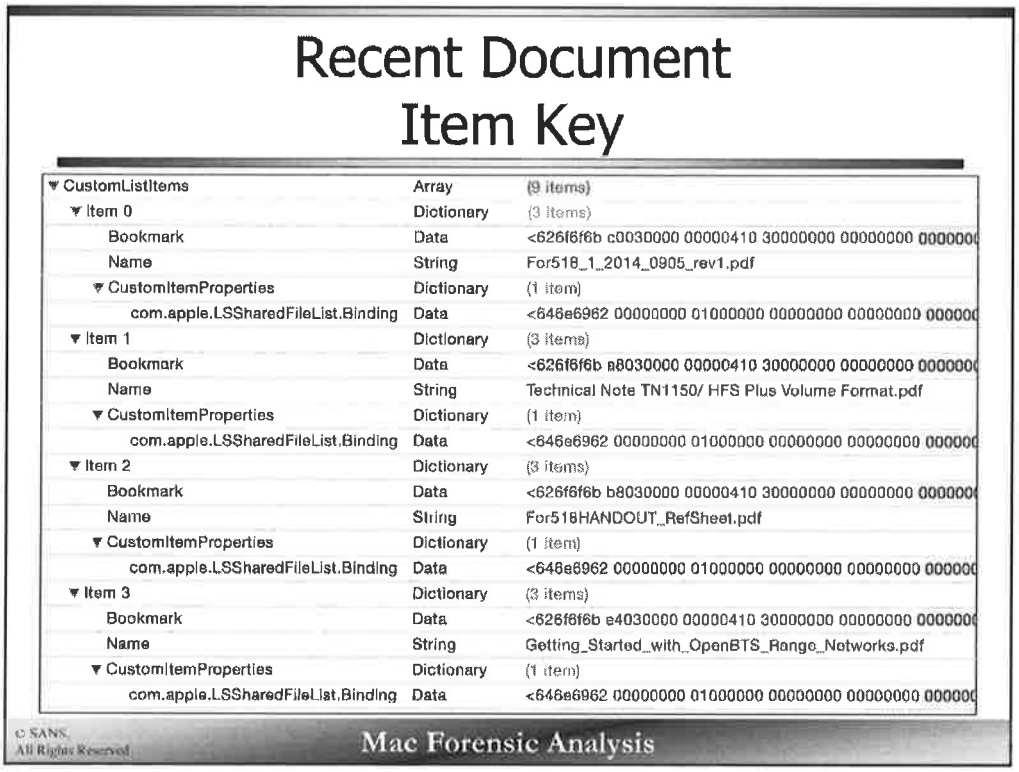
Each application has its own `LSSharedFileList` property list.

The existence of a particular application property list file means the application was used by the user at one point in time.



Each `LSSharedFileList` property list contains a key, `MaxAmount`, that contains the maximum amount of recent documents. The default for most applications is 10.

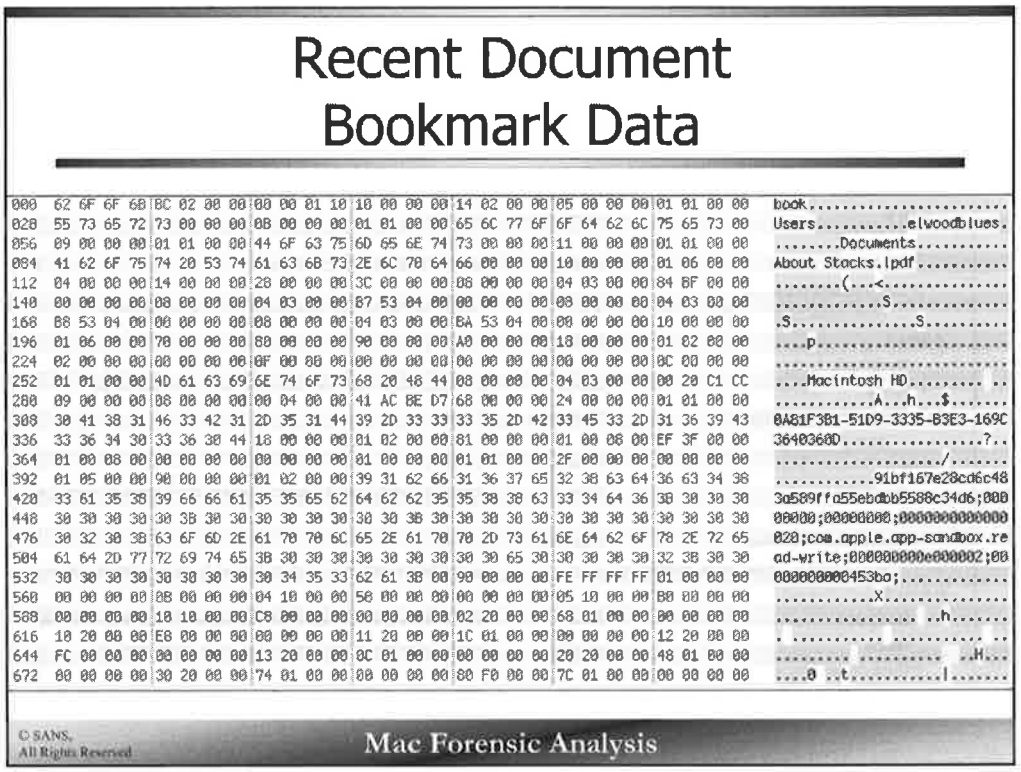
Generally, these MRU lists will have the newest item as Item 0 and older items going to Item 9 (default).



Each `Item` key contains the following keys. Each of these keys provides more information for each recent document.

- Bookmark Data
- Document Name
- Icon/Binding Information

The `Icon/Binding` data may let us determine what icon the file is using, in this case PDF. The document name may not be enough information to find the document if it has been deleted. Extracting the `Bookmark` data reveals more data, shown on the next slide.



The `Bookmark` data can be extracted and viewed in a hex editor. A `Bookmark` is a reference mechanism that is used to find a file or directory, similar to Windows Link files.

Each bookmark starts off with the header "`book`".

In the screenshot above various items of interest can be viewed:

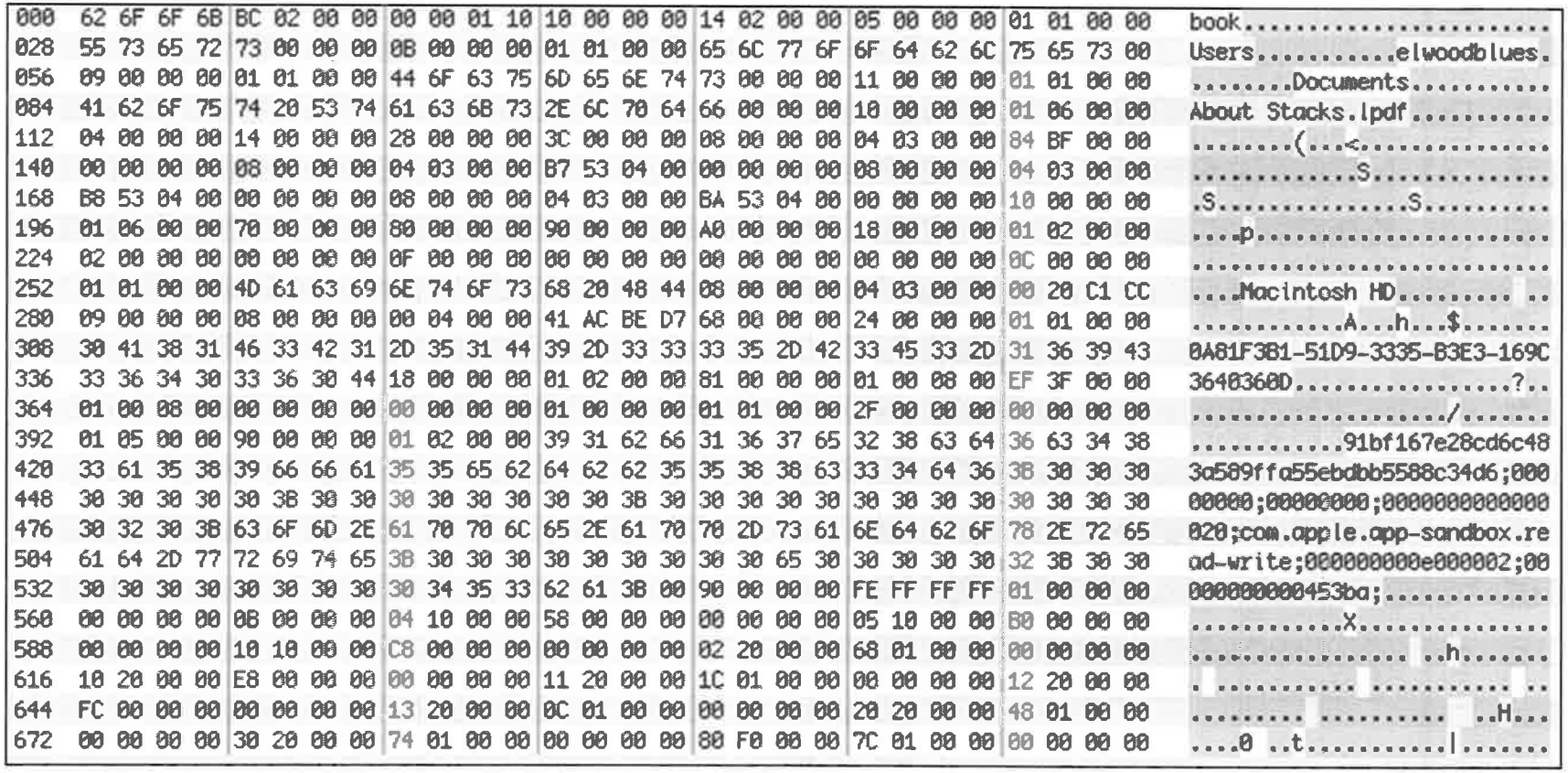
- Strings making up the file path (`/Users/elwoodblues/Document/About Stacks.lpdf`)
- Volume Name (`Macintosh HD`)
- Volume GUID (`0A81F3B1-51D9-3335-B3E3-169C-3640360D`)
- Apple Sandbox data (`91bf167….com.apple.app-sanbox.read-write…`)

On newer systems, this Bookmark data may also contain PNG icons that can be extracted.

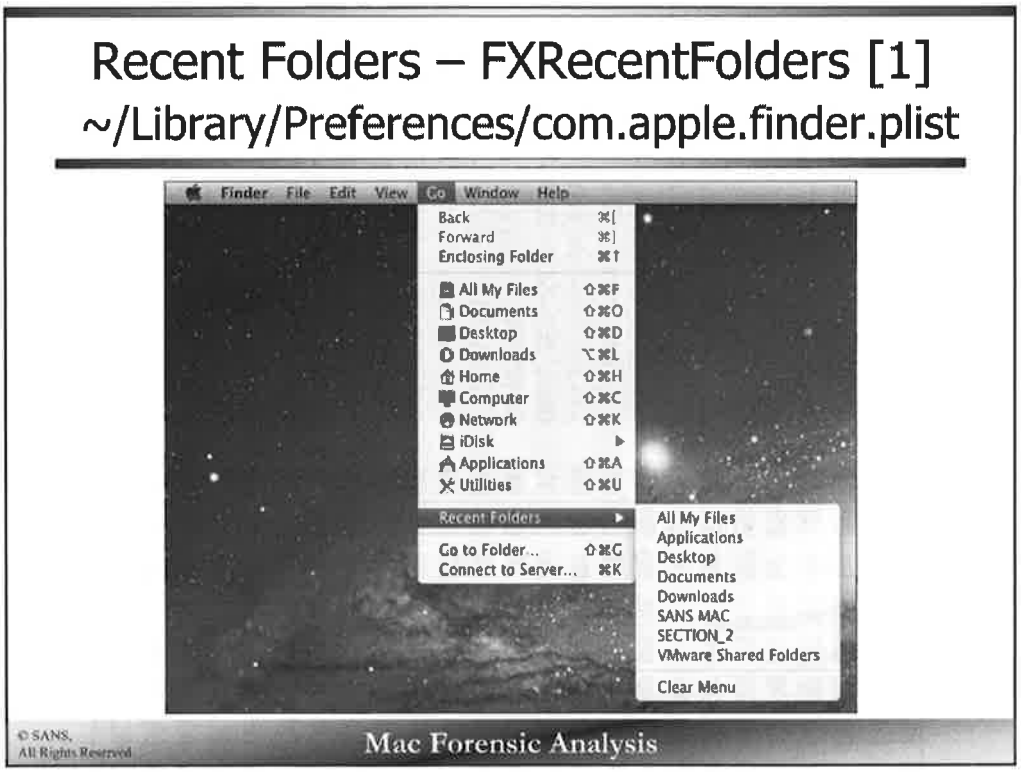
References:

CFURL -

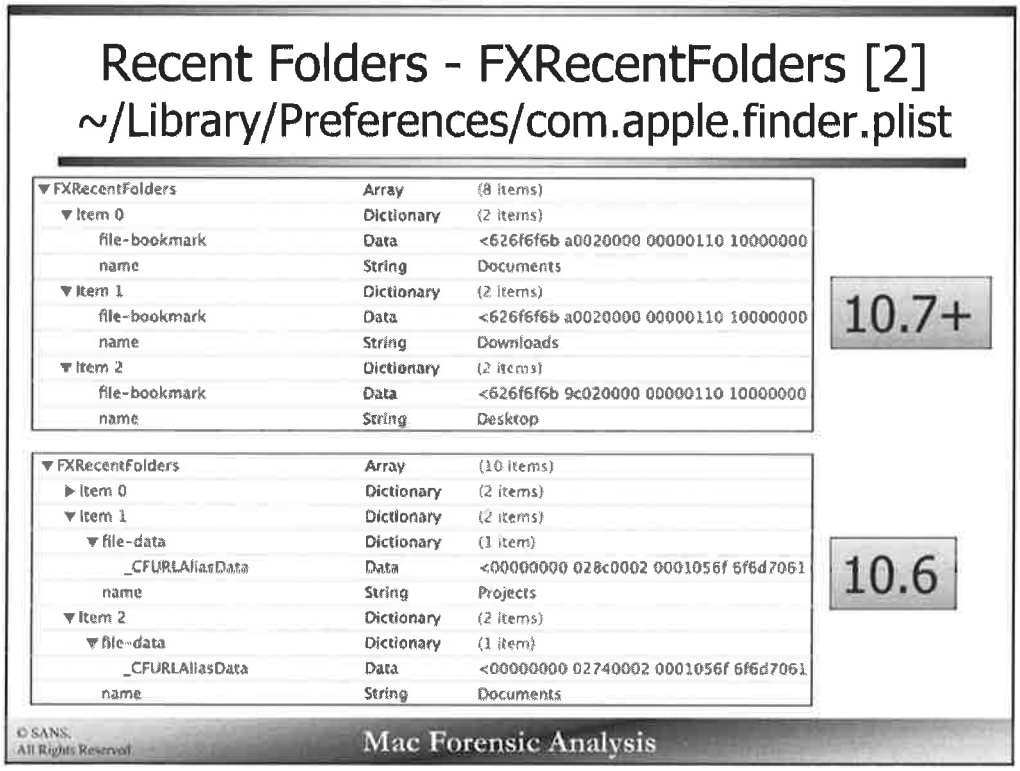
https://developer.apple.com/library/mac/#documentation/CoreFOundation/Reference/CFURLRef/Reference/reference.html



```
000  62 6F 6F 6B BC 02 00 00 00 00 01 10 10 00 00 00 14 02 00 00 05 00 00 00 01 01 00 00  book............................
028  55 73 65 72 73 00 00 00 0B 00 00 00 01 01 00 00 65 6C 77 6F 6F 64 62 6C 75 65 73 00  Users...........elwoodblues.
056  09 00 00 00 01 01 00 00 44 6F 63 75 6D 65 6E 74 73 00 00 00 11 00 00 00 01 01 00 00  ........Documents...........
084  41 62 6F 75 74 20 53 74 61 63 6B 73 2E 6C 70 64 66 00 00 00 10 00 00 00 01 06 00 00  About Stacks.lpdf...........
112  04 00 00 00 14 00 00 00 28 00 00 00 3C 00 00 00 08 00 00 00 04 03 00 00 84 BF 00 00  ........(...<...............
140  00 00 00 00 08 00 00 00 04 03 00 00 B7 53 04 00 00 00 00 00 08 00 00 00 04 03 00 00  ..............S.............
168  B8 53 04 00 00 00 00 00 08 00 00 00 04 03 00 00 BA 53 04 00 00 00 00 00 10 00 00 00  .S..............S...........
196  01 06 00 00 70 00 00 00 80 00 00 00 90 00 00 00 A0 00 00 00 18 00 00 00 01 02 00 00  ....p...........................
224  02 00 00 00 00 00 00 00 0F 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 0C 00 00 00  ................................
252  01 01 00 00 4D 61 63 69 6E 74 6F 73 68 20 48 44 08 00 00 00 04 03 00 00 00 20 C1 CC  ....Macintosh HD..... ..
280  09 00 00 00 08 00 00 00 00 04 00 00 41 AC BE D7 68 00 00 00 24 00 00 00 01 01 00 00  ............A...h...$.......
308  30 41 38 31 46 33 42 31 2D 35 31 44 39 2D 33 33 33 35 2D 42 33 45 33 2D 31 36 39 43  0A81F3B1-51D9-3335-B3E3-169C
336  33 36 34 30 33 36 30 44 18 00 00 00 01 02 00 00 81 00 00 00 01 00 08 00 EF 3F 00 00  36403600D..............?..
364  01 00 08 00 00 00 00 00 00 00 00 00 01 00 00 00 01 01 00 00 2F 00 00 00 00 00 00 00  ................/.......
392  01 05 00 00 90 00 00 00 01 02 00 00 39 31 62 66 31 36 37 65 32 38 63 64 36 63 34 38  ............91bf167e28cd6c48
420  33 61 35 38 39 66 66 61 35 35 65 62 64 62 62 35 35 38 38 63 33 34 64 36 3B 30 30 30  3a589ffa55ebdbb5588c34d6;000
448  30 30 30 30 30 3B 30 30 30 30 30 30 3B 30 30 30 30 30 30 30 30 30 30 30 30 30 30 30  00000;00000000;0000000000000
476  30 32 30 3B 63 6F 6D 2E 61 70 70 6C 65 2E 61 70 70 2D 73 61 6E 64 62 6F 78 2E 72 65  020;com.apple.app-sandbox.re
504  61 64 2D 77 72 69 74 65 3B 30 30 30 30 30 30 30 30 30 30 65 30 30 30 30 30 32 3B 30 30  ad-write;000000000e000002;00
532  30 30 30 30 30 30 30 30 30 34 35 33 62 61 3B 00 90 00 00 00 FE FF FF FF 01 00 00 00  0000000000453ba;............
560  00 00 00 00 0B 00 00 00 04 10 00 00 58 00 00 00 00 00 00 00 05 10 00 00 B0 00 00 00  ............X...............
588  00 00 00 00 10 10 00 00 C8 00 00 00 00 00 00 00 02 20 00 00 68 01 00 00 00 00 00 00  ..................... ..h.......
616  10 20 00 00 E8 00 00 00 00 00 00 00 11 20 00 00 1C 01 00 00 00 00 00 00 12 20 00 00  . ........... ........... ..
644  FC 00 00 00 00 00 00 00 13 20 00 00 0C 01 00 00 00 00 00 00 20 20 00 00 48 01 00 00  ......... ..........  ..H...
672  00 00 00 00 30 20 00 00 74 01 00 00 00 00 00 00 80 F0 00 00 7C 01 00 00 00 00 00 00  ....0 ..t...........|.......
```



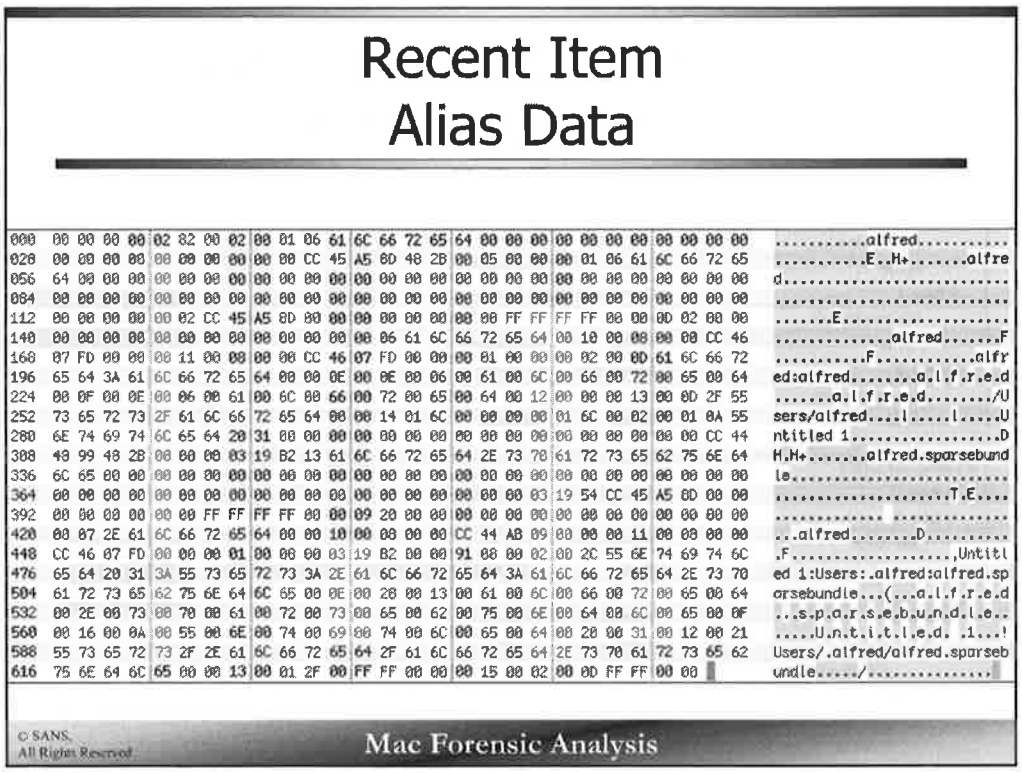
The Recent Folders are located in the `com.apple.finder.plist` in the `Preferences` directory.



Shown above are two examples of the `com.apple.finder.plist` file. Each recent item contains a folder name and a reference link.

On top is an example from a 10.7+ system. The `file-bookmark` is a file reference using the same bookmark format as before.

The bottom screenshot is an example from a 10.6 system, the file reference `_CFURLAliasData` uses another format called alias data, rather than the bookmark format.

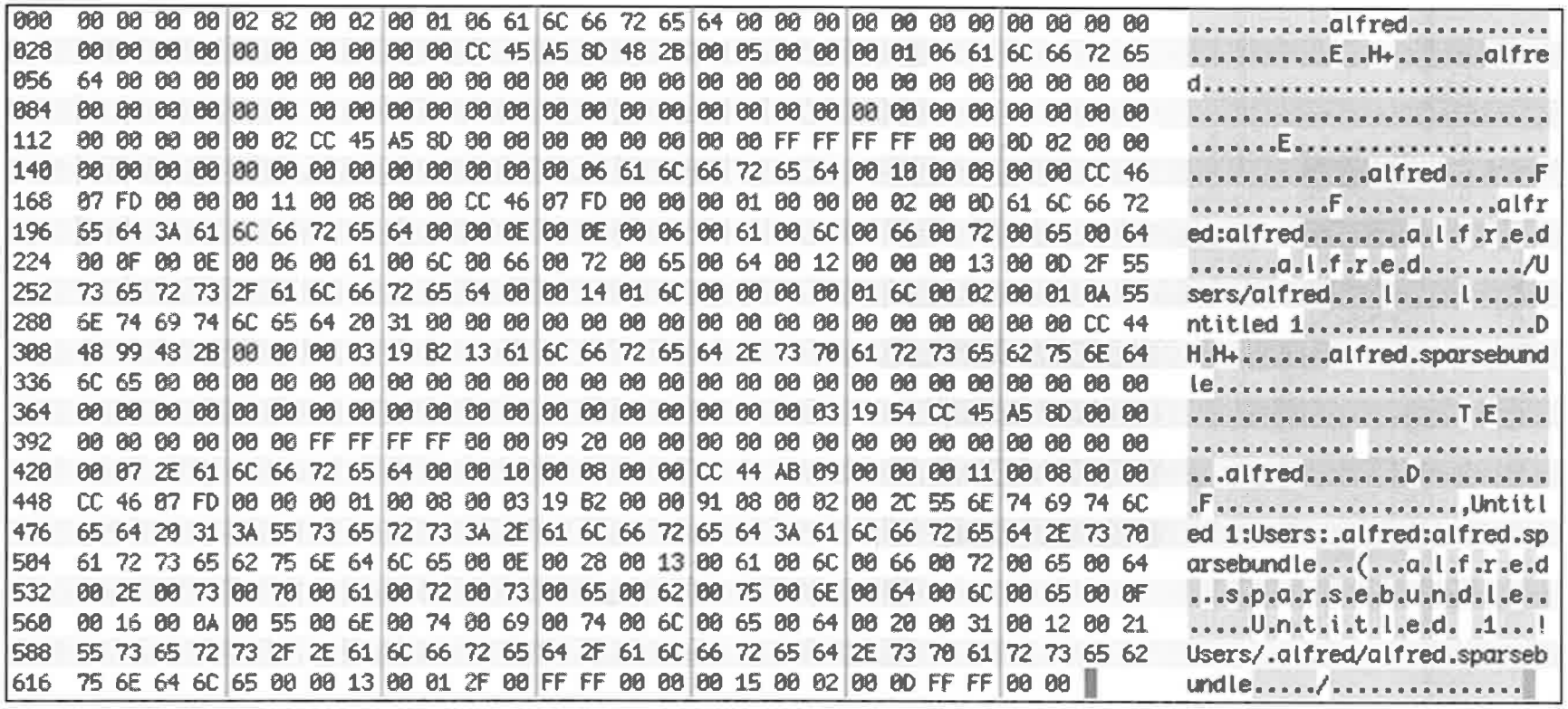


Using the same method we can extract the alias data. It contains many of the same file pointer data found in a file bookmark, as shown in the screenshot above.

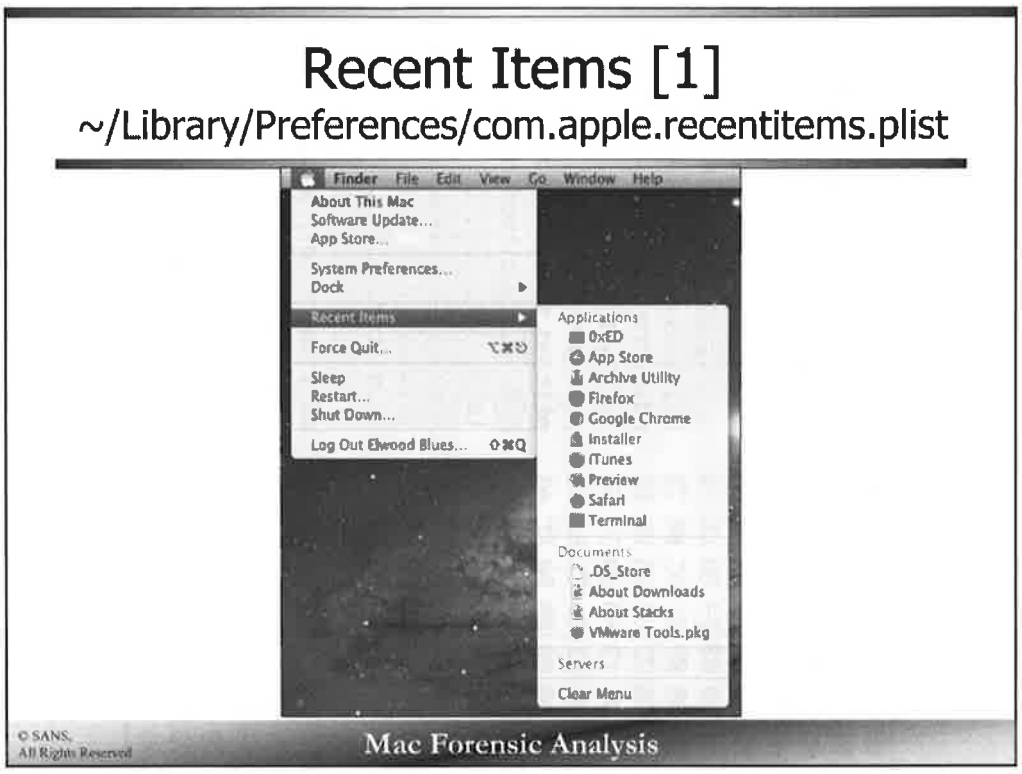
- File Path
- Volume Information

In the screenshot above various items of interest can be viewed:

- Strings making up the file path (`/Users/alfred/alfred.sparsebundle`)
- Volume Name (`Untitled 1`)
- Volume Format - A string that shows the format of the volume, in this case "`H+`", is found on the second line.



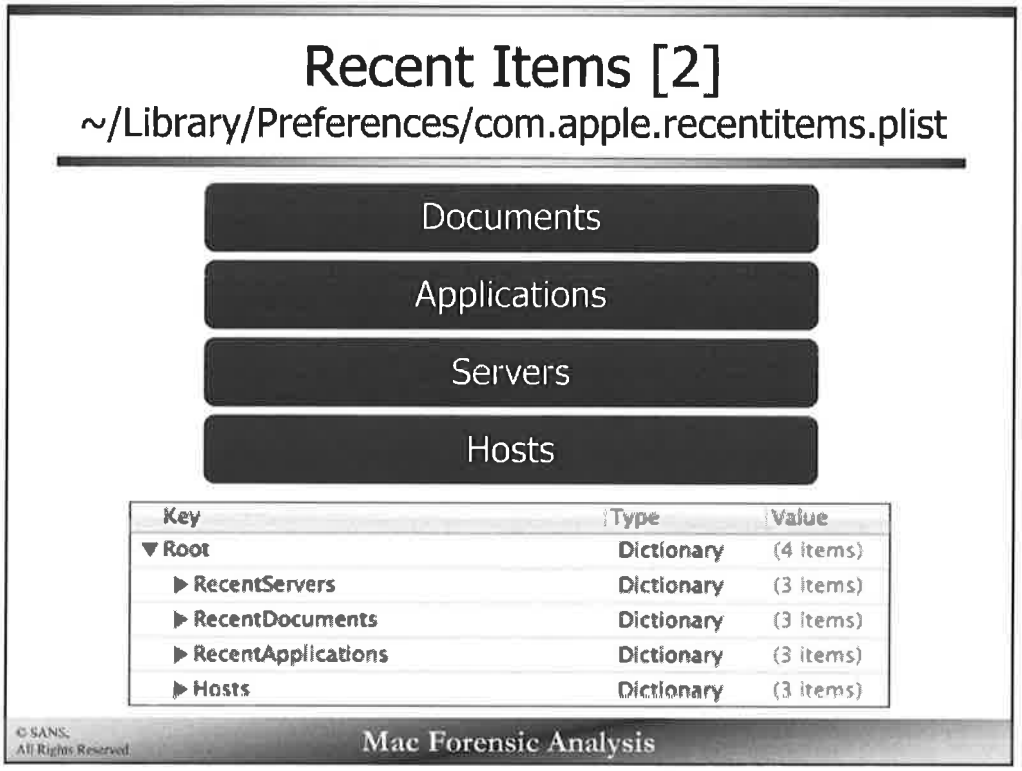
```
000  00 00 00 00 02 82 00 02 00 01 06 61 6C 66 72 65 64 00 00 00 00 00 00 00 00 00 00 00  ...........alfred...........
028  00 00 00 00 00 00 00 00 00 00 CC 45 A5 8D 48 2B 00 05 00 00 00 01 06 61 6C 66 72 65  ...........E..H+.......alfre
056  64 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  d...........................
084  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ...........................
112  00 00 00 00 00 02 CC 45 A5 8D 00 00 00 00 00 00 00 00 FF FF FF FF 00 00 0D 02 00 00  .......E....................
140  00 00 00 00 00 00 00 00 00 00 00 00 00 06 61 6C 66 72 65 64 00 18 00 08 00 00 CC 46  ..............alfred.......F
168  07 FD 00 00 00 11 00 08 00 00 CC 46 07 FD 00 00 00 01 00 00 00 02 00 0D 61 6C 66 72  ...........F............alfr
196  65 64 3A 61 6C 66 72 65 64 00 00 0E 00 0E 00 06 00 61 00 6C 00 66 00 72 00 65 00 64  ed:alfred........a.l.f.r.e.d
224  00 0F 00 0E 00 06 00 61 00 6C 00 66 00 72 00 65 00 64 00 12 00 00 00 13 00 0D 2F 55  .......a.l.f.r.e.d........./U
252  73 65 72 73 2F 61 6C 66 72 65 64 00 00 14 01 6C 00 00 00 00 01 6C 00 02 00 01 0A 55  sers/alfred....l.....l.....U
280  6E 74 69 74 6C 65 64 20 31 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 CC 44  ntitled 1..................D
308  48 99 48 2B 00 00 00 03 19 B2 13 61 6C 66 72 65 64 2E 73 70 61 72 73 65 62 75 6E 64  H.H+.......alfred.sparsebund
336  6C 65 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00  le..........................
364  00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 00 03 19 54 CC 45 A5 8D 00  ......................T.E....
392  00 00 00 00 00 00 FF FF FF FF 00 00 09 20 00 00 00 00 00 00 00 00 00 00 00 00 00 00  ..............  ..............
420  00 07 2E 61 6C 66 72 65 64 00 00 10 00 08 00 00 CC 44 AB 09 00 00 00 11 00 08 00 00  ...alfred........D..........
448  CC 46 07 FD 00 00 00 01 00 08 00 03 19 B2 00 00 91 08 00 02 00 2C 55 6E 74 69 74 6C  .F...................,Untitl
476  65 64 20 31 3A 55 73 65 72 73 3A 2E 61 6C 66 72 65 64 3A 61 6C 66 72 65 64 2E 73 70  ed 1:Users:.alfred:alfred.sp
504  61 72 73 65 62 75 6E 64 6C 65 00 0E 00 28 00 13 00 61 00 6C 00 66 00 72 00 65 00 64  arsebundle...(...a.l.f.r.e.d
532  00 2E 00 73 00 70 00 61 00 72 00 73 00 65 00 62 00 75 00 6E 00 64 00 6C 00 65 00 0F  ...s.p.a.r.s.e.b.u.n.d.l.e..
560  00 16 00 0A 00 55 00 6E 00 74 00 69 00 74 00 6C 00 65 00 64 00 20 00 31 00 12 00 21  .....U.n.t.i.t.l.e.d. .1...!
588  55 73 65 72 73 2F 2E 61 6C 66 72 65 64 2F 61 6C 66 72 65 64 2E 73 70 61 72 73 65 62  Users/.alfred/alfred.sparseb
616  75 6E 64 6C 65 00 00 13 00 01 2F 00 FF FF 00 00 00 15 00 02 00 0D FF FF 00 00 [redacted]  undle...../..............[redacted]
```



The Apple menu in the menu bar has a list of recent items used for each of the following:

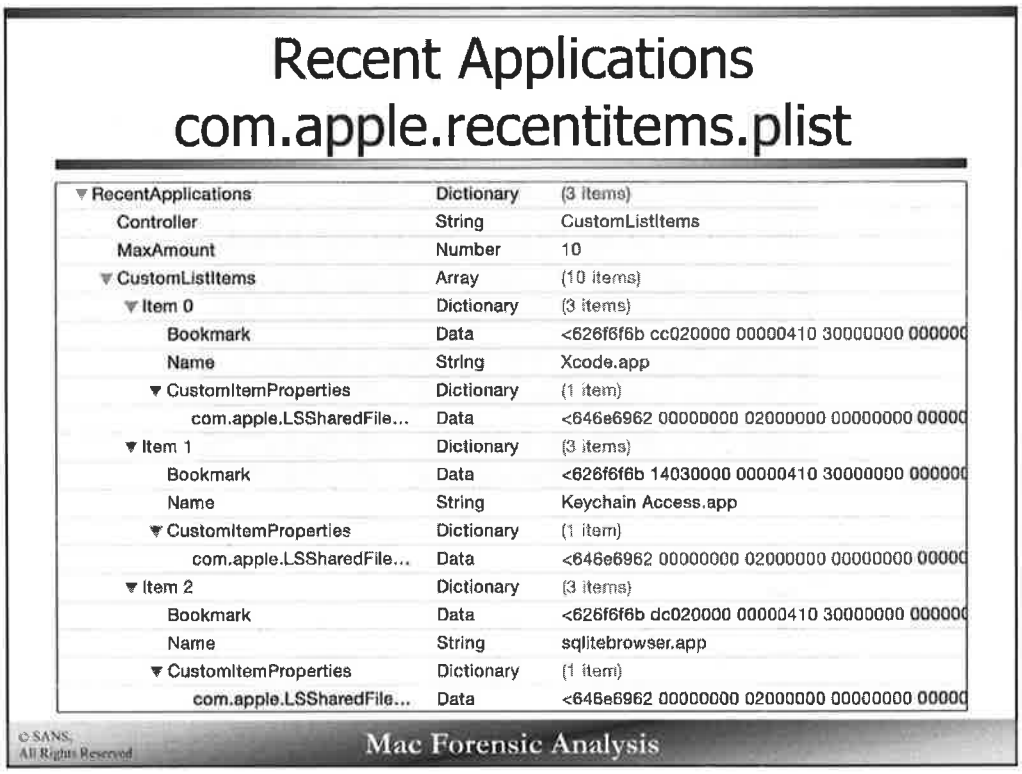
- Applications
- Documents
- Servers

The property list file `com.apple.recentitems.plist` holds this data and also includes recent hosts access via the `Go | Connect to Server` menu.



The `com.apple.recentitems.plist` file is organized by each area:

- Servers
- Documents
- Applications
- Hosts



The `Recent Applications` key contains the most recently used applications, the default is 10 applications.

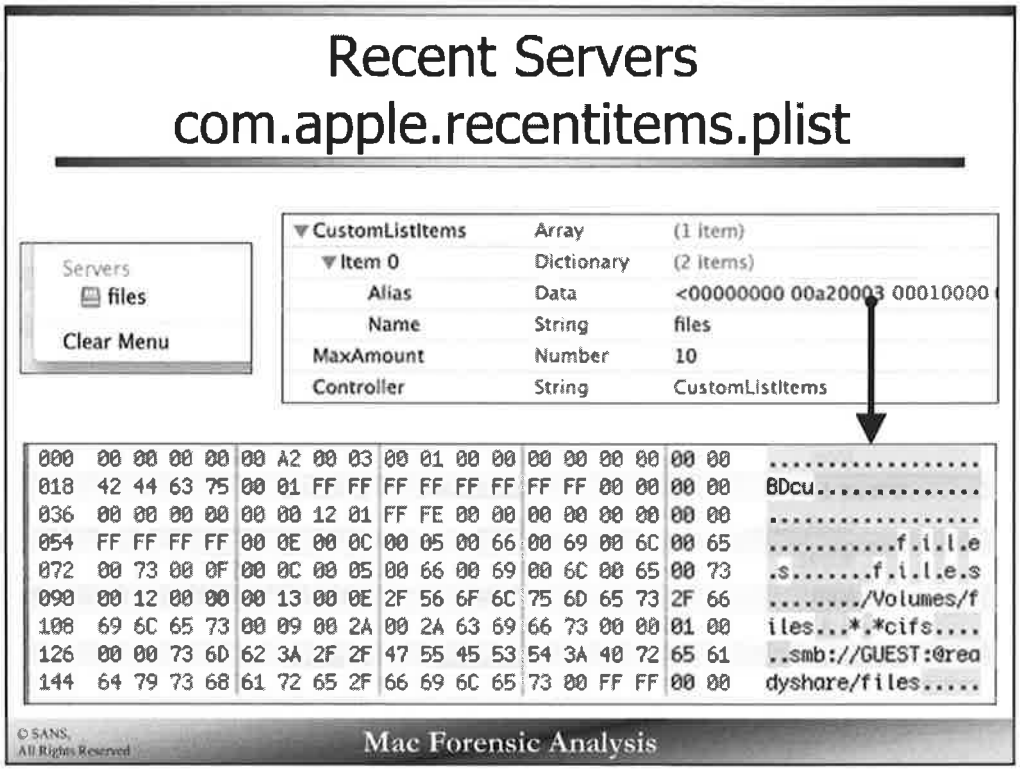
As with previous examples, the bookmark data holds the contents of a file or directory reference. The name and file icon are also included.

Newer systems use `com.apple.LSSharedFileList.binding` to potentially hold a file's application binding, similar to the Icon key that older systems use.

Each icon key is a binary version of the application's icon. This binary data can be extracted and a new file created to view the icon. If you are not familiar with a specific data file, this could be used to determine which application this file was used with.

The `Recent Documents` section holds the most recently accessed documents, the default is 10 documents.

On older systems, an Icon key is available. If the `Icon` data is extracted, it will show an icon of the type of file it is. For example the '`About Stacks`' file will have a PDF Preview icon. Preview is the Apple PDF/Image viewing application.

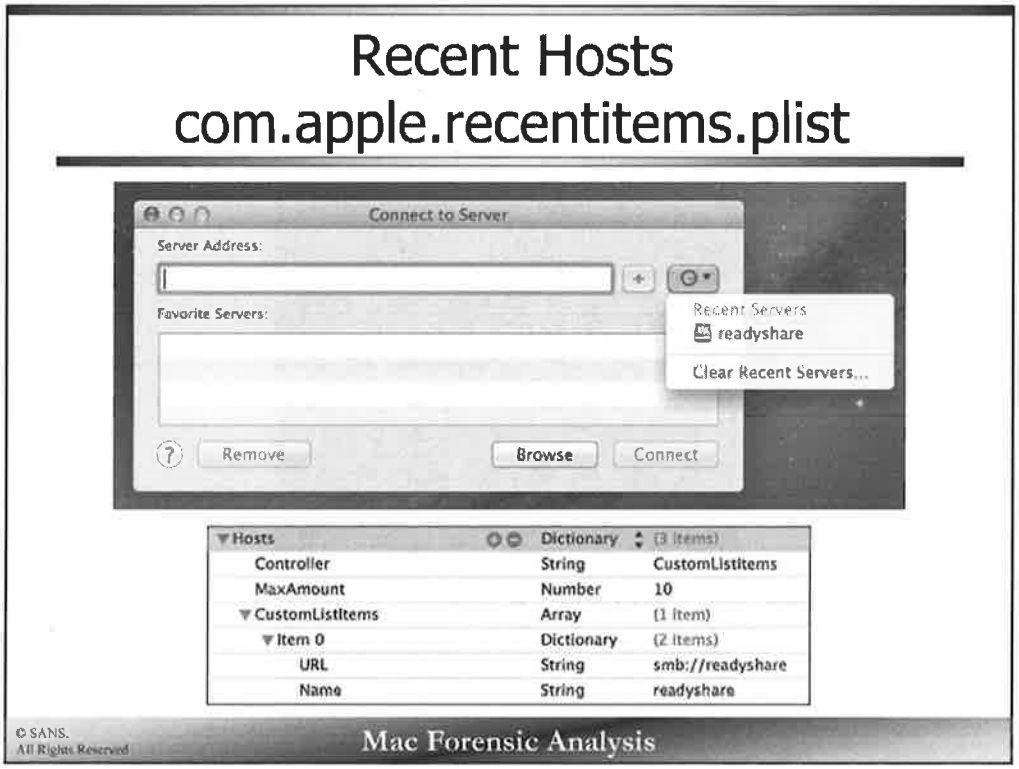


The `Recent Servers` section holds the most recently accessed servers, the default is 10 servers. The data in the `Item` key uses alias data, rather than bookmark data. The alias data can be extracted to view:

- Format of the server volume (This will be discussed in future sections).
- Mount Point (`/Volumes/files`)
- Network location and protocol (`smb://GUEST:@readyshare/files`)

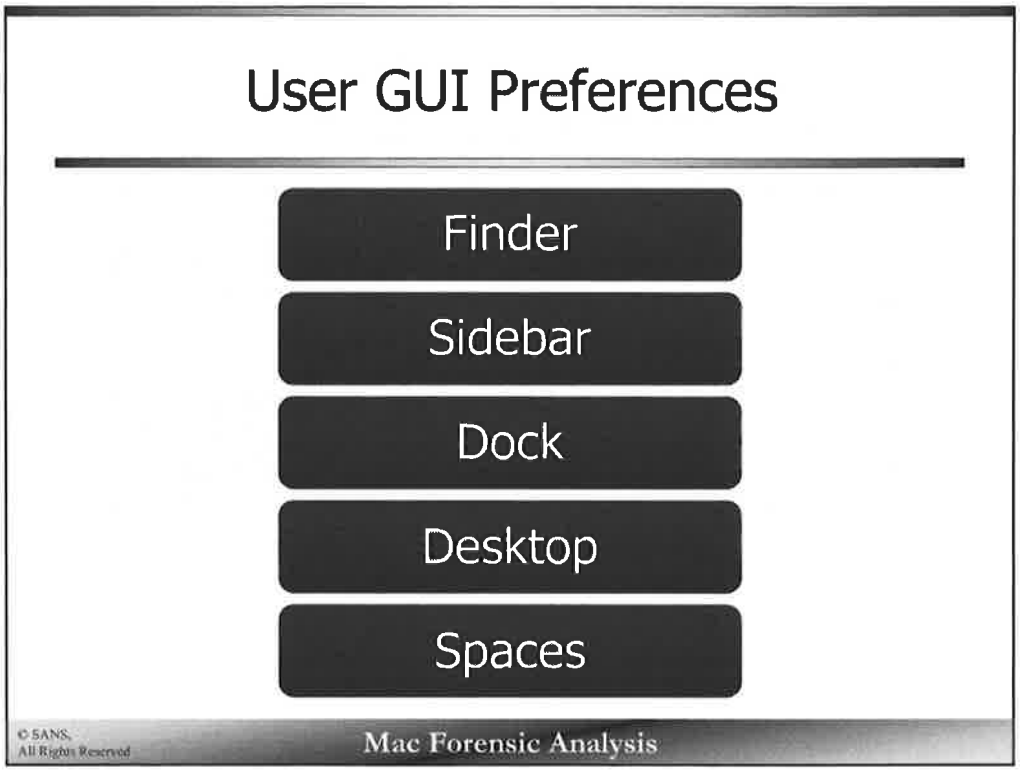
`Recent Servers` will likely be shared directories such as a file share, while recent `Hosts` will contain server specific connections rather than a specific directory.

BDcu

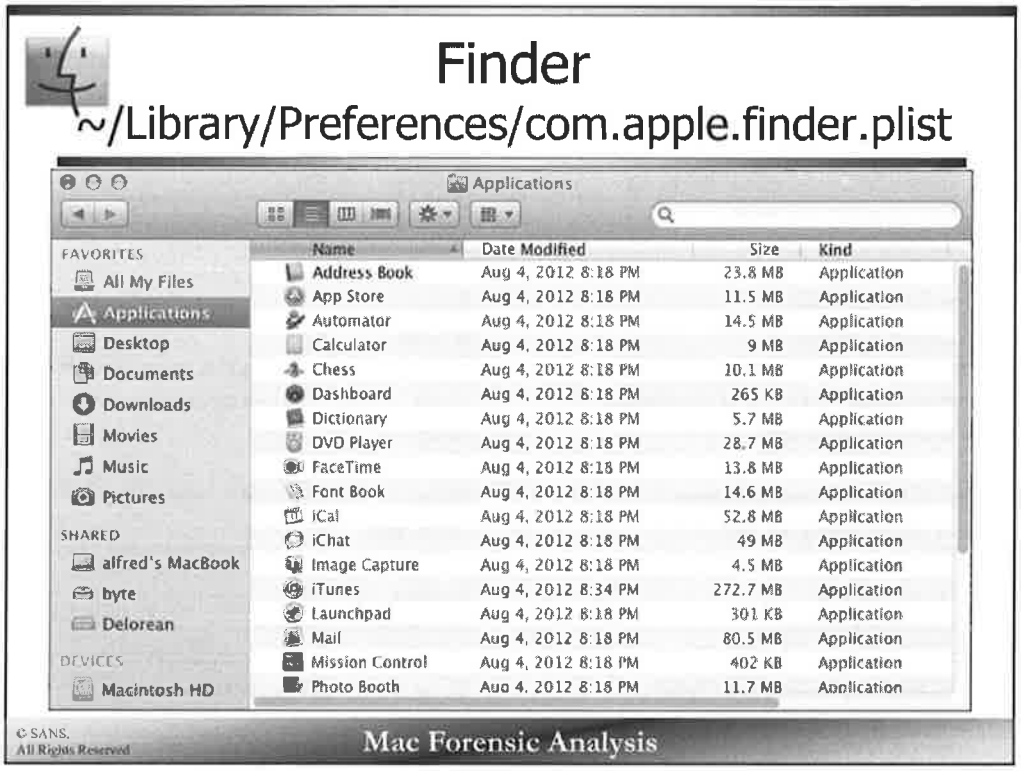


The `Hosts` section holds the most recently accessed hosts, the default is 10 hosts. Recent hosts can be viewed from the `Go | Connect to Server` menu in Finder.

The `Item` data will contain a name and a URL of the share link (`smb://readyshare`)



A user's workspace can say many things about a person. What applications are likely to be used more often, how organized is the user, or what directories are important to them. The OS X interface has many entities that can give you a peek into how a person uses their computer.



The Apple Finder is comparable to the Windows Explorer, it is the main application allowing access to applications, files, directories, networks, and other media.

On the left sidebar, the finder categories are shown, these of course are configurable by the user.

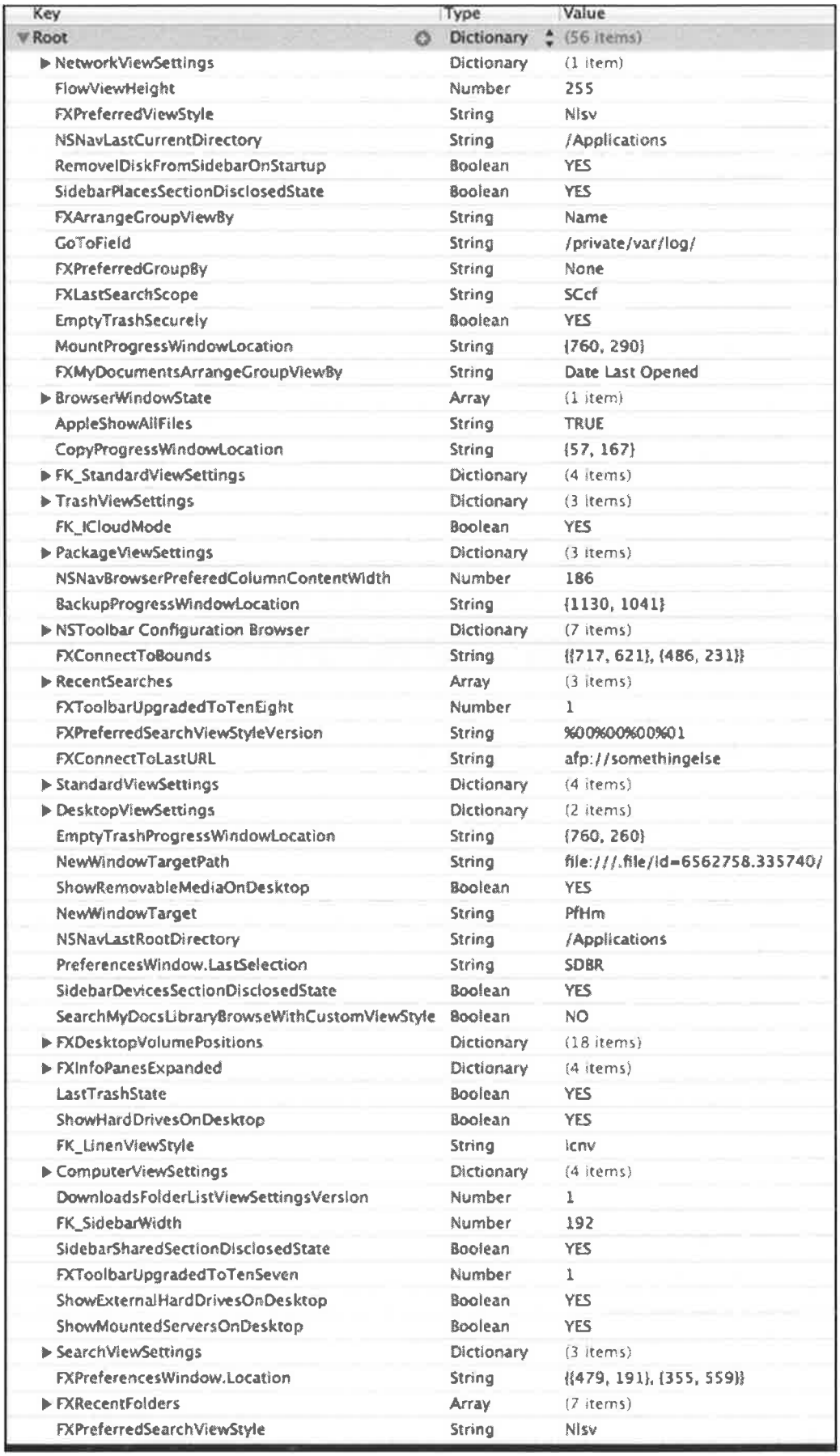
- Favorites – By default the user directories are shown, such as Documents, Pictures, Music, Downloads, etc.
- Shared – These are the other devices on the network than can be used for either file storage, other computers that can be "screen shared", etc.
- Devices – Contains the volumes the system has access to; the host volume (Macintosh HD), USB drives, CDs/DVDs, FireWire, Thunderbolt, mounted DMGs, etc.

Amongst the preference keys that will be covered, the `com.apple.finder.plist` contains a huge amount of user preference data. Almost every configurable item of the Finder can be found in this plist.

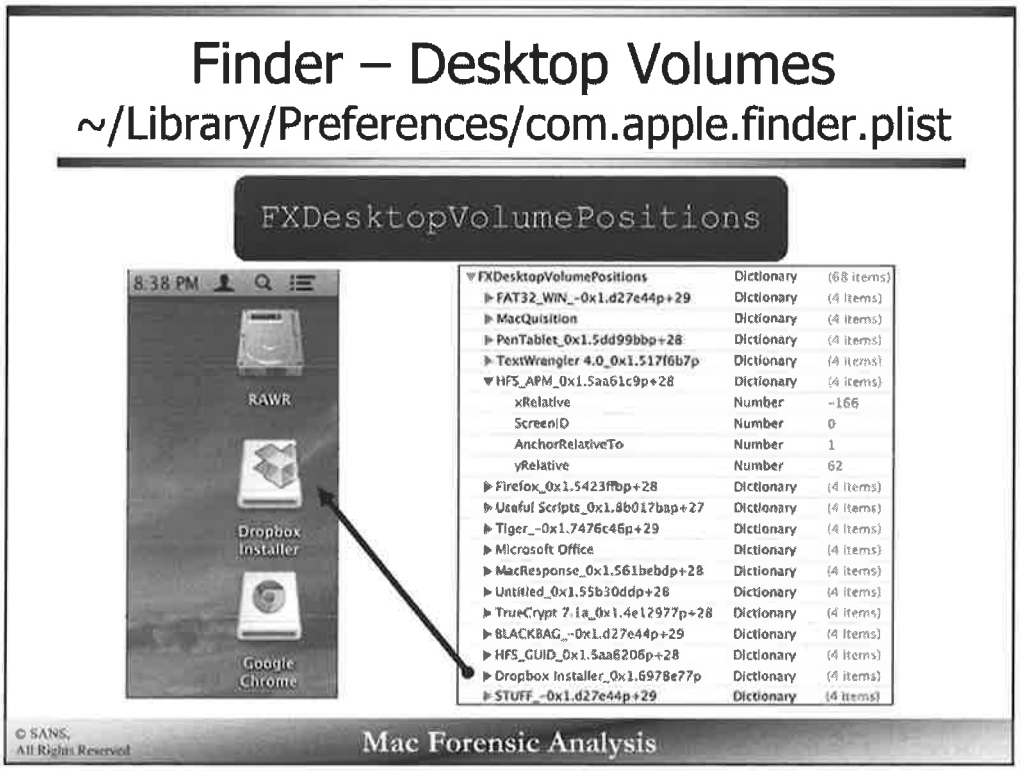
For example:

- Show Mounted Servers
- Show Mounted Hard Drives
- Column Preference
- Empty Trash Securely
- X & Y Coordinates of different GUI features

Note: Each version of OS X may have different property list keys, though the notable ones are the same. 10.6 does not have nearly as many configuration details as systems 10.7+ (see the next page).



Key	Type	Value
▼ Root	Dictionary	(56 items)
▶ NetworkViewSettings	Dictionary	(1 item)
FlowViewHeight	Number	255
FXPreferredViewStyle	String	Nlsv
NSNavLastCurrentDirectory	String	/Applications
RemoveIDiskFromSidebarOnStartup	Boolean	YES
SidebarPlacesSectionDisclosedState	Boolean	YES
FXArrangeGroupViewBy	String	Name
GoToField	String	/private/var/log/
FXPreferredGroupBy	String	None
FXLastSearchScope	String	SCcf
EmptyTrashSecurely	Boolean	YES
MountProgressWindowLocation	String	{760, 290}
FXMyDocumentsArrangeGroupViewBy	String	Date Last Opened
▶ BrowserWindowState	Array	(1 item)
AppleShowAllFiles	String	TRUE
CopyProgressWindowLocation	String	{57, 167}
▶ FK_StandardViewSettings	Dictionary	(4 items)
▶ TrashViewSettings	Dictionary	(3 items)
FK_ICloudMode	Boolean	YES
▶ PackageViewSettings	Dictionary	(3 items)
NSNavBrowserPreferedColumnContentWidth	Number	186
BackupProgressWindowLocation	String	{1130, 1041}
▶ NSToolbar Configuration Browser	Dictionary	(7 items)
FXConnectToBounds	String	{{717, 621}, {486, 231}}
▶ RecentSearches	Array	(3 items)
FXToolbarUpgradedToTenEight	Number	1
FXPreferredSearchViewStyleVersion	String	%00%00%00%01
FXConnectToLastURL	String	afp://somethingelse
▶ StandardViewSettings	Dictionary	(4 items)
▶ DesktopViewSettings	Dictionary	(2 items)
EmptyTrashProgressWindowLocation	String	{760, 260}
NewWindowTargetPath	String	file:///.file/id=6562758.335740/
ShowRemovableMediaOnDesktop	Boolean	YES
NewWindowTarget	String	PfHm
NSNavLastRootDirectory	String	/Applications
PreferencesWindow.LastSelection	String	SDBR
SidebarDevicesSectionDisclosedState	Boolean	YES
SearchMyDocsLibraryBrowseWithCustomViewStyle	Boolean	NO
▶ FXDesktopVolumePositions	Dictionary	(18 items)
▶ FXInfoPanesExpanded	Dictionary	(4 items)
LastTrashState	Boolean	YES
ShowHardDrivesOnDesktop	Boolean	YES
FK_LinenViewStyle	String	icnv
▶ ComputerViewSettings	Dictionary	(4 items)
DownloadsFolderListViewSettingsVersion	Number	1
FK_SidebarWidth	Number	192
SidebarSharedSectionDisclosedState	Boolean	YES
FXToolbarUpgradedToTenSeven	Number	1
ShowExternalHardDrivesOnDesktop	Boolean	YES
ShowMountedServersOnDesktop	Boolean	YES
▶ SearchViewSettings	Dictionary	(3 items)
FXPreferencesWindow.Location	String	{{479, 191}, {355, 559}}
▶ FXRecentFolders	Array	(7 items)
FXPreferredSearchViewStyle	String	Nlsv



The Finder application stores the mounted volumes in the `com.apple.finder.plist`. (Remember we already discussed how this plist also stores the `Recent Folders` in the `FXRecentFolders` key.)

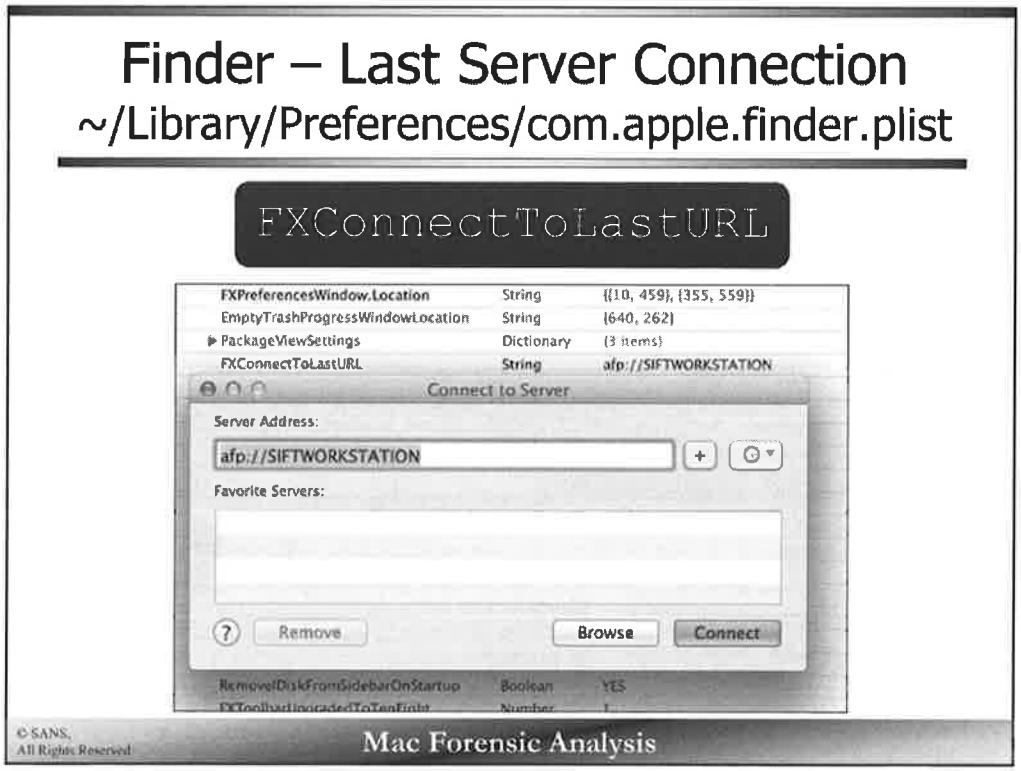
The `FXDesktopVolumePositions` key, mainly stores the `X` and `Y` coordinates of volumes when mounted on the Desktop. This key can be useful for forensic analysts because it stores all the volumes that were mounted by the user. It unfortunately does not store the date when it was last mounted, but we may be able to determine that in Section 3 of this course with volume analysis in the logs. Knowing when a volume is mounted can help an investigator determine when it was in use.

Each volume is listed by it's mounted volume name, some entries have an appended number*, it is unknown what this number represents.

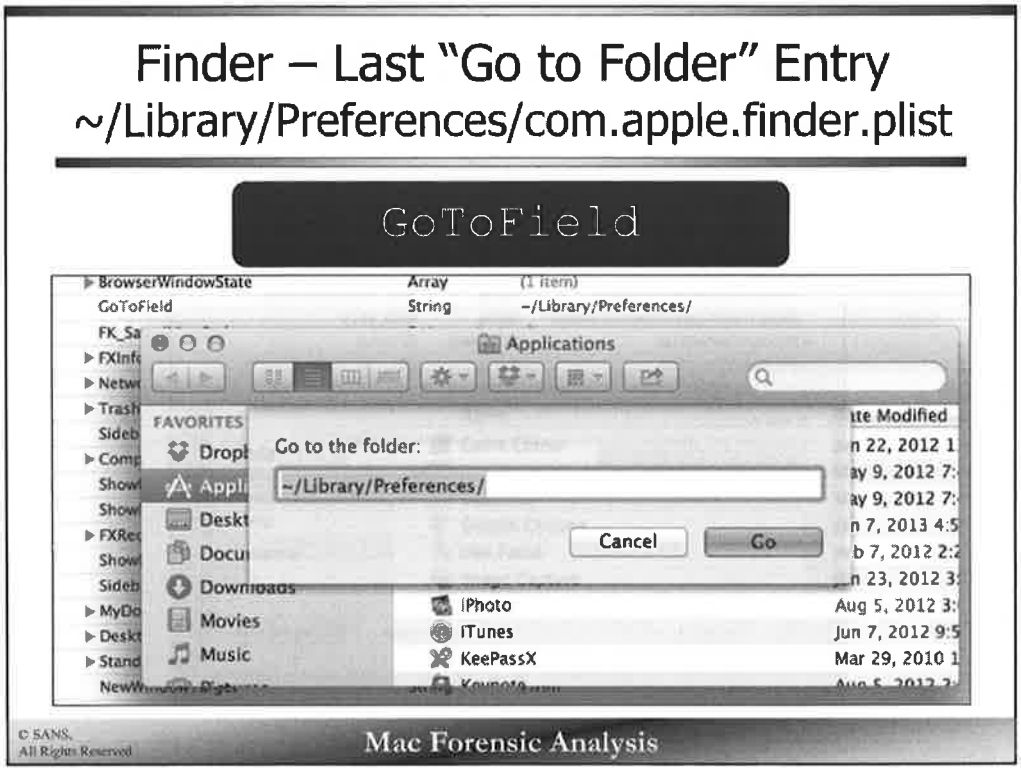
Note: If the user does not have the Finder preferences configured to show items on the desktop, this key will not exist.

*Bonus Points – Determine what significance (insignificance?) this number represents. The author would be eternally grateful.

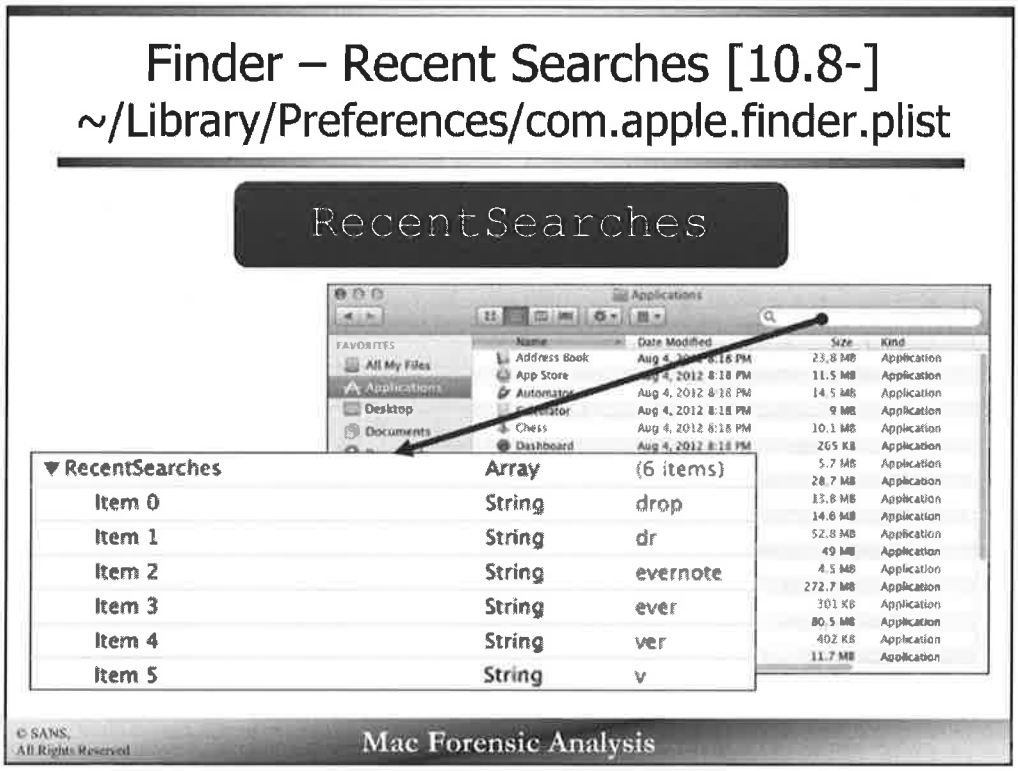
▼ FXDesktopVolumePositions	Dictionary	(68 items)
▶ FAT32_WIN_-0x1.d27e44p+29	Dictionary	(4 items)
▶ MacQuisition	Dictionary	(4 items)
▶ PenTablet_0x1.5dd99bbp+28	Dictionary	(4 items)
▶ TextWrangler 4.0_0x1.517f6b7p	Dictionary	(4 items)
▼ HFS_APM_0x1.5aa61c9p+28	Dictionary	(4 items)
xRelative	Number	-166
ScreenID	Number	0
AnchorRelativeTo	Number	1
yRelative	Number	62
▶ Firefox_0x1.5423ffbp+28	Dictionary	(4 items)
▶ Useful Scripts_0x1.8b017bap+27	Dictionary	(4 items)
▶ Tiger_-0x1.7476c46p+29	Dictionary	(4 items)
▶ Microsoft Office	Dictionary	(4 items)
▶ MacResponse_0x1.561bebdp+28	Dictionary	(4 items)
▶ Untitled_0x1.55b30ddp+28	Dictionary	(4 items)
▶ TrueCrypt 7.1a_0x1.4e12977p+28	Dictionary	(4 items)
▶ BLACKBAG_-0x1.d27e44p+29	Dictionary	(4 items)
▶ HFS_GUID_0x1.5aa6206p+28	Dictionary	(4 items)
▶ Dropbox Installer_0x1.6978e77p	Dictionary	(4 items)
▶ STUFF_-0x1.d27e44p+29	Dictionary	(4 items)



The `FXConnectToLastURL` key stores the last connection from the "`Connect to Server`" (`Go | Connect to Server`) function found in the Finder menu.

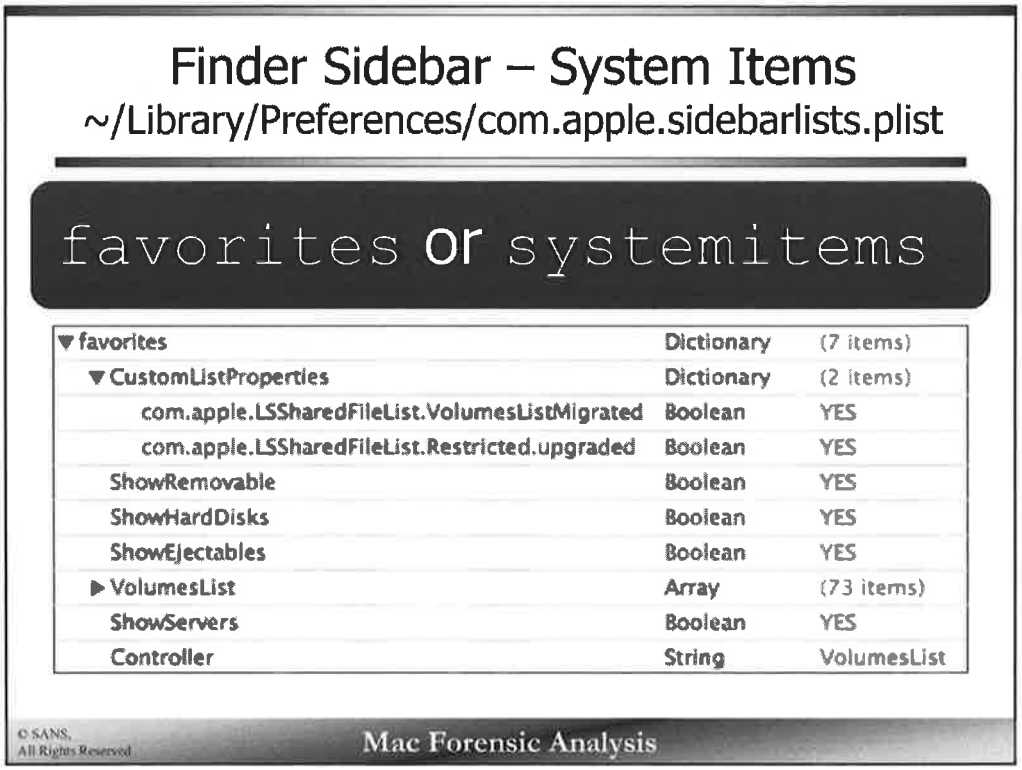


The `GoToField` key stores the last item from the "`Go to the folder:`" function (`Go | Go to Folder...`). This menu option can be used to go directly to a folder using a static or relative file path.



The `RecentSearches` key stores items that have been searched for using the search bar in the top-right corner of the Finder window. It does a "live" search so you may see parts of a search term that is created while the user is typing.

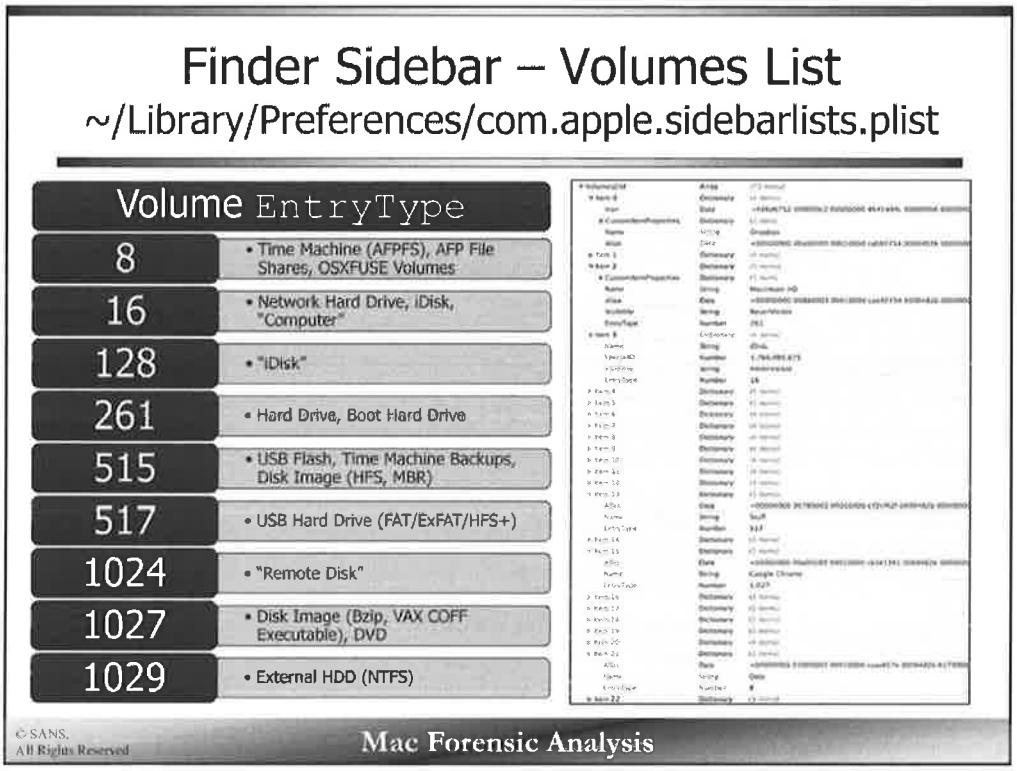
Note: This list does not appear to get populated unless an item is clicked on from the search results. It also appears this is only on 10.8 and older systems.



The `com.apple.sidebarlists.plist` contains the items found in the sidebar of the finder.

The `systemitems` or `favorites` key (version dependent) contains configurable items such as:

- Show Removable Drives (USB, External HDDs)
- Show Hard Drives
- Show Ejectables (iDevices, CDs, DVDs)
- Show Servers



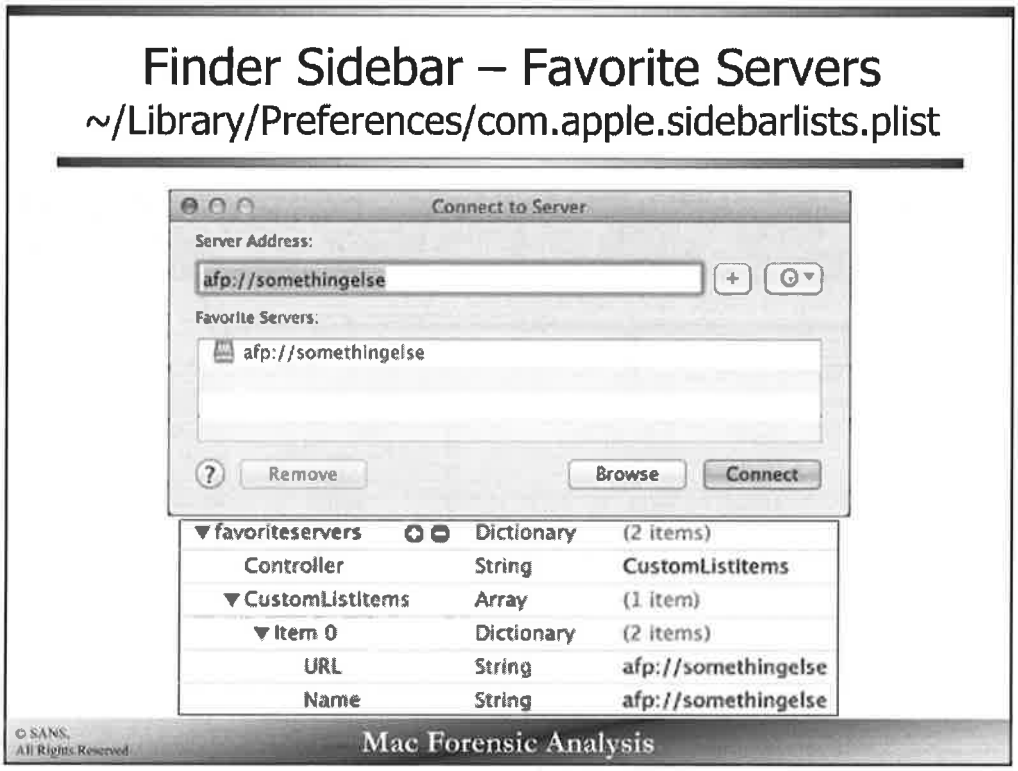
The `VolumesList` keys contain the items found in the sidebar of the finder. This key may contain a large history of items related to mounted volumes.

Each entry may have:

- `Alias` Data
- `Name` – Volume Name
- `Icon` Data – If the volume uses a particular icon.
- `Visibility` Setting – If configured to be viewable by the user or not.
- `EntryType` – Type of disk, some of the types are listed in the table above.

The `EntryType` keys may have different meanings depending on the version of OS X you are analyzing.

▼VolumesList	Array	(73 items)
▼Item 0	Dictionary	(4 items)
Icon	Data	<496d6752 000000c2 00000000 4642494c 000000b6 0000000
▶CustomItemProperties	Dictionary	(1 item)
Name	String	Dropbox
Alias	Data	<00000000 00a00003 00010000 cab93754 0000482b 0000000
▶Item 1	Dictionary	(4 items)
▼Item 2	Dictionary	(5 items)
▶CustomItemProperties	Dictionary	(1 item)
Name	String	Macintosh HD
Alias	Data	<00000000 00880003 00010000 cab93754 0000482b 0000000
Visibility	String	NeverVisible
EntryType	Number	261
▼Item 3	Dictionary	(4 items)
Name	String	iDisk
SpecialID	Number	1,766,093,675
Visibility	String	NeverVisible
EntryType	Number	16
▶Item 4	Dictionary	(5 items)
▶Item 5	Dictionary	(3 items)
▶Item 6	Dictionary	(4 items)
▶Item 7	Dictionary	(4 items)
▶Item 8	Dictionary	(4 items)
▶Item 9	Dictionary	(4 items)
▶Item 10	Dictionary	(4 items)
▶Item 11	Dictionary	(4 items)
▶Item 12	Dictionary	(4 items)
▼Item 13	Dictionary	(3 items)
Alias	Data	<00000000 00780003 00010000 c72cf62f 0000482b 0000000
Name	String	Stuff
EntryType	Number	517
▶Item 14	Dictionary	(3 items)
▼Item 15	Dictionary	(3 items)
Alias	Data	<00000000 00a00003 00010000 cb3e1361 0000482b 0000000
Name	String	Google Chrome
EntryType	Number	1,027
▶Item 16	Dictionary	(3 items)
▶Item 17	Dictionary	(3 items)
▶Item 18	Dictionary	(3 items)
▶Item 19	Dictionary	(3 items)
▶Item 20	Dictionary	(4 items)
▼Item 21	Dictionary	(3 items)
Alias	Data	<00000000 03000003 00010000 caae657e 0000482b 6173000
Name	String	Data
EntryType	Number	8
▶Item 22	Dictionary	(3 items)

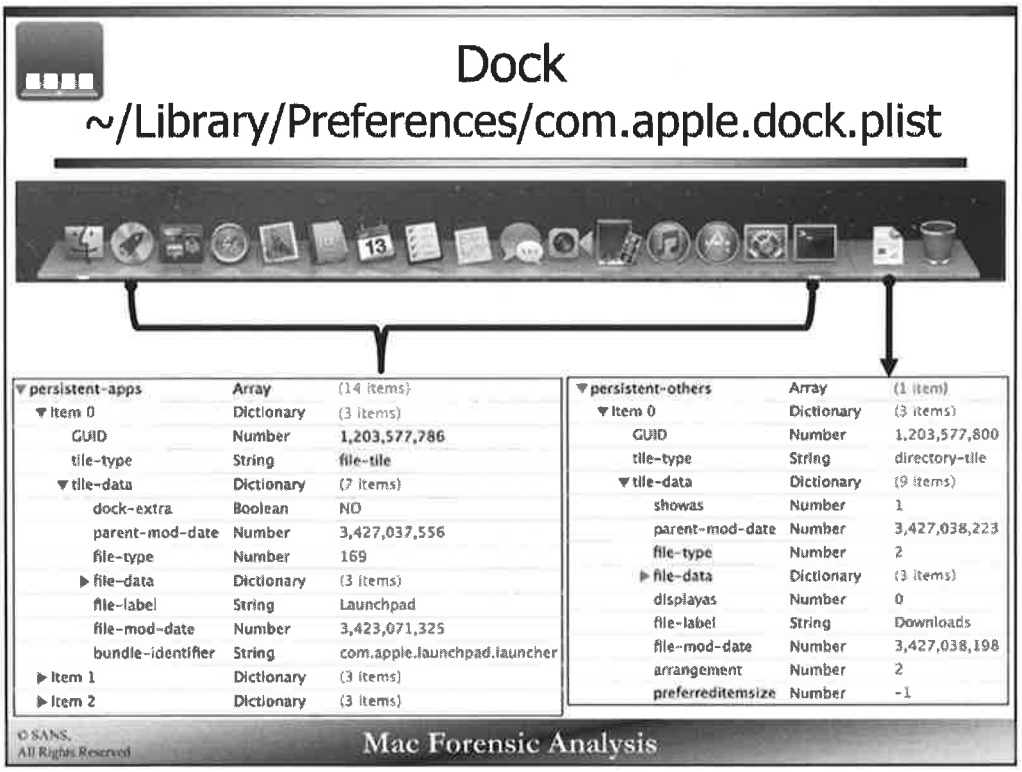


While the `com.apple.finder.plist` stores the last server that may have been connected to, the `com.apple.sidebarlists.plist` contains the favorite servers.

If the `favoriteservers` key does not exist, it means the user does not have any "favorite" servers configured.

Each saved server will have two components:

- `URL` – Server URL
- `Name` – Favorite name, the default is the server URL



The dock can be used to determine what applications a user is more likely to use often.

The `com.apple.dock.plist` contains two main keys:

- `persistent-apps` – These are the applications on the left of the dock separator (not including Finder). These applications are listed in order from left to right (Item 0, Item 1, Item 2…Item N).
- `persistent-others` – These are folders that can be docked and viewed, the default folder in the dock is Downloads, however many times you'll also see the Document directory.

Each Item has data about the application or directory included:

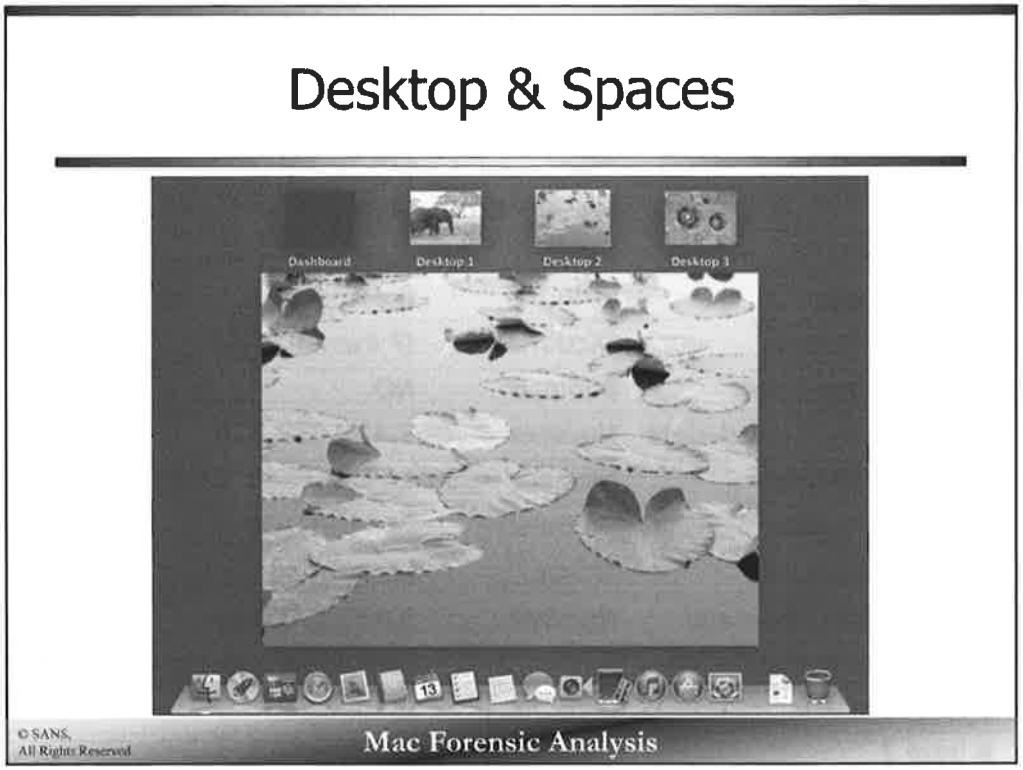
- View Types
- File Data, include Alias data and file path.
- Dock Label
- Modification Dates (File & Parent)

Other preference items may be found in this property list:

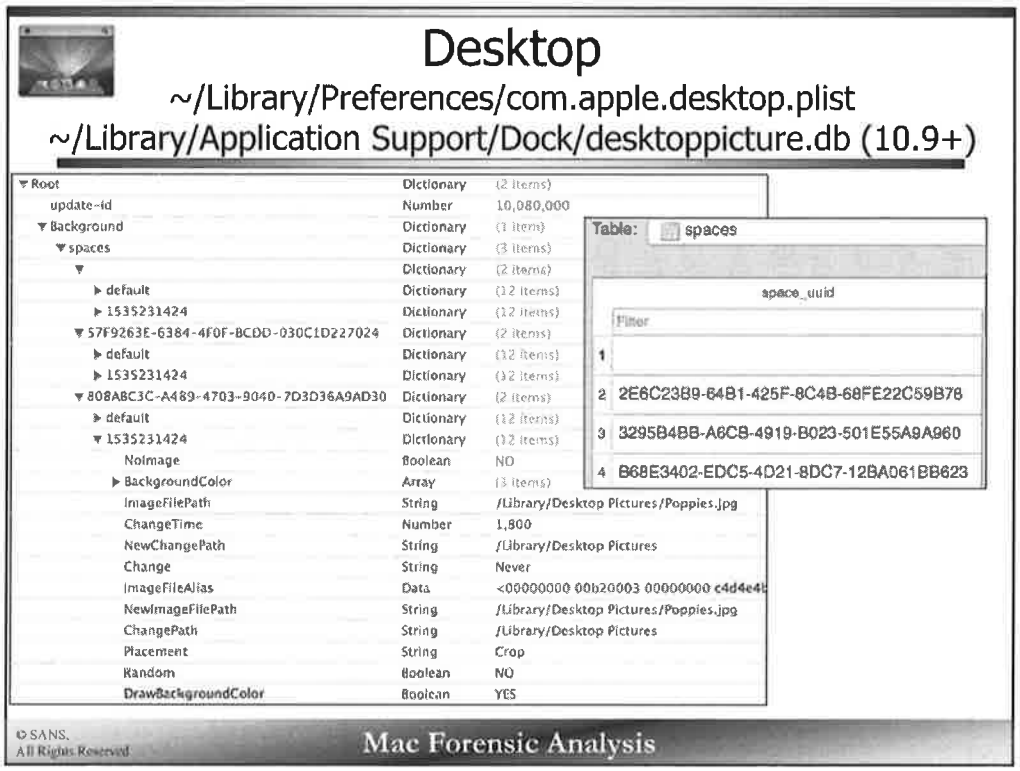
- Dock Auto Hide
- Full Trash
- Icon Sizes
- Dock Magnification

▼ persistent-apps	Array	(14 items)
▼ Item 0	Dictionary	(3 items)
GUID	Number	1,203,577,786
tile-type	String	file-tile
▼ tile-data	Dictionary	(7 items)
dock-extra	Boolean	NO
parent-mod-date	Number	3,427,037,556
file-type	Number	169
▶ file-data	Dictionary	(3 items)
file-label	String	Launchpad
file-mod-date	Number	3,423,071,325
bundle-identifier	String	com.apple.launchpad.launcher
▶ Item 1	Dictionary	(3 items)
▶ Item 2	Dictionary	(3 items)

▼ persistent-others	Array	(1 item)
▼ Item 0	Dictionary	(3 items)
GUID	Number	1,203,577,800
tile-type	String	directory-tile
▼ tile-data	Dictionary	(9 items)
showas	Number	1
parent-mod-date	Number	3,427,038,223
file-type	Number	2
▶ file-data	Dictionary	(3 items)
displayas	Number	0
file-label	String	Downloads
file-mod-date	Number	3,427,038,198
arrangement	Number	2
preferreditemsize	Number	-1



The OS X desktop can have more than one desktop. Shown in the screenshot above, there are three desktops and the dashboard. Each desktop is called a "Space".

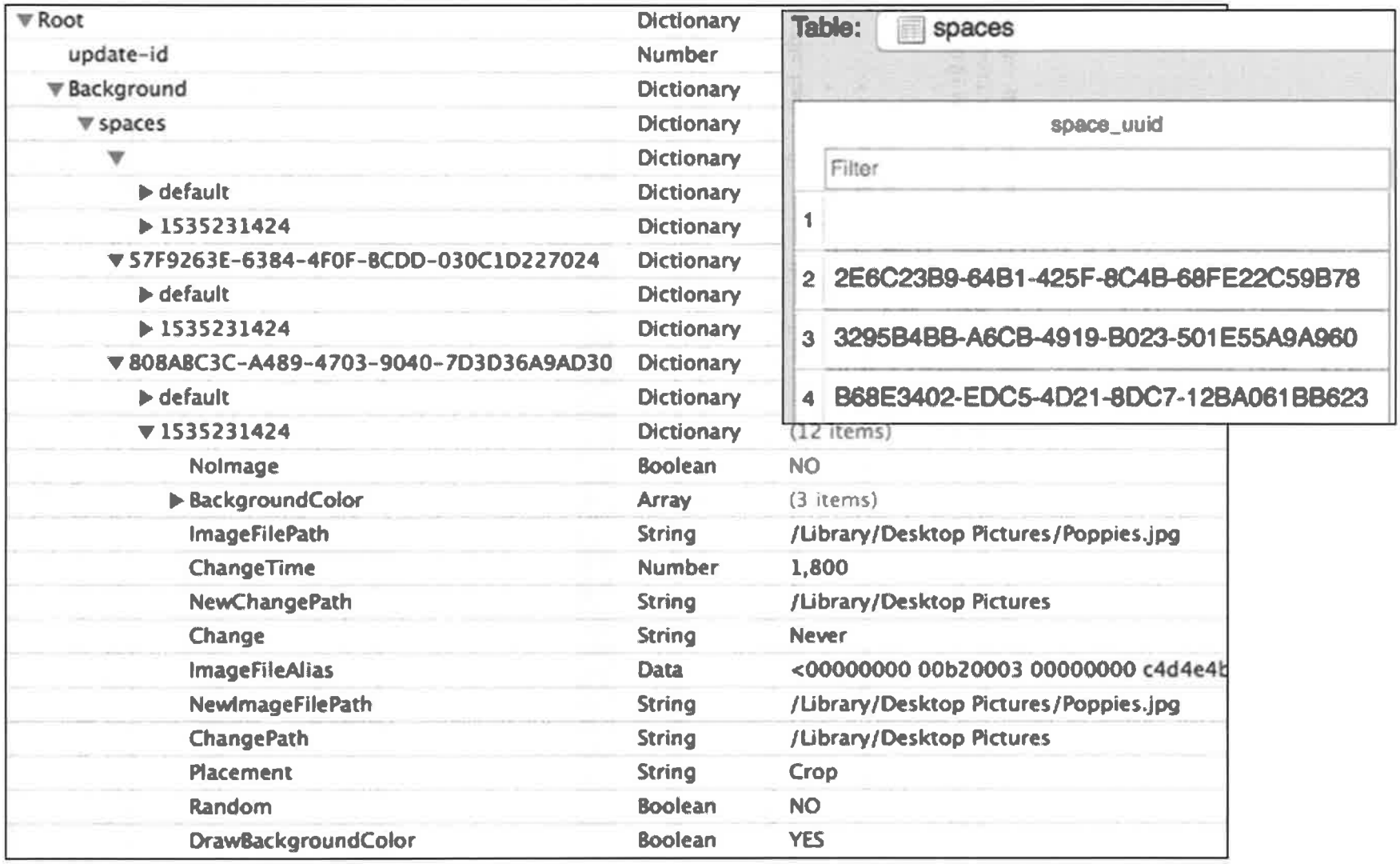


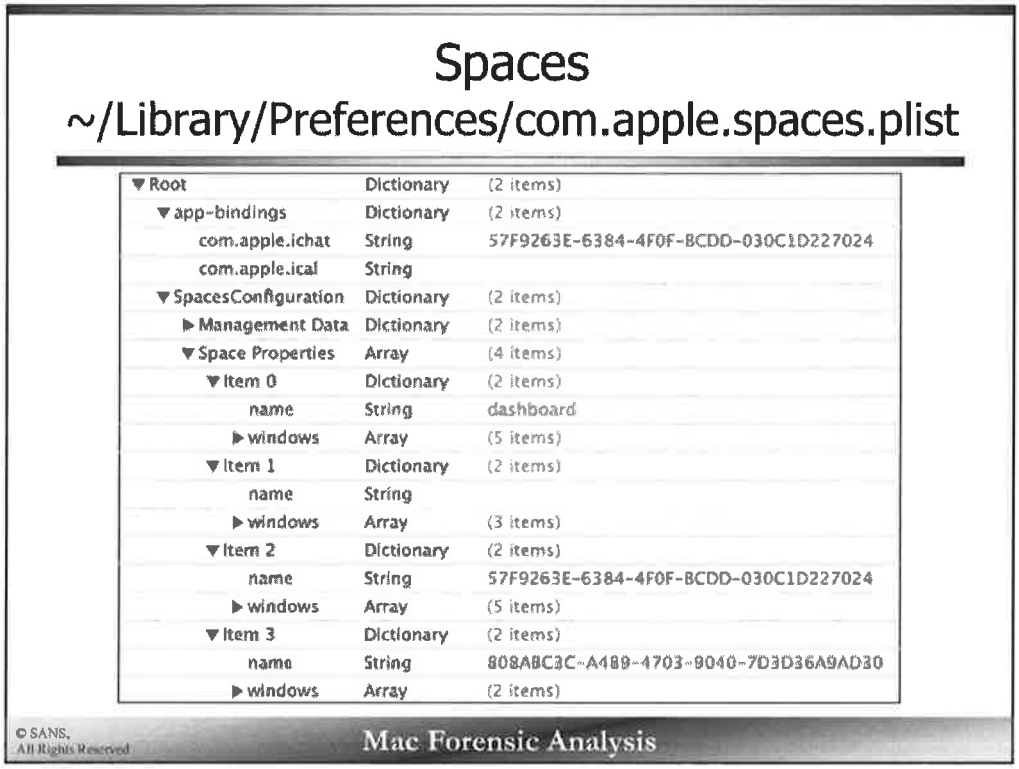
The `com.apple.desktop.plist` file contains details for each of the desktop spaces. Each desktop has a GUID identifier, except for the first desktop where it is left blank.

Each desktop has two profiles; a default and a numbered one. In general the data will be the same for each profile. Each profile contains the configuration data for that desktop including:

- Background Image Data
- Change Background – The screenshot example does not change the background. If this were true, it would show "`TimeInterval`"

On 10.9+ systems much of this information is stored in a database located `~/Library/Application Support/Dock/desktoppicture.db`. In the inset screenshot, the spaces table is shown with space GUIDs – the first space still does not have an assigned GUID, it is blank.


Root Dictionary
update-id Number
Background Dictionary
spaces Dictionary
Dictionary
default Dictionary
1535231424 Dictionary
57F9263E-6384-4F0F-8CDD-030C1D227024 Dictionary
default Dictionary
1535231424 Dictionary
808ABC3C-A489-4703-9040-7D3D36A9AD30 Dictionary
default Dictionary
1535231424 Dictionary (12 items)
NoImage Boolean NO
BackgroundColor Array (3 items)
ImageFilePath String /Library/Desktop Pictures/Poppies.jpg
ChangeTime Number 1,800
NewChangePath String /Library/Desktop Pictures
Change String Never
ImageFileAlias Data <00000000 00b20003 00000000 c4d4e4b
NewImageFilePath String /Library/Desktop Pictures/Poppies.jpg
ChangePath String /Library/Desktop Pictures
Placement String Crop
Random Boolean NO
DrawBackgroundColor Boolean YES
Table: spaces
space_uuid
Filter
1
2 2E6C23B9-64B1-425F-8C4B-68FE22C59B78
3 3295B4BB-A6CB-4919-B023-501E55A9A960
4 B68E3402-EDC5-4D21-8DC7-12BA061BB623



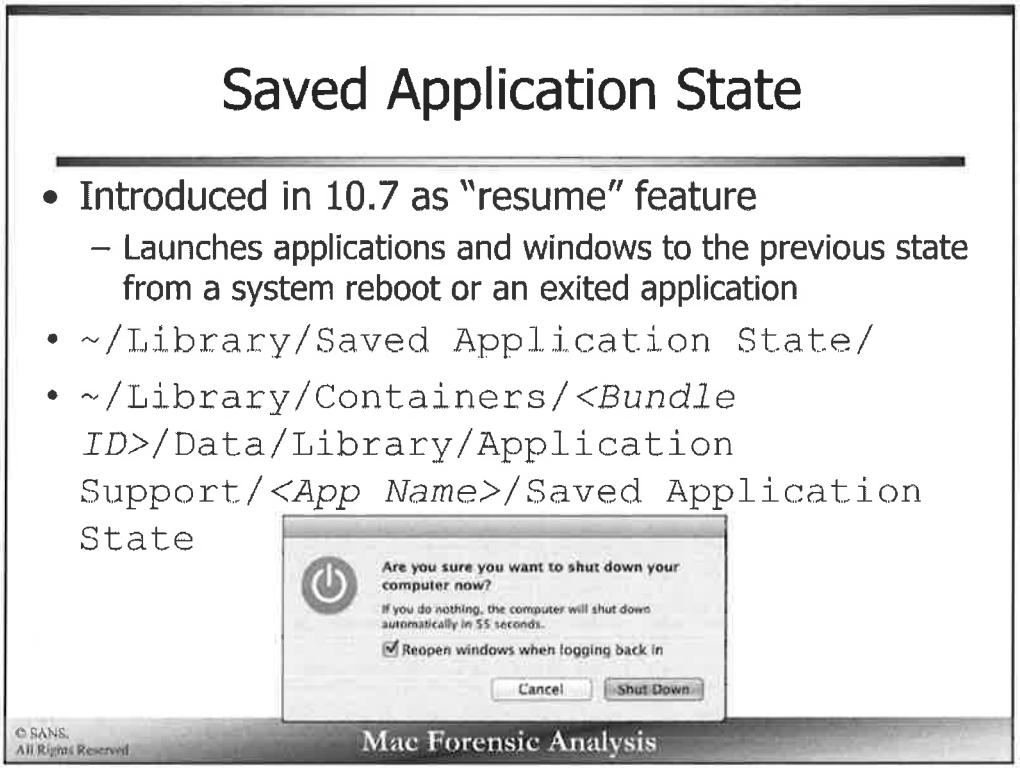
The `com.apple.spaces.plist` shows the same desktop GUIDs from the `com.apple.desktop.plist` file.

Each space can have certain applications bound to it. In the example above, iChat is bound to the second space, while iCal is bound to the first space (remember the first space does not have a GUID, it is blank.) These are stored in the `app-bindings` key.

Each space has a GUID associated with it, these can be correlated to the GUIDs found in `com.apple.desktop.plist`.

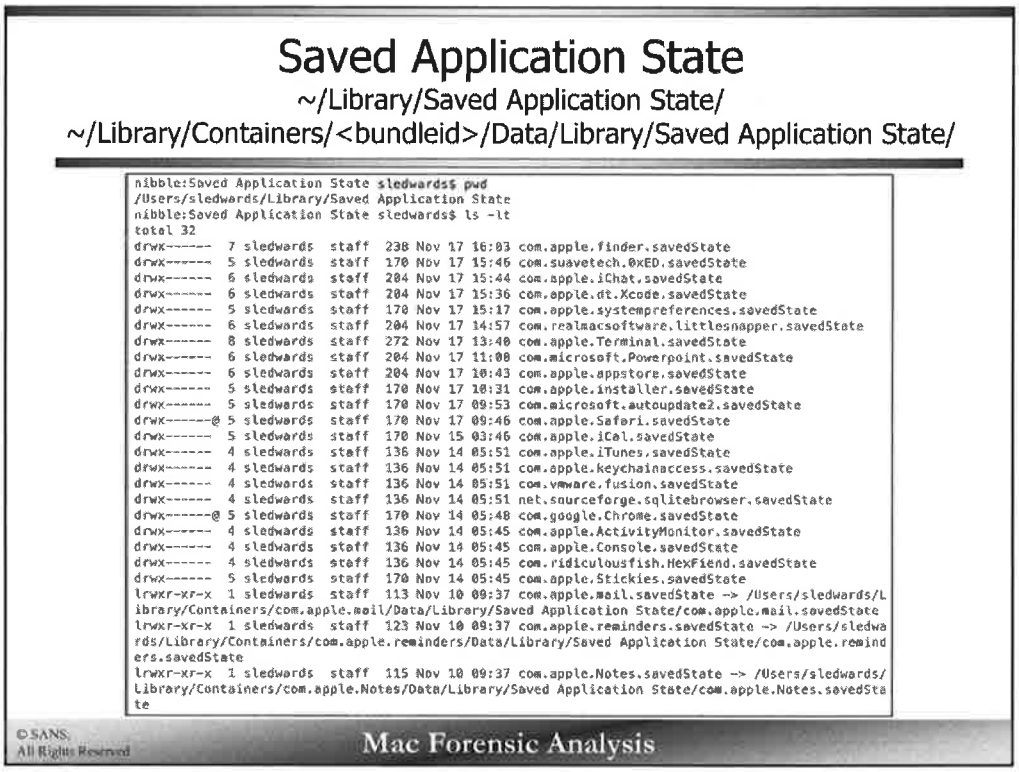
10.6 does not have a `com.apple.spaces.plist` file, this information will be found in the `com.apple.dock.plist` file.

▼ Root	Dictionary	(2 items)
▼ app-bindings	Dictionary	(2 items)
com.apple.ichat	String	57F9263E-6384-4F0F-BCDD-030C1D227024
com.apple.ical	String	
▼ SpacesConfiguration	Dictionary	(2 items)
▶ Management Data	Dictionary	(2 items)
▼ Space Properties	Array	(4 items)
▼ Item 0	Dictionary	(2 items)
name	String	dashboard
▶ windows	Array	(5 items)
▼ Item 1	Dictionary	(2 items)
name	String	
▶ windows	Array	(3 items)
▼ Item 2	Dictionary	(2 items)
name	String	57F9263E-6384-4F0F-BCDD-030C1D227024
▶ windows	Array	(5 items)
▼ Item 3	Dictionary	(2 items)
name	String	808ABC3C-A489-4703-9040-7D3D36A9AD30
▶ windows	Array	(2 items)



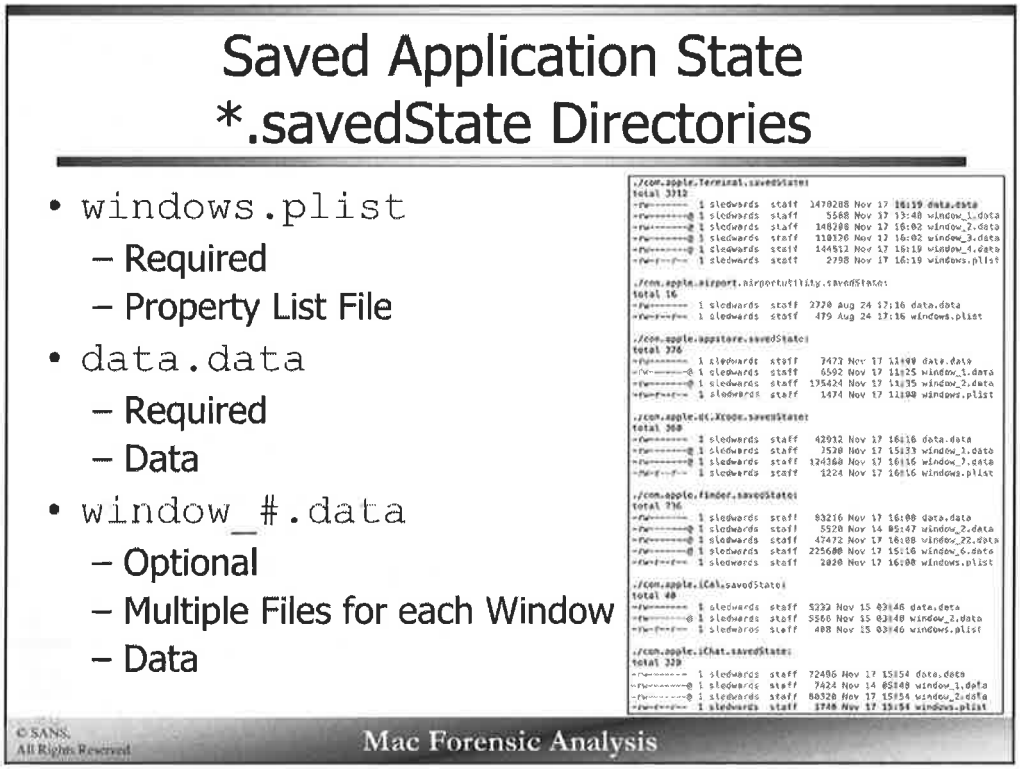
The Saved Application State, introduced in 10.7, is used to return applications to their previous state after a reboot. When shutting down a user is given a choice to "`reopen windows when logging back in`".

The directory `~/Library/Saved Application State/` also contains links to the `.savedState` directories in the sandboxed application containers.



Each application has its own directory named `<Bundle ID>.savedState`. Notice the links to the application sandbox containers.

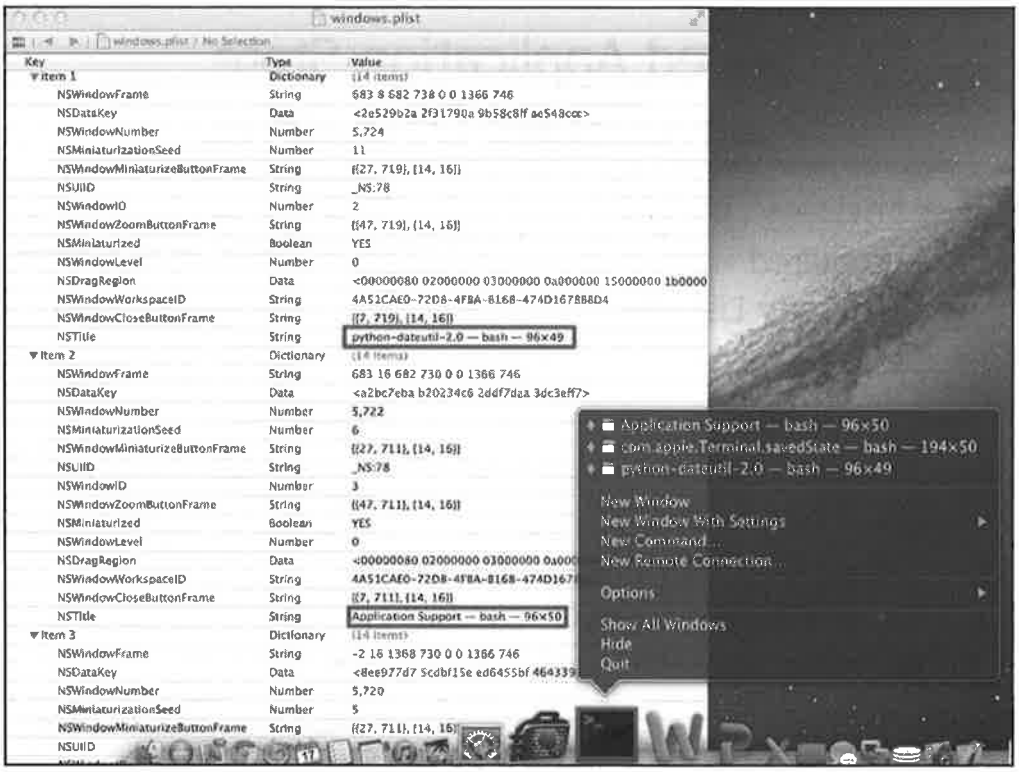
The existence of these directories mean the user has used these applications.



Each `.savedState` directory contains at least two files:

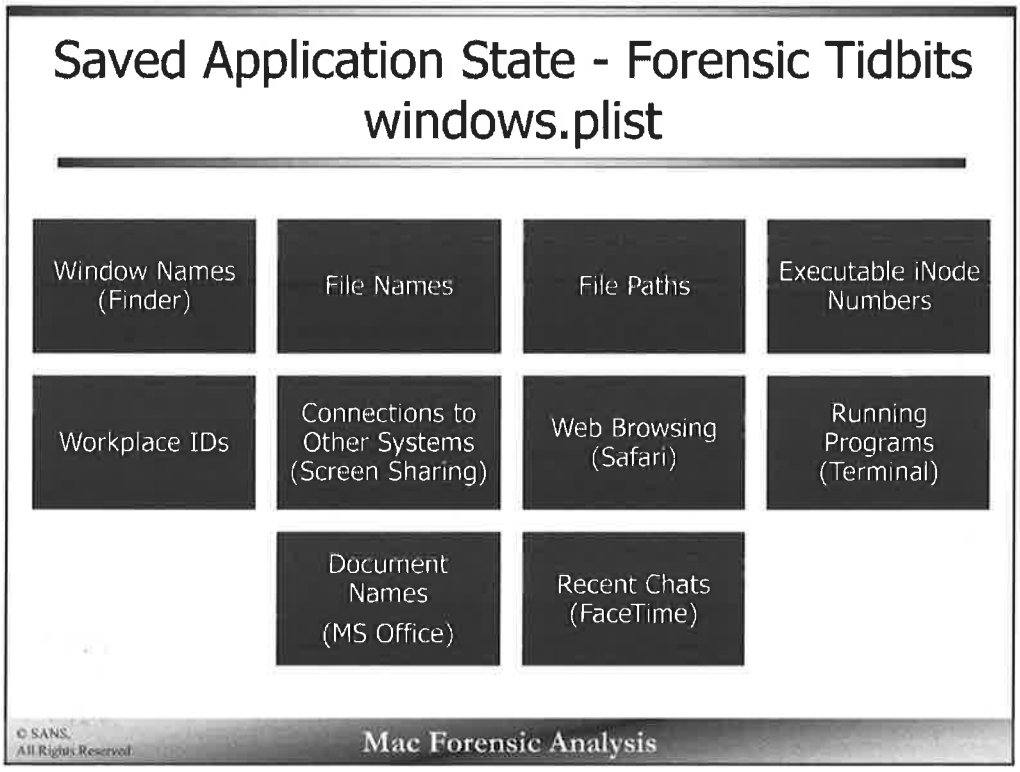
- `windows.plist`
- `data.data`

Additional files named `windows_1.data`, `windows_2.data`, etc. may also be created.



The `windows.plist` file contains information about the application windows that should be "reopened".

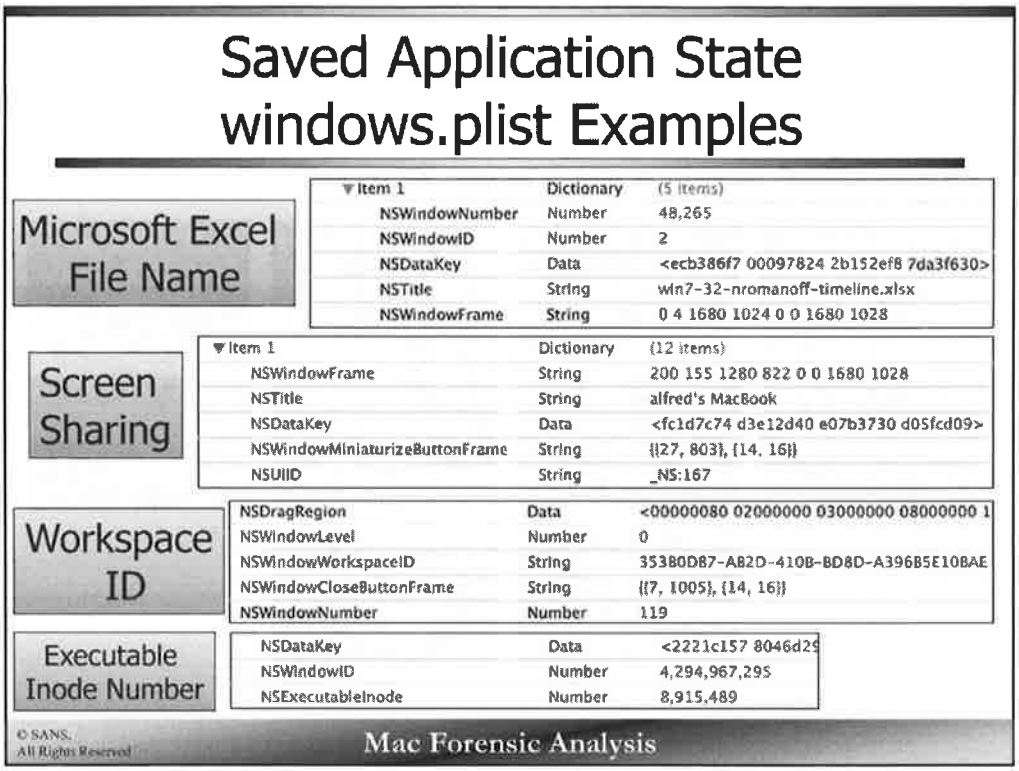
While there are some windows positioning data, the interesting part is the `NSTitle` for each Item. This contains the name of the application window. In the screenshot the Terminal window names are shown in the right-click menu. This information may help a forensic analyst determine what was running or otherwise happening with that application.



Lots of good forensic tidbits can be found looking at the `windows.plist` file for an application.

- Finder holds the names of opened Windows.
- File Names are often found, specifically with applications that open documents. Microsoft Office will have the names of opened files because they use the filename as the window name.
- Workplace IDs, used to keep track of Spaces can be used to determine if an application is opened on a specific Space.
- Screen Sharing will save the names of connected systems.
- Browser programs like Safari save the web page title as the window name.
- Programs running in Terminal can be seen in window names, look for things like `sudo`, `ssh`, etc.

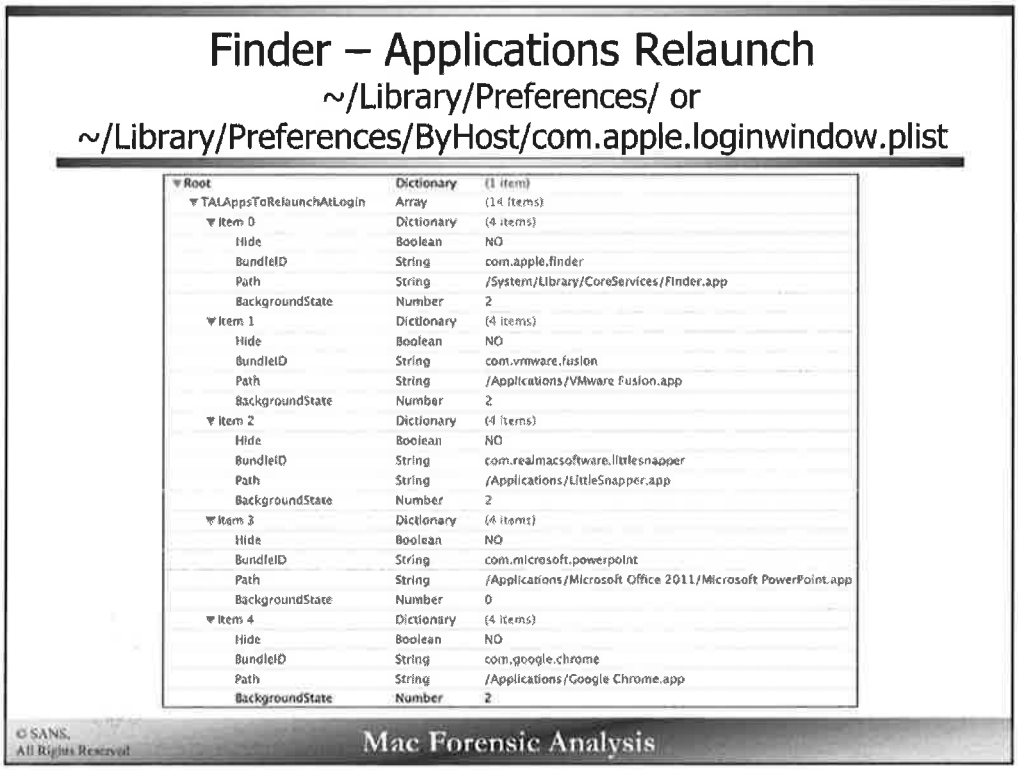
Many `window.plist` files have a key named `NSExecutableInode`, this is used to store the iNode number of the related application executable.



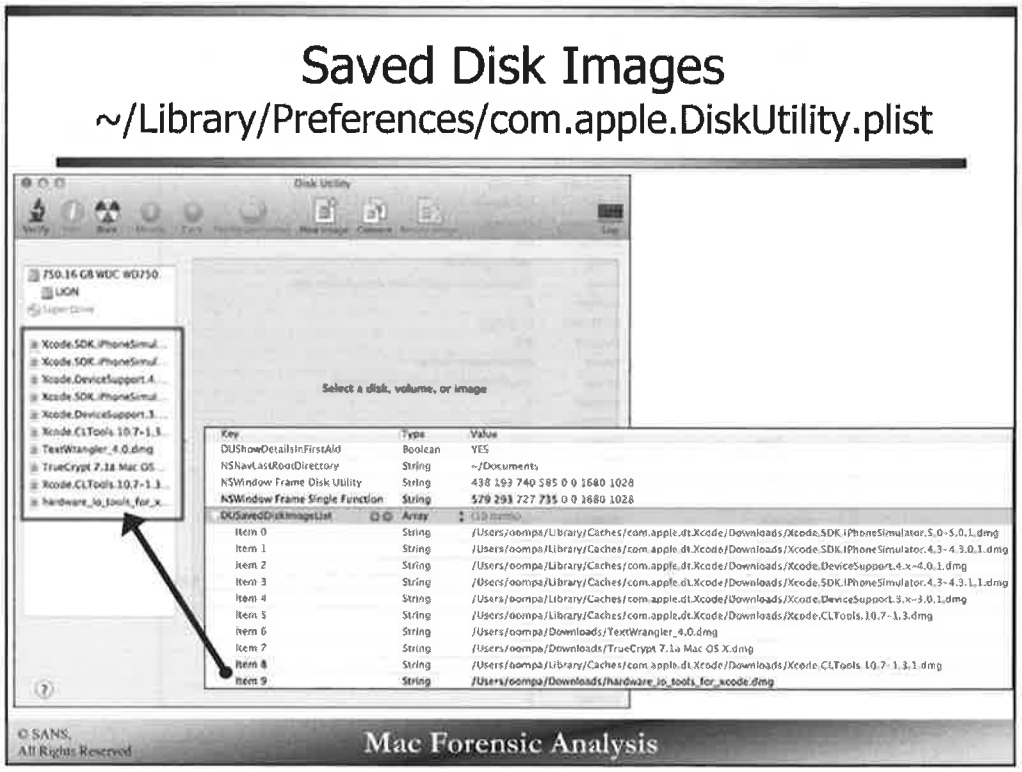
The screenshots show examples of what can be found in Saved Application State `windows.plist` files.

From top to bottom:

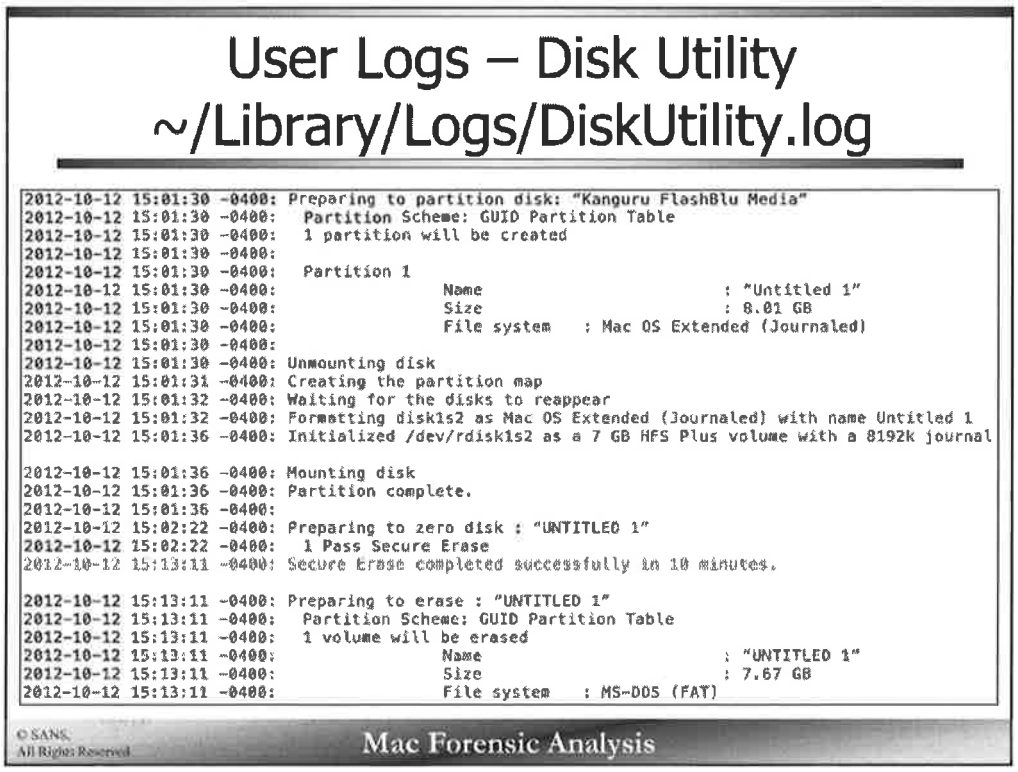
- A filename used with Microsoft Excel
- A connected Screen Sharing System – `alfred's MacBook`
- A Workspace ID used in Spaces
- An `NSExecutableInode` number that can be tracked back to a specific application.



Applications set to relaunch can be found in the `com.apple.loginwindow.plist` property list located in `~/Library/Preferences/` or the `~/Library/Preferences/ByHost/` directories.



The Disk Utility application often saves disk images in the sidebar of Disk Utility. These disk images are saved in the `com.apple.DiskUtility.plist` property list in the user's `Preferences` directory.



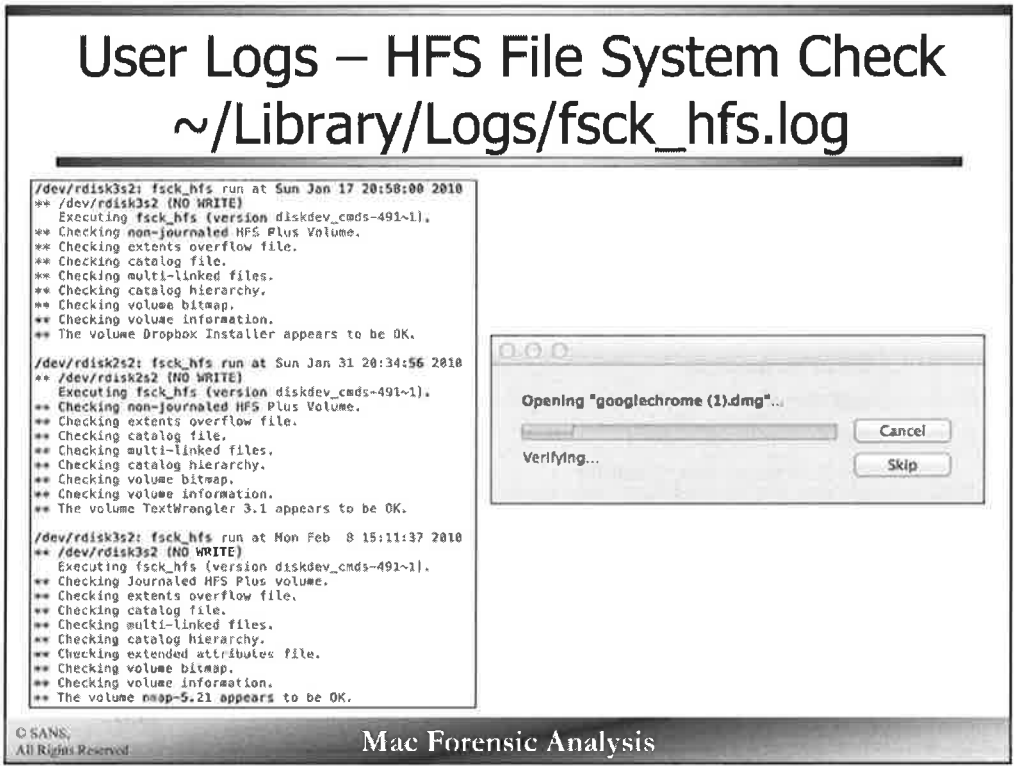
Each user will have their own `DiskUtility.log`.

This log contains a history of:

- Mounting Volumes
- Partitioning
- Formatting
- Erasing Volumes
- DMG Creation and Manipulation
- System Restoration
- …and more disk and volume manipulation

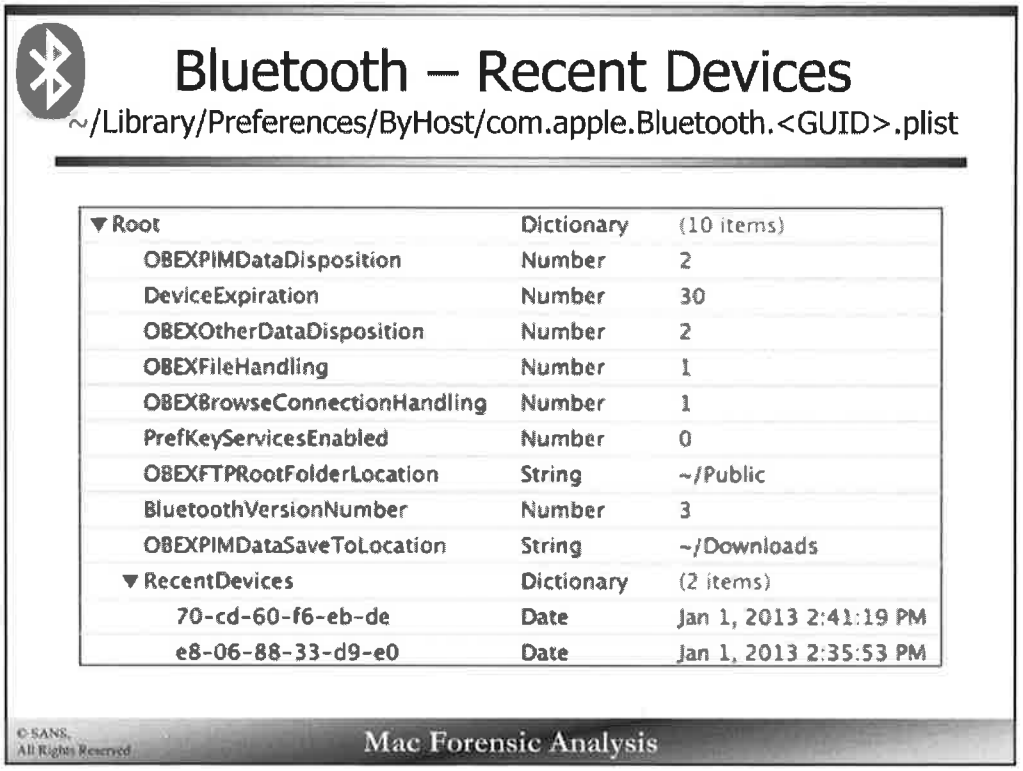
Volume metadata can be found here also, an example is shown on the next page.

```
Name:      Macintosh HD
Type :     Encrypted Logical Partition
Disk Identifier :   disk1
Mount Point :       /
Disk Status :       Online
System Name :       OS X
System Version :    10.8.1
System Build :      12B19
System Copyright :  1983-2012 Apple Inc.
File System :       Mac OS Extended (Journaled, Encrypted)
Writable :          Yes
Universal Unique Identifier : 3981E2E6-0CAC-3A3E-BE1D-
90D583F89A5D
Capacity :          748.98 GB (748,977,844,224 Bytes)
Used :    398.62 GB (398,620,389,376 Bytes)
Number of Files :   935,405
Number of Folders :          218,939
Owners Enabled :    Yes
Can Turn Owners Off :        Yes
Can Repair Permissions :     Yes
Can Be Verified :   Yes
Can Be Repaired :   Yes
Can Be Formatted :  Yes
Bootable :          Yes
Supports Journaling :        Yes
Journaled :         Yes
```



Each user will have their own `fsck_hfs.log`.

This log keeps track of mounted volumes that execute the "verifying" process when a user opens a disk image file. This can show programs that the user may have installed, when the disk image was opened, and the location on the `/dev/` tree it may be found (i.e., `/dev/rdisk3s2`).



To know what Bluetooth devices may have been used, we can view the `com.apple.Bluetooth` property list in the `ByHost` directory of the user's `Preferences` directory.

The `RecentDevices` key holds the Bluetooth MAC address of the device, and the date it was last paired.

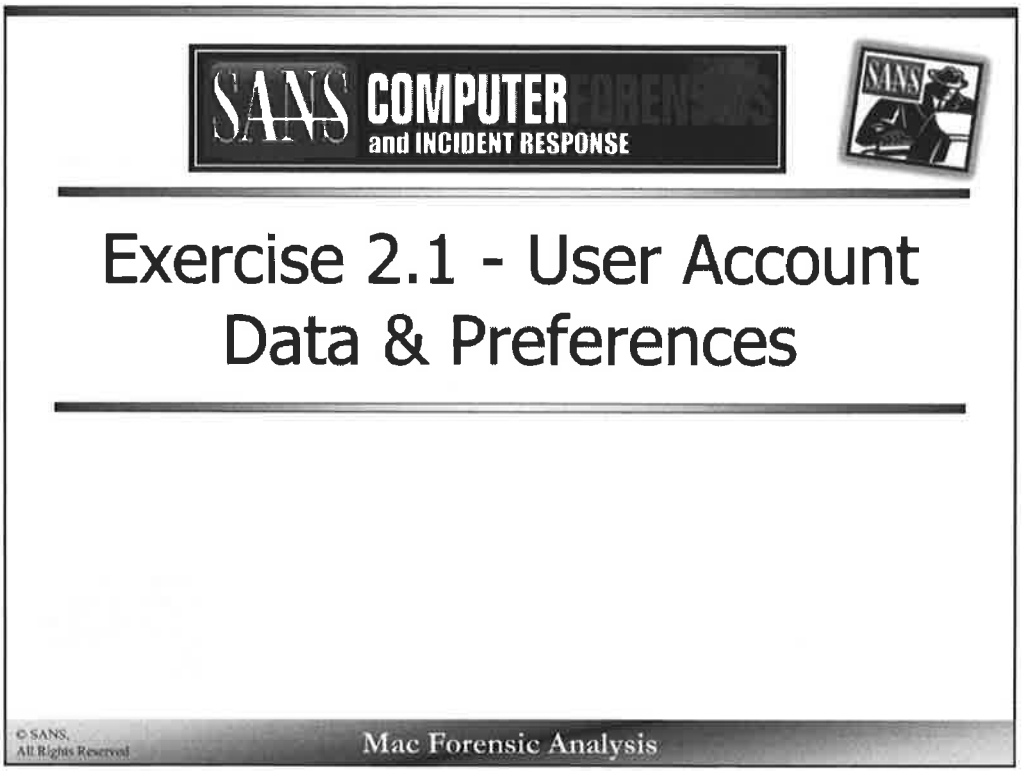
Printers – MRU

~/Library/Preferences/org.cups.PrintingPrefs.plist

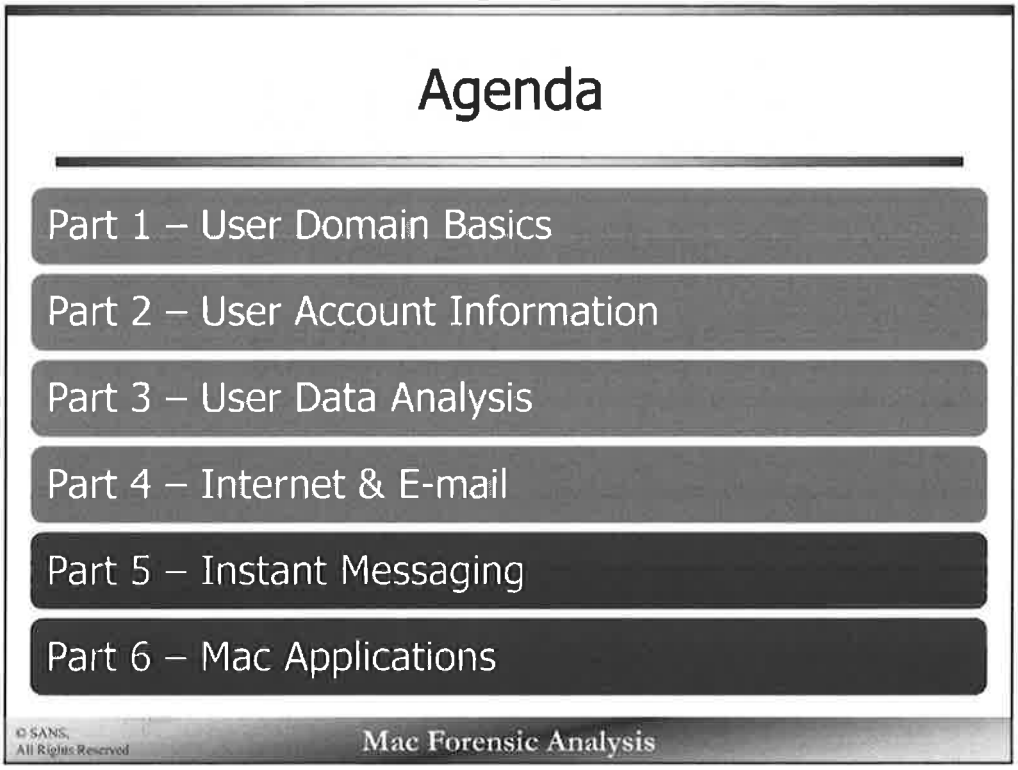
▼Root	Dictionary	(1 item)
▼LastUsedPrinters	Array	(1 item)
▼Item 0	Dictionary	(2 items)
Network	String	192.168.1.254
PrinterID	String	Brother_HL_2170W_series

What printer a user has used now or in the past may come up during an investigation. The property list `org.cups.PrinterPrefs.plist` keeps track of what printers were used.

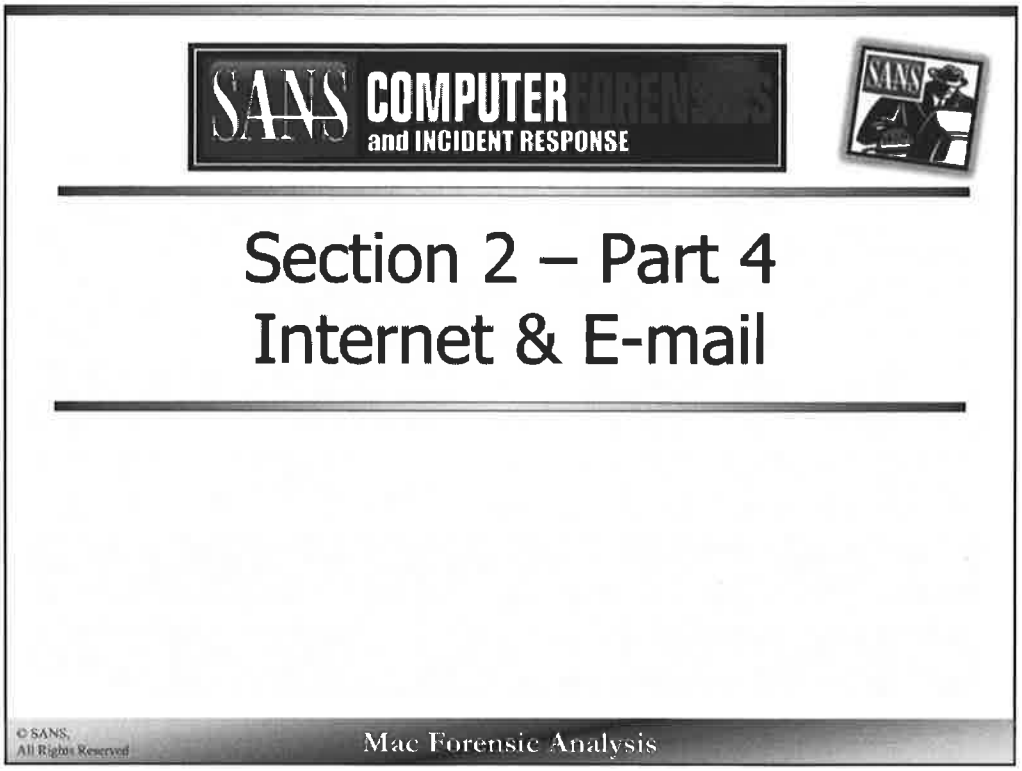
Each printer will have a network address (IP of default gateway) and a name (`PrinterID`). Section 3 will detail additional artifacts of printing.



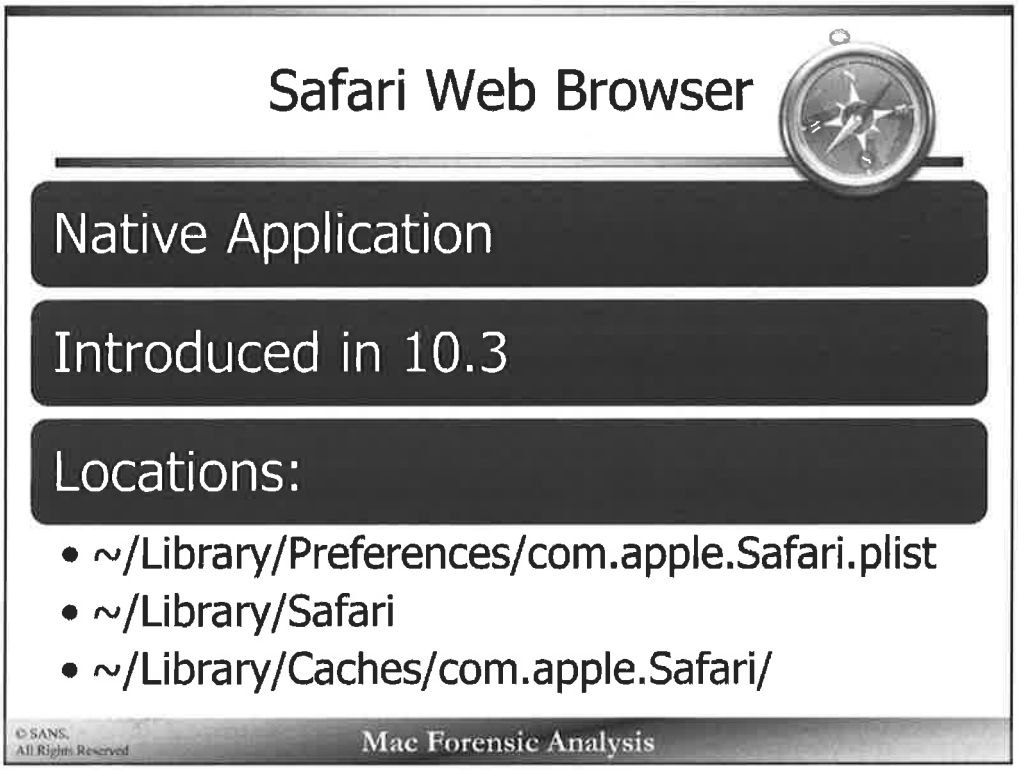
This page intentionally left blank.



This page intentionally left blank.

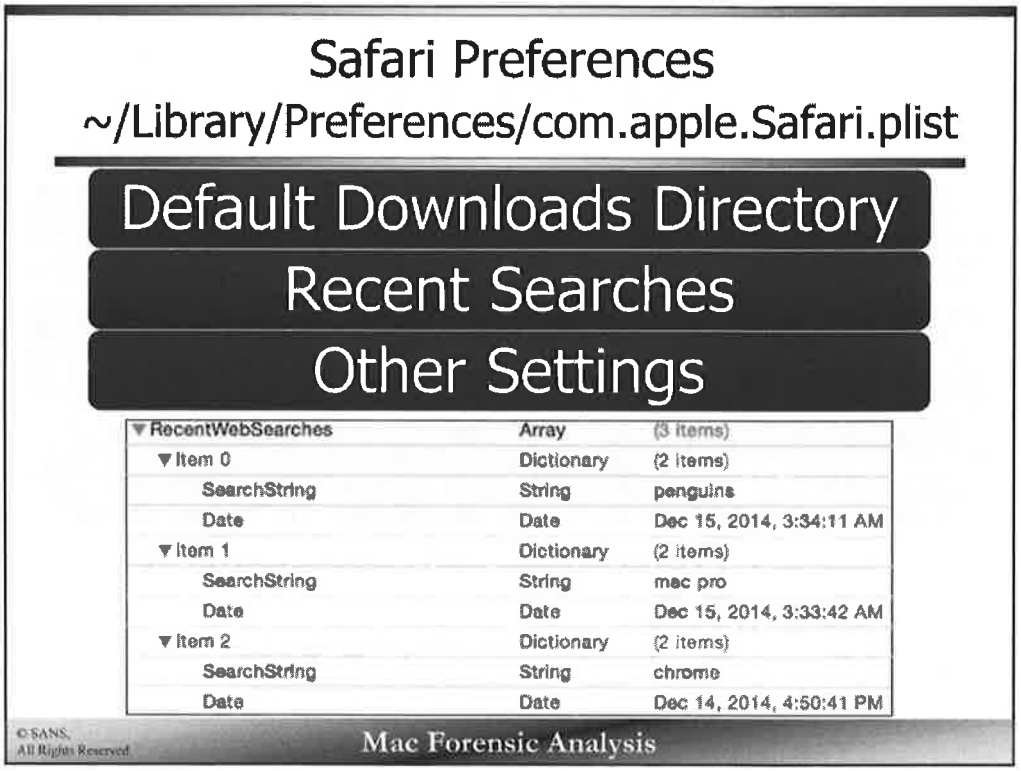


This page intentionally left blank.



The web browser native to OS X since 10.3 is the Safari Web Browser. Safari forensic artifacts are stored in different files in various directories.

Other browsers may also be used on OS X such as Firefox, Google Chrome, and Opera.

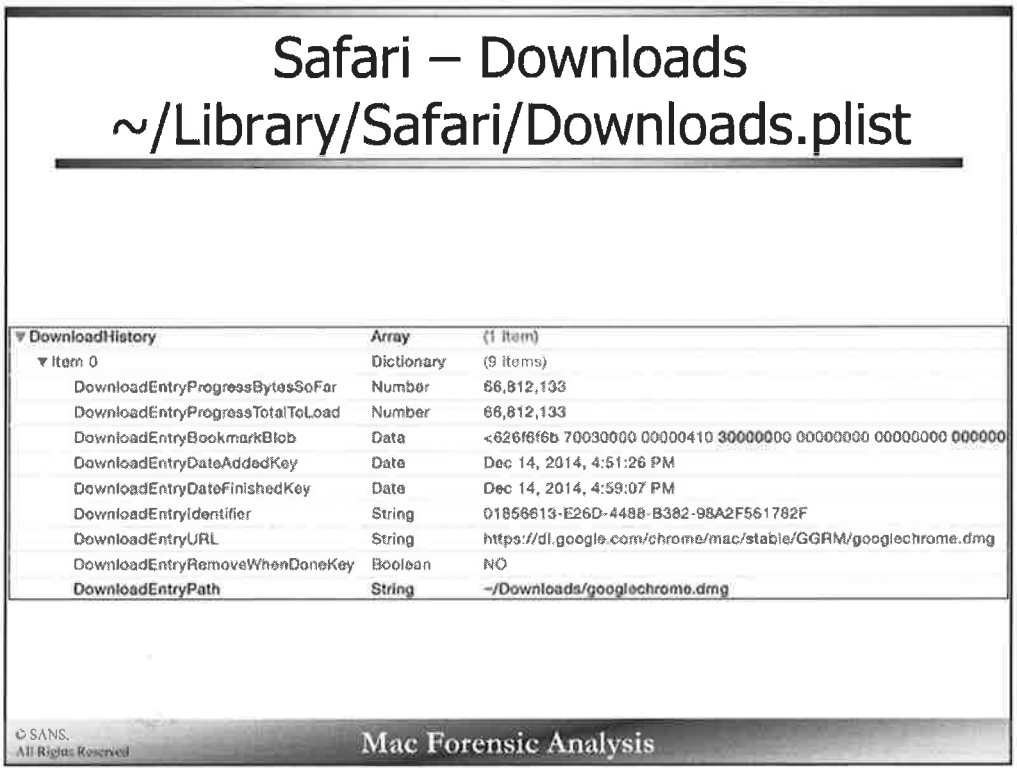


The Safari preferences file, `com.apple.Safari.plist`, can be found in the users `Preferences` directory.

In the screenshot above we can see some of the recent search strings and the default downloads directory. Other preference settings include:

- Permissions
- Timestamp (Launch, Update, Cache, etc.)
- Sync Information
- GUI Settings

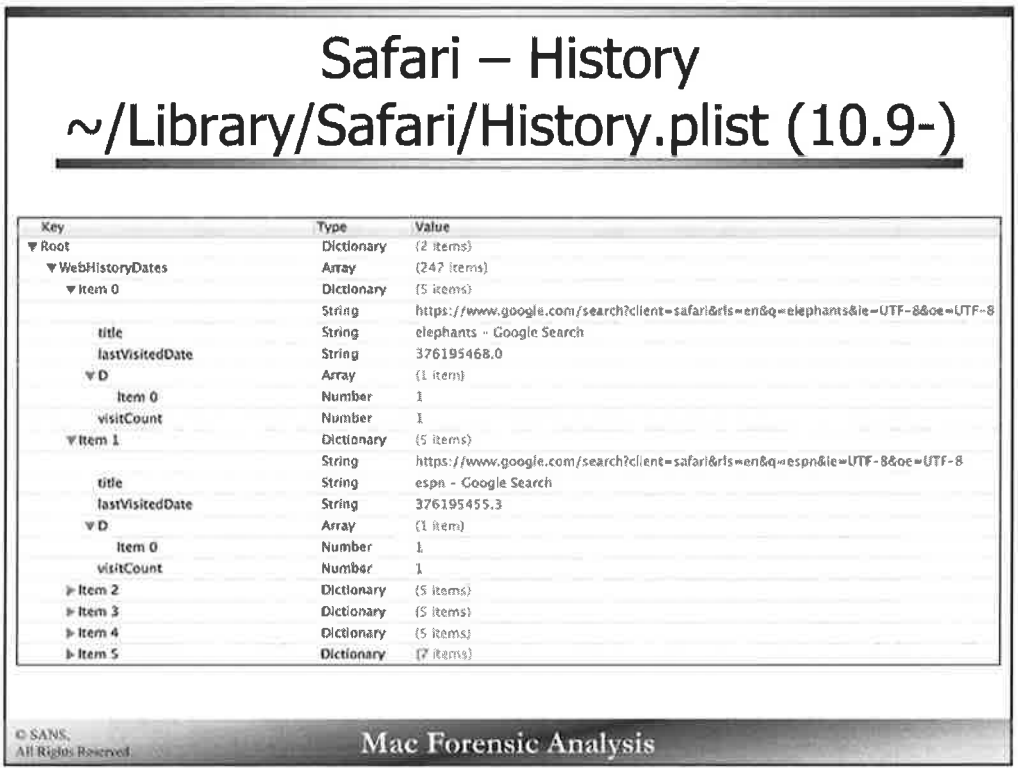
Newer versions of OS X provide a timestamp in the `Date` key when this term was searched for.



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

- **Download Entry Identifier** – Each download has a unique GUID
- **Download Entry URL** – A URL is saved showing where the download originated
- **Download Entry Progress Total to Load, Progress Bytes So Far** - These keys store the total bytes and downloaded bytes for the download. The download file may not always have the same quantity of bytes if the download was canceled or otherwise stopped.
- **Download Entry Path** – Where the item was downloaded to, very likely will be in the default Downloads Directory (`~/Downloads`)
- **Download Entry Bookmark/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+)

▼ 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 000000
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

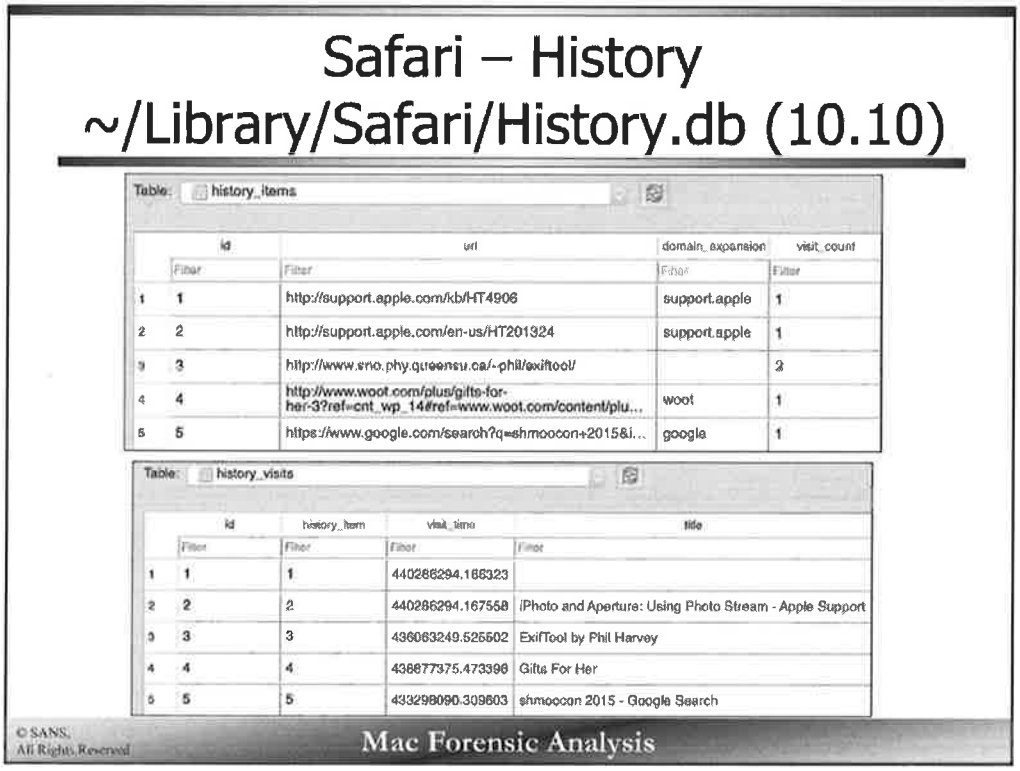


The `History.plist` file located in the Safari directory holds the Internet history. Each history `Item` contains the following data:

- **Blank key** – Contains the visited URL
- **Title** – Page Title
- **Last Visited Date** – Date the page was last visited in Webkit data format/Mac absolute time. The windows-based tool, Dcode from Digital Detective and Mac-based tool, Epoch Converter from BlackBag Technologies are free tools to decode this date format.
- **Visit Count** – The number of times a page was visited.

Key	Type	Value
▼Root	Dictionary	(2 items)
▼WebHistoryDates	Array	(247 items)
▼Item 0	Dictionary	(5 items)
	String	https://www.google.com/search?client=safari&rls=en&q=elephants&ie=UTF-8&oe=UTF-8
title	String	elephants - Google Search
lastVisitedDate	String	376195468.0
▼D	Array	(1 item)
Item 0	Number	1
visitCount	Number	1
▼Item 1	Dictionary	(5 items)
	String	https://www.google.com/search?client=safari&rls=en&q=espn&ie=UTF-8&oe=UTF-8
title	String	espn - Google Search
lastVisitedDate	String	376195455.3
▼D	Array	(1 item)
Item 0	Number	1
visitCount	Number	1
▶Item 2	Dictionary	(5 items)
▶Item 3	Dictionary	(5 items)
▶Item 4	Dictionary	(5 items)
▶Item 5	Dictionary	(7 items)

106

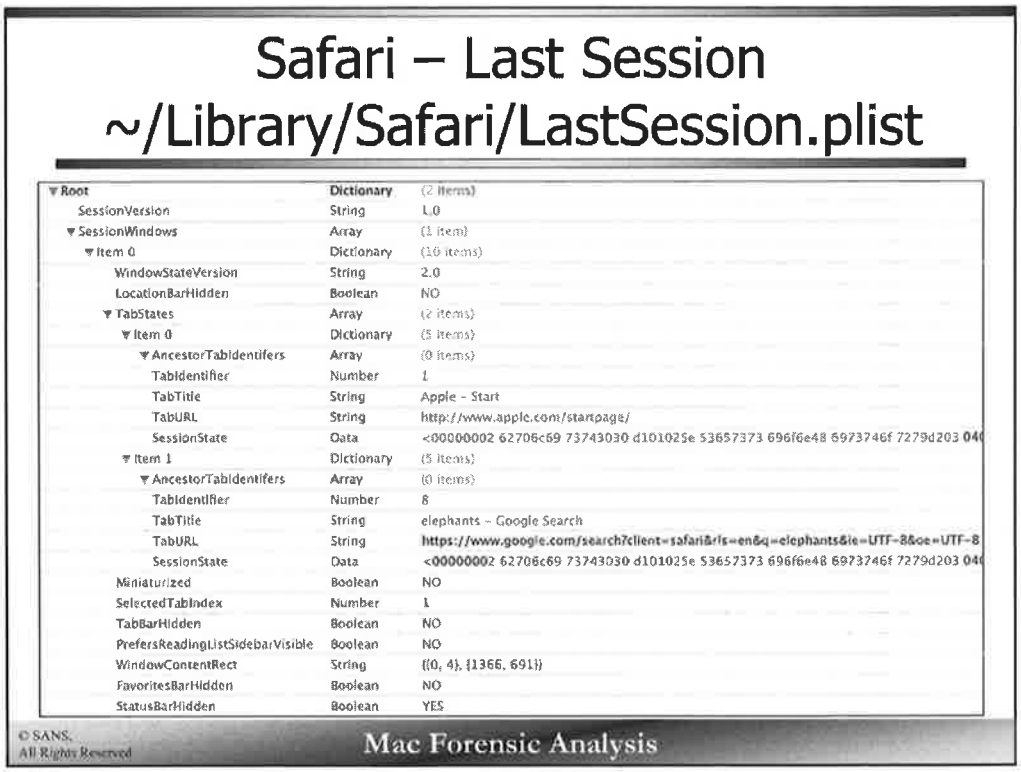


The `History.db` `SQLite` database located in the Safari directory holds the Internet history, similar information found in the previous `History.plist` file.

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.

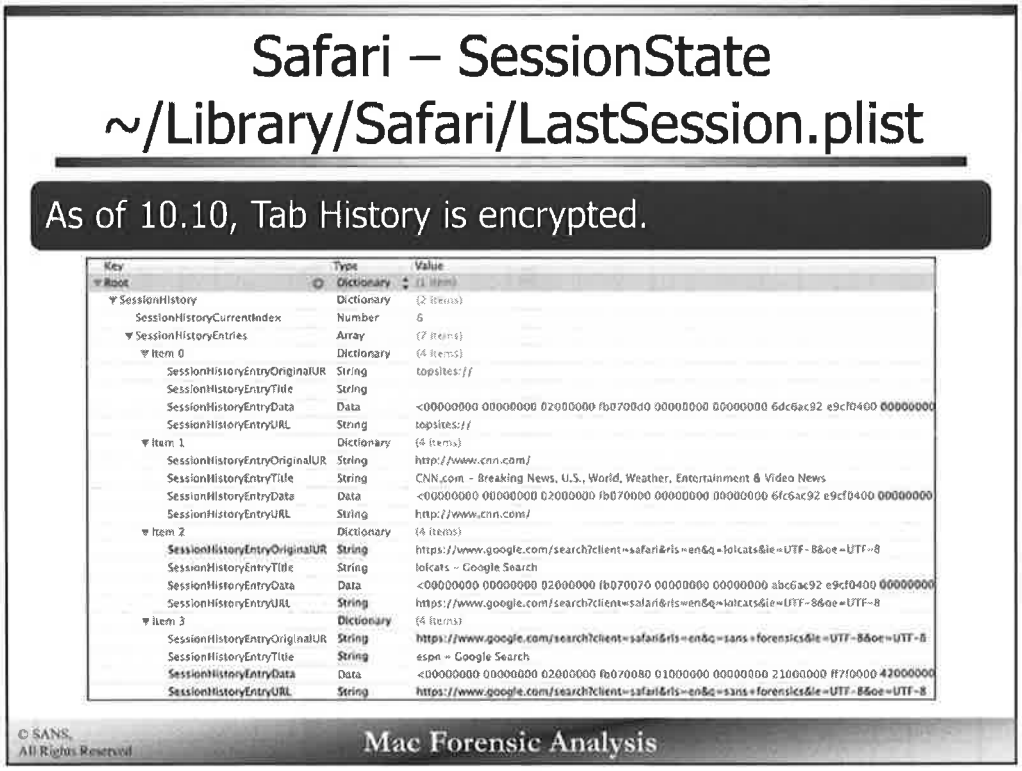
You can correlate the data in these two tables with the "`id`" column.



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 web page title and URL, respectively. The binary property list field `SessionState` key holds more information, shown on the next slide.

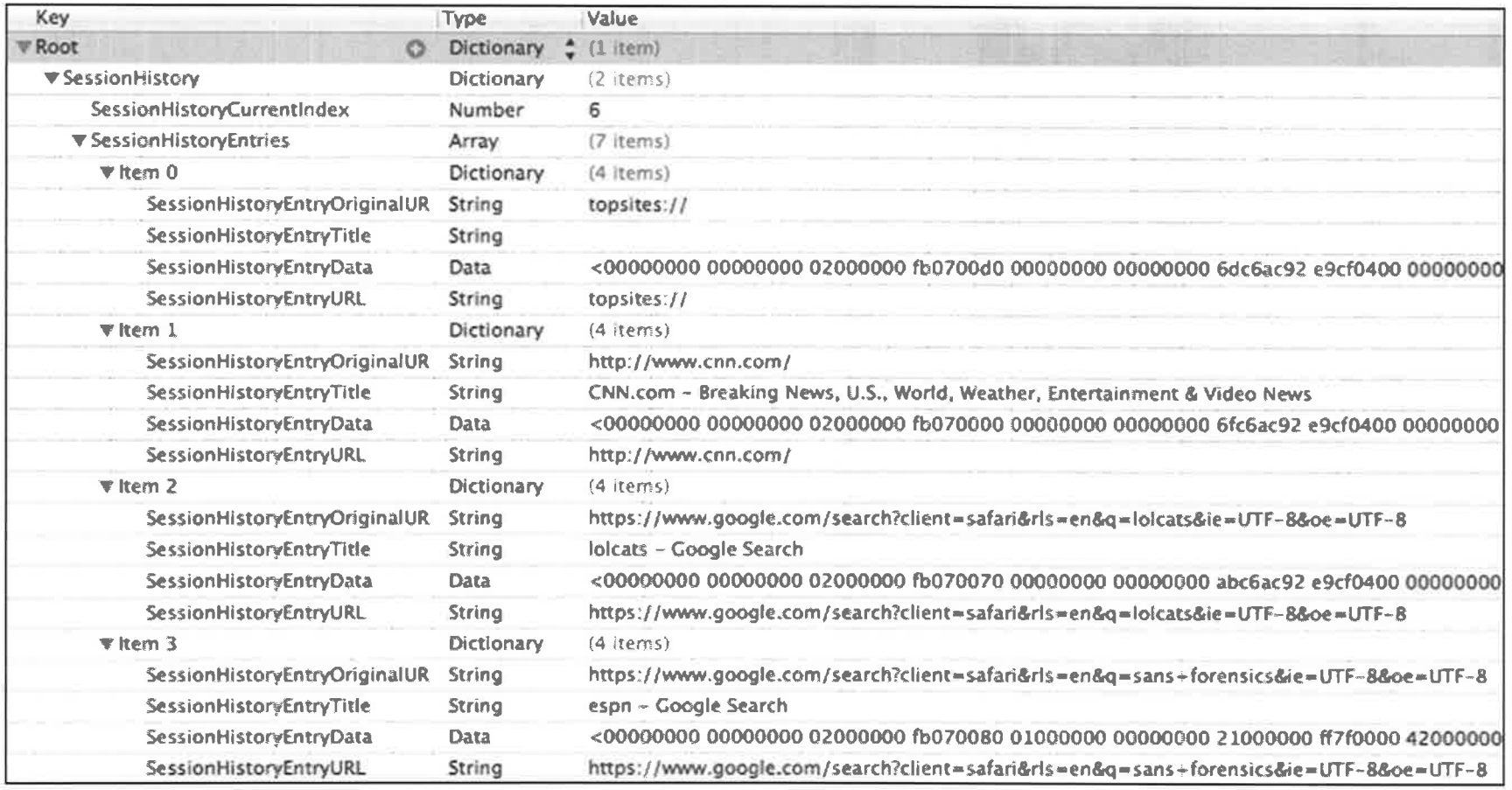
▼Root	Dictionary	(2 items)
SessionVersion	String	1.0
▼SessionWindows	Array	(1 item)
▼Item 0	Dictionary	(10 items)
WindowStateVersion	String	2.0
LocationBarHidden	Boolean	NO
▼TabStates	Array	(2 items)
▼Item 0	Dictionary	(5 items)
▼AncestorTabIdentifers	Array	(0 items)
TabIdentifier	Number	1
TabTitle	String	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)
▼AncestorTabIdentifers	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



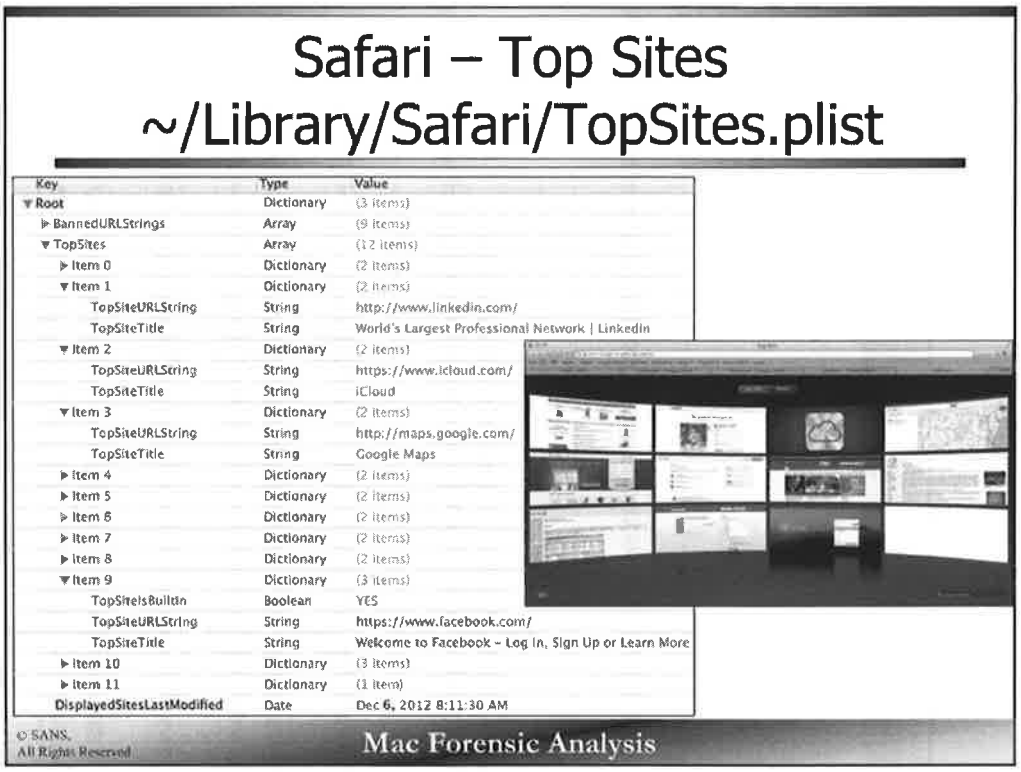
The binary property list in the `SessionState` key of the `LastSession.plist` can be extracted into another file and viewed. Unique to this property list, the first four bytes need to be removed (`0x00000002`) to get the `bplist00` file signature in the correct position.

This extracted binary property list contains the history of each tab in the `LastSession.plist`. This is how users can still go back/forth in the tab, even when Safari closed.

For example, when this property list was extracted, this tab previously visited CNN, performed a Google search for `lolcats`, and another search for `espn`.



Key	Type	Value
▼ Root	Dictionary	(1 item)
▼ SessionHistory	Dictionary	(2 items)
SessionHistoryCurrentIndex	Number	6
▼ SessionHistoryEntries	Array	(7 items)
▼ Item 0	Dictionary	(4 items)
SessionHistoryEntryOriginalUR	String	topsites://
SessionHistoryEntryTitle	String	
SessionHistoryEntryData	Data	<00000000 00000000 02000000 fb0700d0 00000000 00000000 6dc6ac92 e9cf0400 00000000
SessionHistoryEntryURL	String	topsites://
▼ Item 1	Dictionary	(4 items)
SessionHistoryEntryOriginalUR	String	http://www.cnn.com/
SessionHistoryEntryTitle	String	CNN.com - Breaking News, U.S., World, Weather, Entertainment & Video News
SessionHistoryEntryData	Data	<00000000 00000000 02000000 fb070000 00000000 00000000 6fc6ac92 e9cf0400 00000000
SessionHistoryEntryURL	String	http://www.cnn.com/
▼ Item 2	Dictionary	(4 items)
SessionHistoryEntryOriginalUR	String	https://www.google.com/search?client=safari&rls=en&q=lolcats&ie=UTF-8&oe=UTF-8
SessionHistoryEntryTitle	String	lolcats - Google Search
SessionHistoryEntryData	Data	<00000000 00000000 02000000 fb070070 00000000 00000000 abc6ac92 e9cf0400 00000000
SessionHistoryEntryURL	String	https://www.google.com/search?client=safari&rls=en&q=lolcats&ie=UTF-8&oe=UTF-8
▼ Item 3	Dictionary	(4 items)
SessionHistoryEntryOriginalUR	String	https://www.google.com/search?client=safari&rls=en&q=sans+forensics&ie=UTF-8&oe=UTF-8
SessionHistoryEntryTitle	String	espn - Google Search
SessionHistoryEntryData	Data	<00000000 00000000 02000000 fb070080 01000000 00000000 21000000 ff7f0000 42000000
SessionHistoryEntryURL	String	https://www.google.com/search?client=safari&rls=en&q=sans+forensics&ie=UTF-8&oe=UTF-8

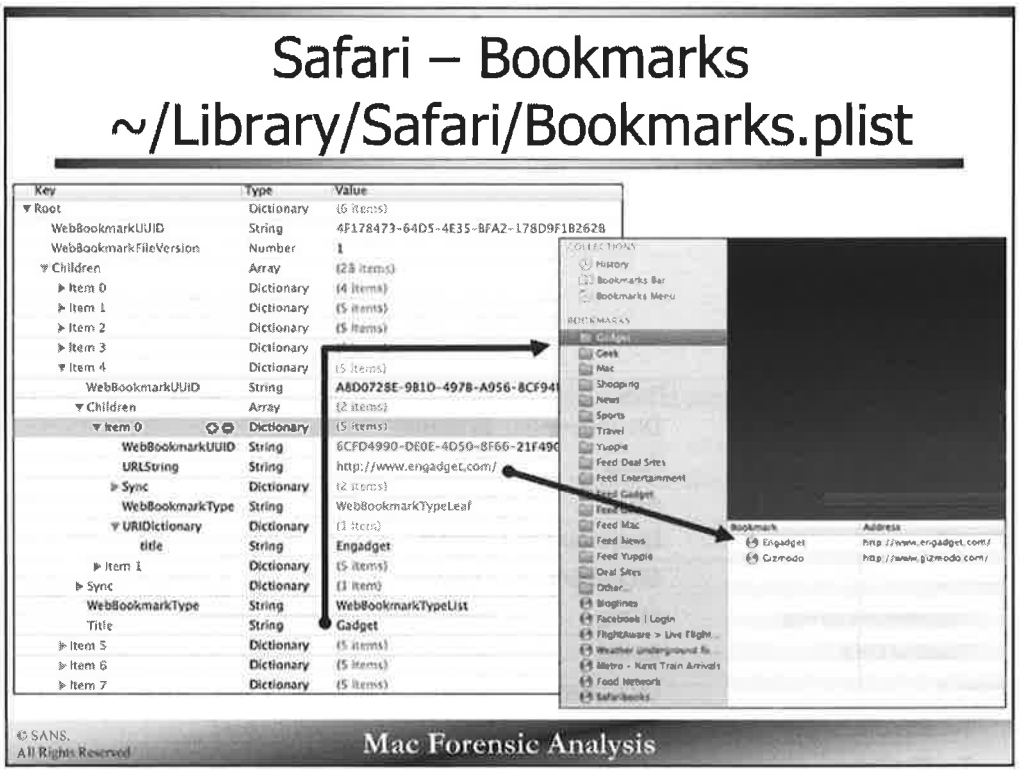


Safari's default view when creating a new tab, is the "`Top Sites`". This page holds the most frequently visited sites but can also be configured by the user. The `TopSites.plist` in the Safari directory contains the URL and title for each top site.

The forensic analyst should know that Safari has default entries in "built-into" Top Sites. In the screenshot above, Item 9 contains the key `TopSiteIsBuiltIn` – this is a default Top Site entry.

Sites may also be "pinned" by the user. The key `TopSiteIsPinned` will signify that a user has specially "pinned" a webpage to the Top Sites view.

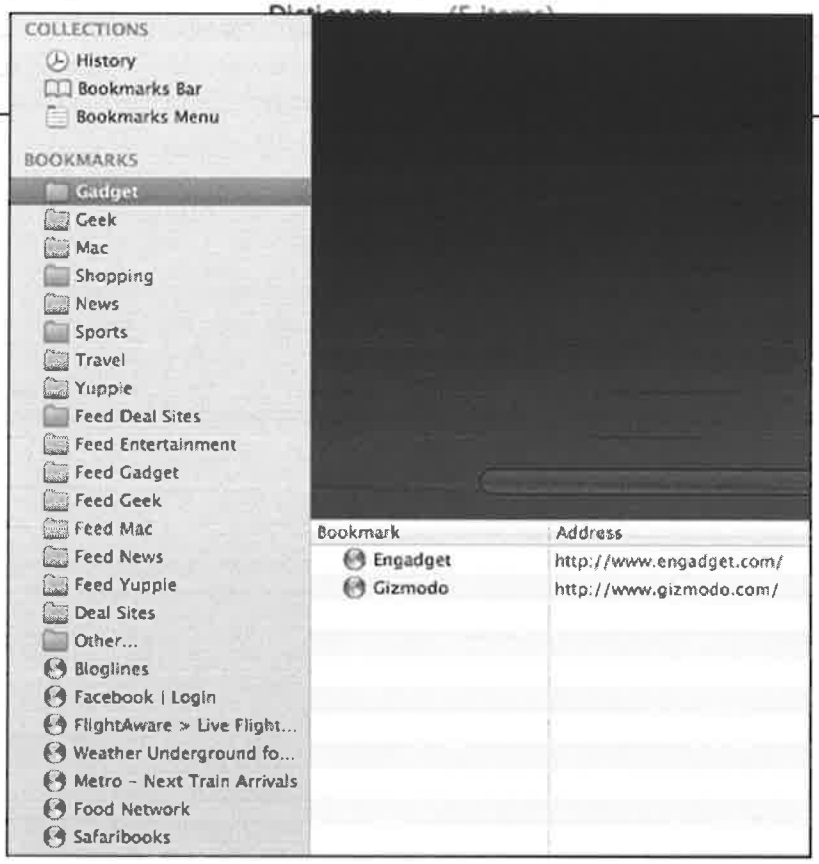
Key	Type	Value
▼ Root	Dictionary	(3 items)
▶ BannedURLStrings	Array	(9 items)
▼ TopSites	Array	(12 items)
▶ Item 0	Dictionary	(2 items)
▼ Item 1	Dictionary	(2 items)
TopSiteURLString	String	http://www.linkedin.com/
TopSiteTitle	String	World's Largest Professional Network | LinkedIn
▼ Item 2	Dictionary	(2 items)
TopSiteURLString	String	https://www.icloud.com/
TopSiteTitle	String	iCloud
▼ Item 3	Dictionary	(2 items)
TopSiteURLString	String	http://maps.google.com/
TopSiteTitle	String	Google Maps
▶ Item 4	Dictionary	(2 items)
▶ Item 5	Dictionary	(2 items)
▶ Item 6	Dictionary	(2 items)
▶ Item 7	Dictionary	(2 items)
▶ Item 8	Dictionary	(2 items)
▼ Item 9	Dictionary	(3 items)
TopSiteIsBuiltin	Boolean	YES
TopSiteURLString	String	https://www.facebook.com/
TopSiteTitle	String	Welcome to Facebook - Log In, Sign Up or Learn More
▶ Item 10	Dictionary	(3 items)
▶ Item 11	Dictionary	(1 item)
DisplayedSitesLastModified	Date	Dec 6, 2012 8:11:30 AM



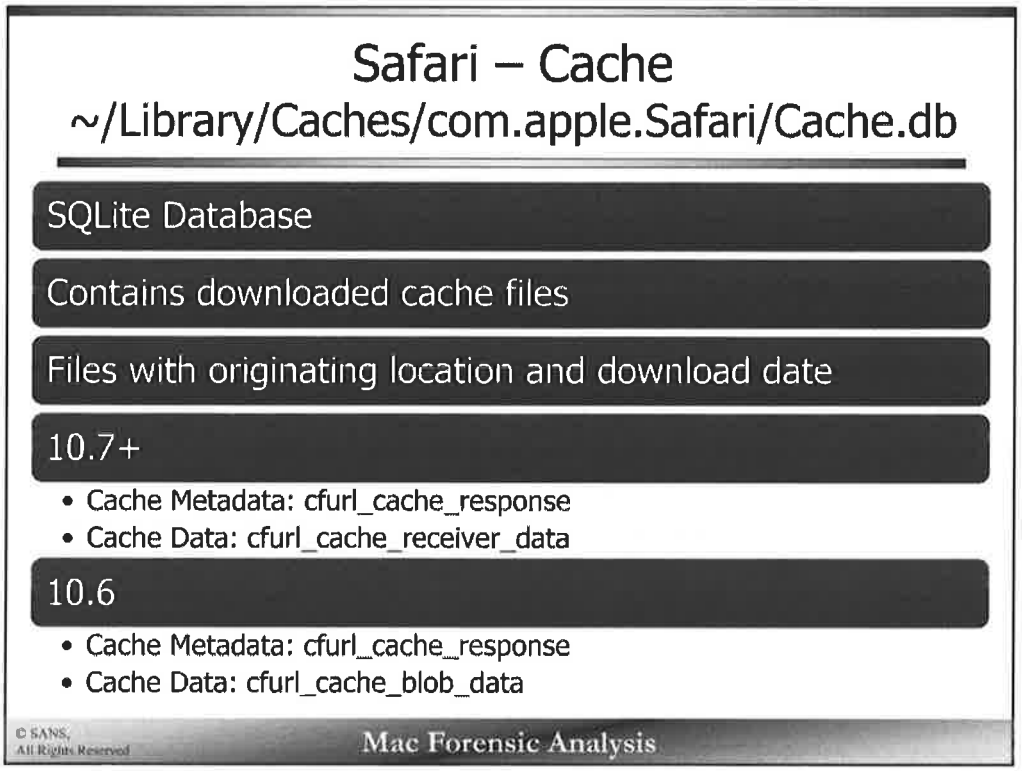
The Safari Bookmarks are saved in `Bookmarks.plist`. This property list is heavily nested and may take some time to go through. Each Item may contain multiple children bookmarks.

In the screenshots above, `Item 4` contains the Bookmark folder named "`Gadget`", this folder contains two children `Item 0` and `Item 1`. Each bookmark item contains a `WebBookmarkUUID` GUID, the URL, and the title of the bookmark.

Key	Type	Value
▼ Root	Dictionary	(6 items)
WebBookmarkUUID	String	4F178473-64D5-4E35-8FA2-178D9F1B262B
WebBookmarkFileVersion	Number	1
▼ Children	Array	(28 items)
▶ Item 0	Dictionary	(4 items)
▶ Item 1	Dictionary	(5 items)
▶ Item 2	Dictionary	(5 items)
▶ Item 3	Dictionary	(6 items)
▼ Item 4	Dictionary	(5 items)
WebBookmarkUUID	String	A8D0728E-9B1D-497B-A956-8CF94F6BED61
▼ Children	Array	(2 items)
▼ Item 0	Dictionary	(5 items)
WebBookmarkUUID	String	6CFD4990-DE0E-4D50-8F66-21F49036A68B
URLString	String	http://www.engadget.com/
▶ Sync	Dictionary	(2 items)
WebBookmarkType	String	WebBookmarkTypeLeaf
▼ URIDictionary	Dictionary	(1 item)
title	String	Engadget
▶ Item 1	Dictionary	(5 items)
▶ Sync	Dictionary	(1 item)
WebBookmarkType	String	WebBookmarkTypeList
Title	String	Gadget
▶ Item 5		
▶ Item 6		
▶ Item 7		

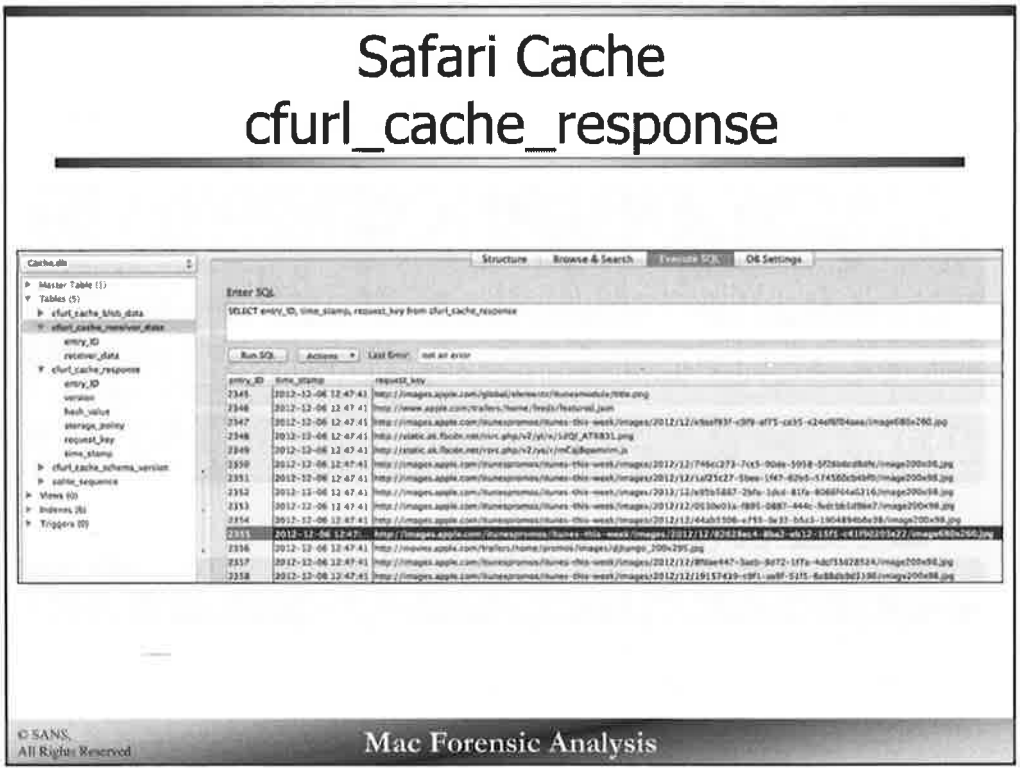


Safari does a unique function where it periodically saves screen captures of the web browser screen. The Webpage Previews directory may contain many JPEG and PNG images of webpages that were once viewed. This can be useful to see what a user may have been looking at from a snapshot in time.



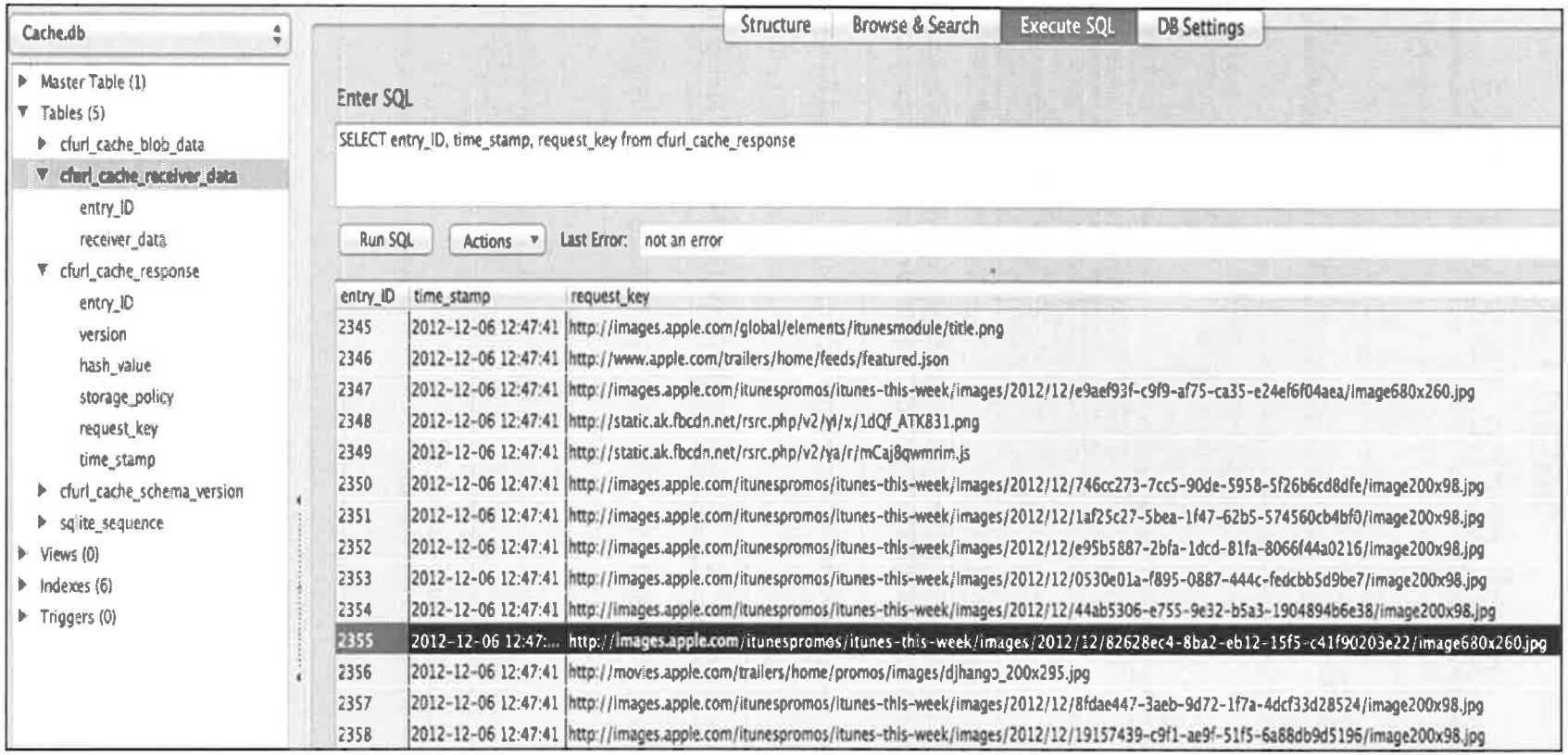
The SQLite database `Cache.db` located in the `~/Library/Caches/com.apple.Safari/` directory contains the downloaded cache files. Each file has a corresponding location and download date.

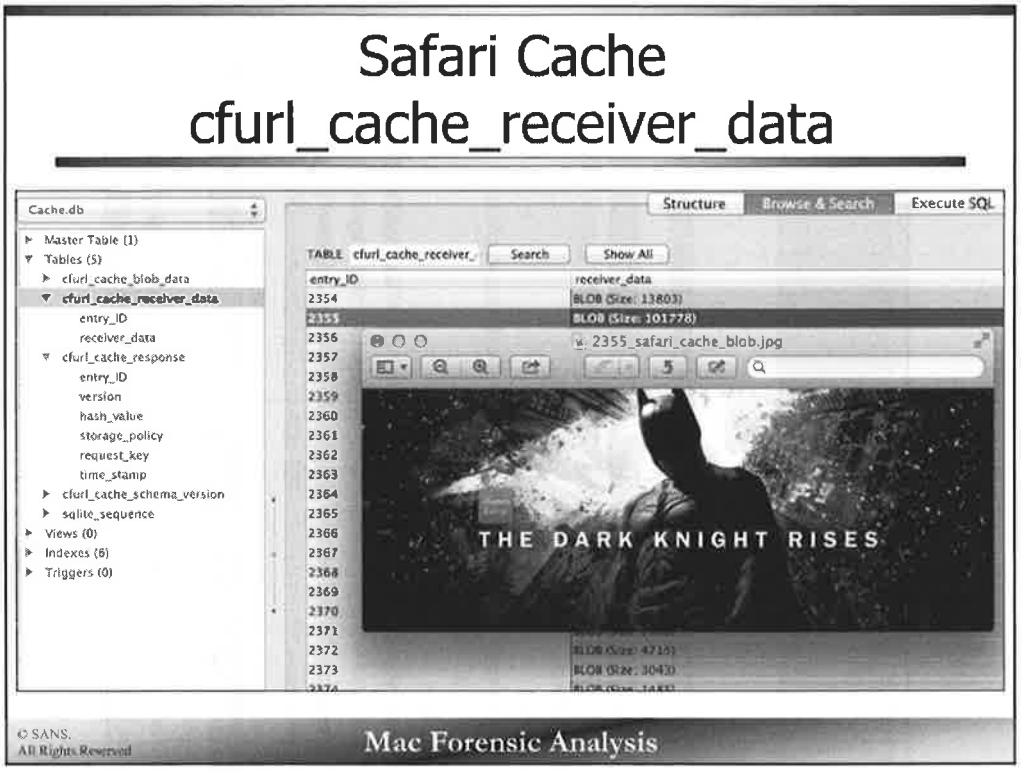
Databases for 10.6 and 10.7+ systems differ slightly, but each contains a metadata table and a data table in the database.



The `cfurl_cache_response` table contains cache file metadata including the file cached and its corresponding timestamp.

The screenshot shown is using the SQLite Manager plugin for the Firefox browser to view the SQLite database. The SQL query "`SELECT entry_ID, time_stamp, request_key from cfurl_cache_response;`" was used to display this information shown.


Cache.db
Structure
Browse & Search
Execute SQL
DB Settings
Master Table (1)
Tables (5)
cfurl_cache_blob_data
cfurl_cache_receiver_data
entry_ID
receiver_data
cfurl_cache_response
entry_ID
version
hash_value
storage_policy
request_key
time_stamp
cfurl_cache_schema_version
sqlite_sequence
Views (0)
Indexes (6)
Triggers (0)
Enter SQL
SELECT entry_ID, time_stamp, request_key from cfurl_cache_response
Run SQL
Actions
Last Error: not an error
entry_ID | time_stamp | request_key
2345 | 2012-12-06 12:47:41 | http://images.apple.com/global/elements/itunesmodule/title.png
2346 | 2012-12-06 12:47:41 | http://www.apple.com/trailers/home/feeds/featured.json
2347 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/e9aef93f-c9f9-af75-ca35-e24ef6f04aea/image680x260.jpg
2348 | 2012-12-06 12:47:41 | http://static.ak.fbcdn.net/rsrc.php/v2/yl/x/1dQf_ATK831.png
2349 | 2012-12-06 12:47:41 | http://static.ak.fbcdn.net/rsrc.php/v2/ya/r/mCaj8qwmrim.js
2350 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/746cc273-7cc5-90de-5958-5f26b6cd8dfe/image200x98.jpg
2351 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/1af25c27-5bea-1f47-62b5-574560cb4bf0/image200x98.jpg
2352 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/e95b5887-2bfa-1dcd-81fa-8066f44a0216/image200x98.jpg
2353 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/0530e01a-f895-0887-444c-fedcbb5d9be7/image200x98.jpg
2354 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/44ab5306-e755-9e32-b5a3-1904894b6e38/image200x98.jpg
2355 | 2012-12-06 12:47:... | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/82628ec4-8ba2-eb12-15f5-c41f90203e22/image680x260.jpg
2356 | 2012-12-06 12:47:41 | http://movies.apple.com/trailers/home/promos/images/djhango_200x295.jpg
2357 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/8fdae447-3aeb-9d72-1f7a-4dcf33d28524/image200x98.jpg
2358 | 2012-12-06 12:47:41 | http://images.apple.com/itunespromos/itunes-this-week/images/2012/12/19157439-c9f1-ae9f-51f5-6a88db9d5196/image200x98.jpg

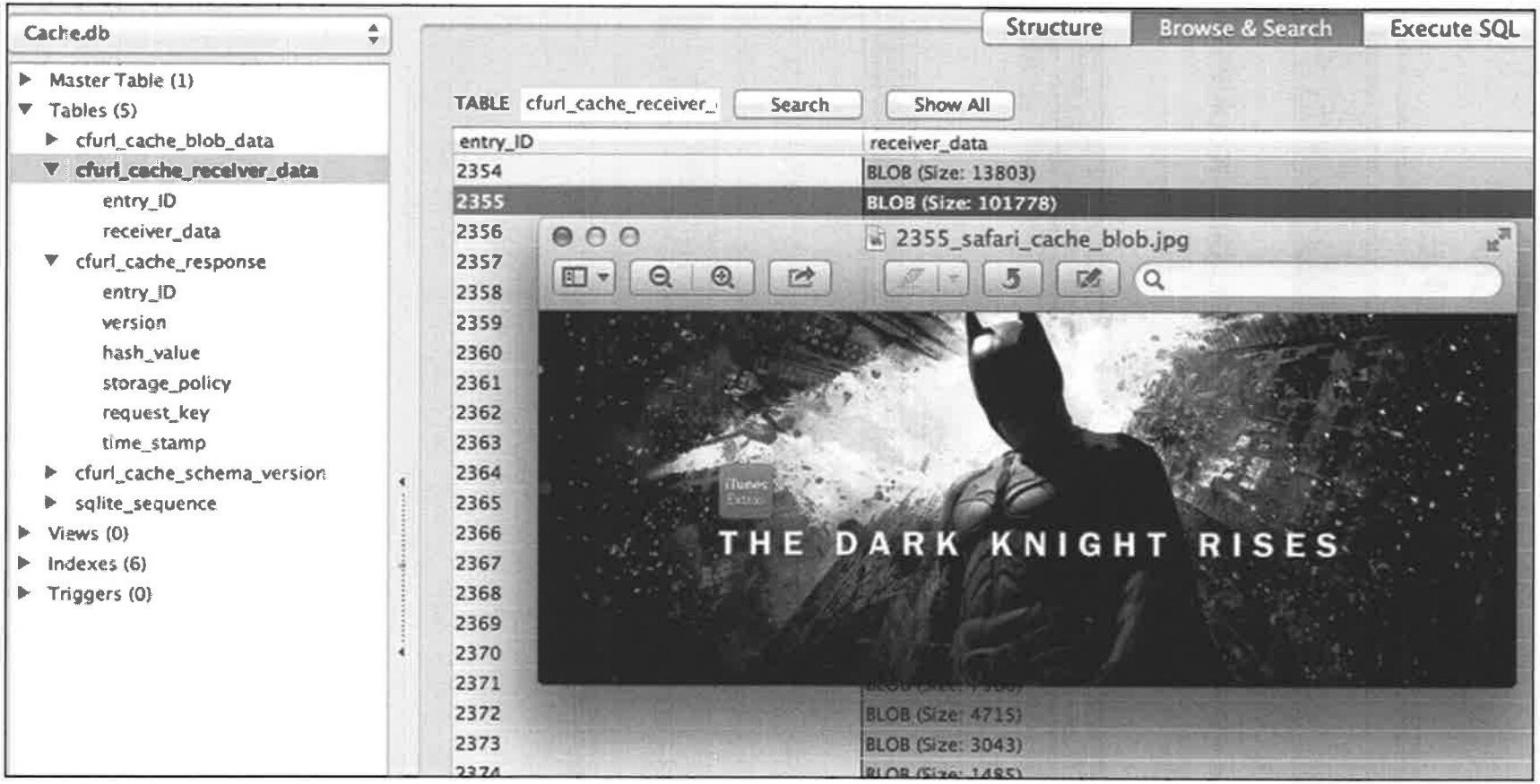


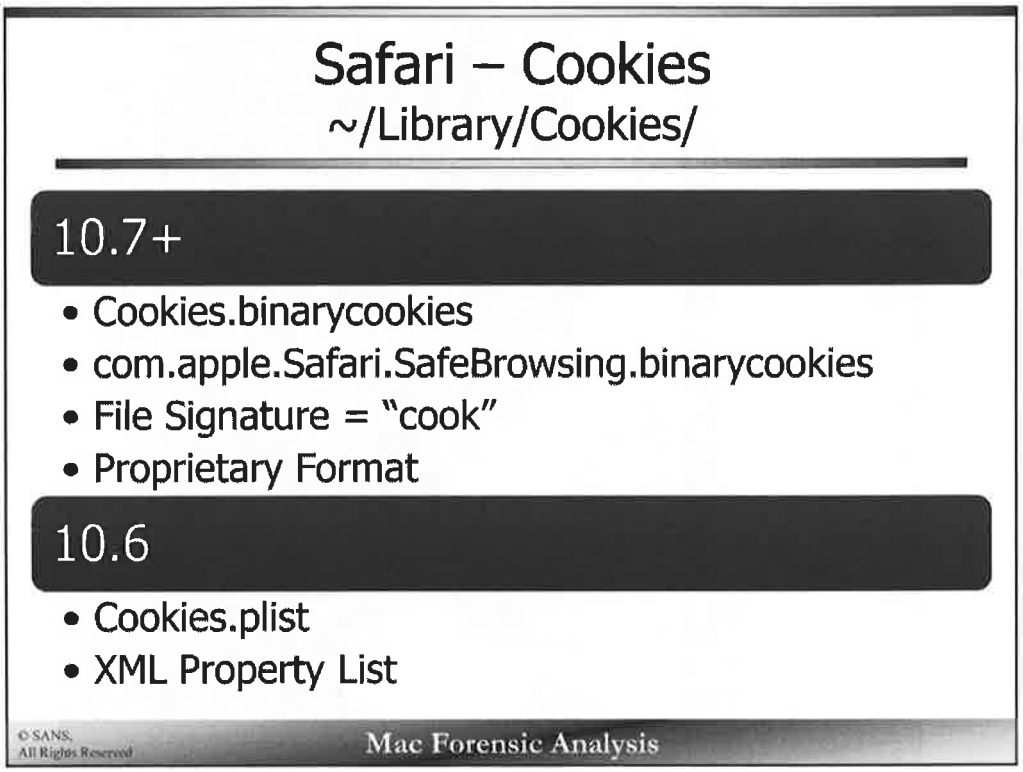
The `cfurl_cache_receiver_data` table contains the cached file. The cached file can be matched up with its metadata by using the `entry_ID` number.

The screenshot shown is using the SQLite Manager plugin for the Firefox browser to view the SQLite database.

You may execute this SQL command to correlate this cache information. The following command was provided by Antonio Merola:

```
select cfurl_cache_response.time_stamp, cfurl_cache_response.request_key,
cfurl_cache_receiver_data.receiver_data from cfurl_cache_receiver_data,
cfurl_cache_response where cfurl_cache_response.entry_ID ==
cfurl_cache_receiver_data.entry_ID order by
cfurl_cache_response.time_stamp;
```

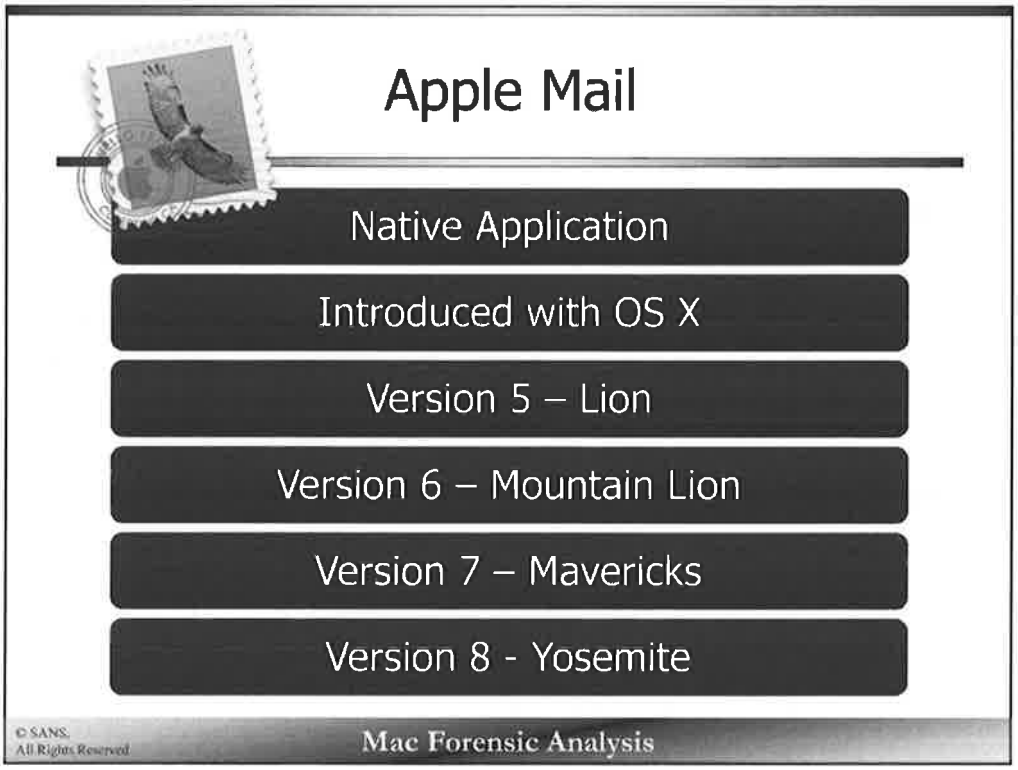

Cache.db
Structure
Browse & Search
Execute SQL
Master Table (1)
Tables (5)
cfurl_cache_blob_data
cfurl_cache_receiver_data
entry_ID
receiver_data
cfurl_cache_response
entry_ID
version
hash_value
storage_policy
request_key
time_stamp
cfurl_cache_schema_version
sqlite_sequence
Views (0)
Indexes (6)
Triggers (0)
TABLE cfurl_cache_receiver_
Search
Show All
entry_ID
receiver_data
2354
BLOB (Size: 13803)
2355
BLOB (Size: 101778)
2356
2357
2358
2359
2360
2361
2362
2363
2364
2365
2366
2367
2368
2369
2370
2371
2372
BLOB (Size: 4715)
2373
BLOB (Size: 3043)
2355_safari_cache_blob.jpg
THE DARK KNIGHT RISES



Safari stores cookies in a separate directory from everything else, `~/Library/Cookies/`.

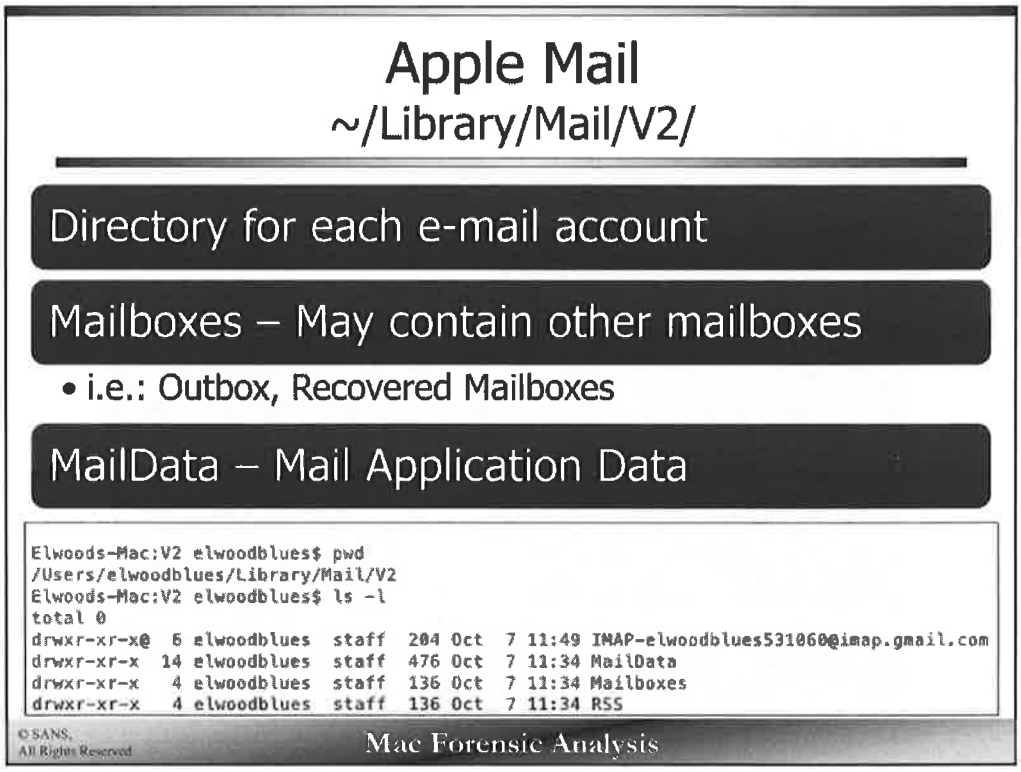
On 10.7+ systems the cookies are stored in the `Cookies.binarycookies` and `com.apple.Safari.SafeBrowsing.binarycookies`. This file starts with the file header "`cook`". The file format for these binary cookies are a proprietary format.

On 10.6 systems the cookies are stored in a plaintext XML property list file.



The native e-mail 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!

In general, you will find a different version of Mail on the last three versions of OS X. The inner workings of the last three versions all work nearly the same way.

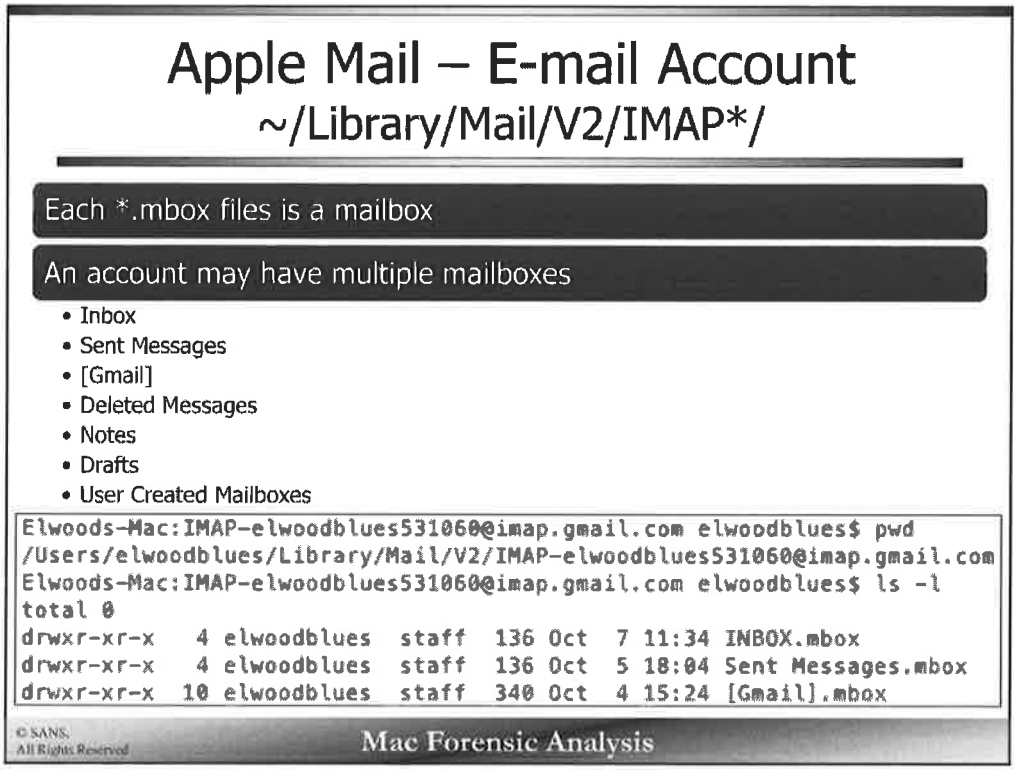


Mail on 10.7+ systems are located in the `~/Library/Mail/V2/` (10.6 Mail is located in `~/Library/Mail/`, there is no "V2" or Version 2 directory). This directory contains a sub-directory for each mailbox that the user has. The screenshot above shows one e-mail account, an IMAP Gmail account for elwoodblues531060@gmail.com.

The Mailboxes directory contains system mailboxes usually including a "`Deleted Messages.mbox`" and/or "`Outbox.mbox`".

The `MailData` contains the Mail Application Data, which will include preferences such as:

- Smart Mailboxes – Customized Mailboxes
- Syncing Preferences
- E-mail Rules – Automatically move messages as they come in.
- Signatures
- Account Information



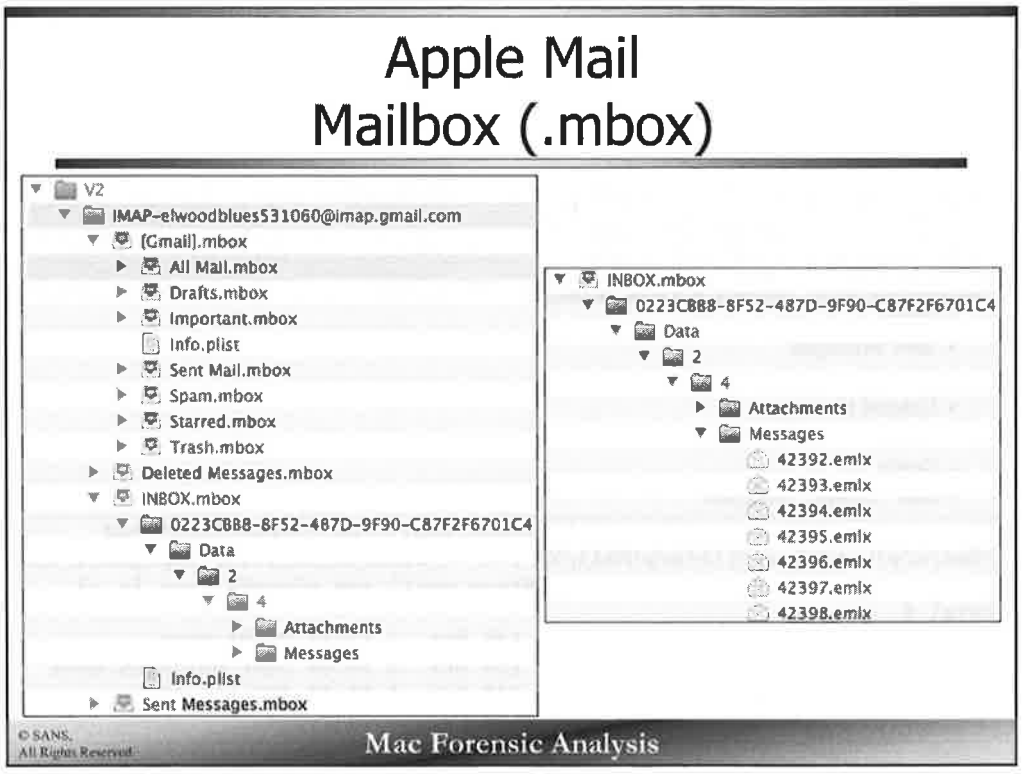
Each Mailbox Account directory (`IMAP-elwoodblues531060@imap.gmail.com`) contains one or more mailbox (`mbox`) directories.

The screenshot shows three mailbox (`mbox`) directories.

- `INBOX.mbox`
- `Sent Messages.mbox`
- `[Gmail].box`

POP accounts will be distinguished with "`POP`" instead of `IMAP` in the directory name.

Note: 10.6 uses `imapbox` rather than `mbox` for its IMAP e-mail accounts.



Each mailbox contains one `Info.plist` file containing the number of unread messages (`IMAPMailboxUnseenCount` Key), display settings and the mailbox name.

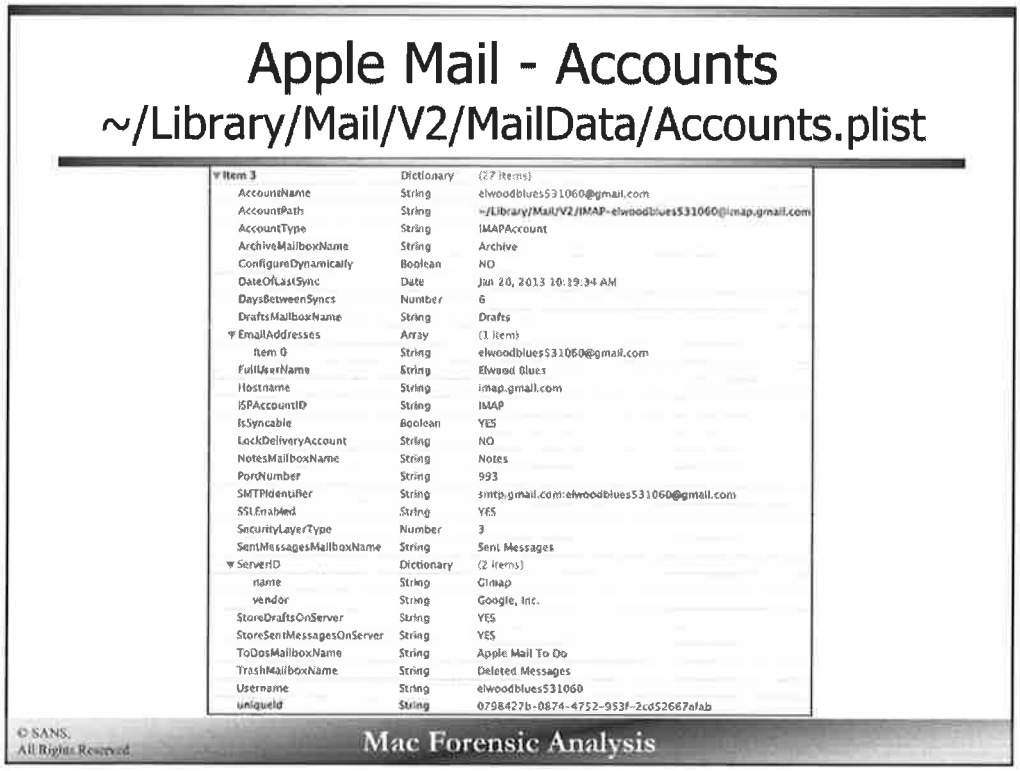
The GUID directory contains the raw e-mail messages (`emlx`) and e-mail metadata in the Messages directory.

- Messages – Contains the raw e-mail messages (`.emlx`), with an appended property list containing message metadata.
- Attachments – Contains the message file attachments.

The metadata may include the following items:

- Colorization
- Mailbox
- Message Sent Timestamp
- E-mail Subject

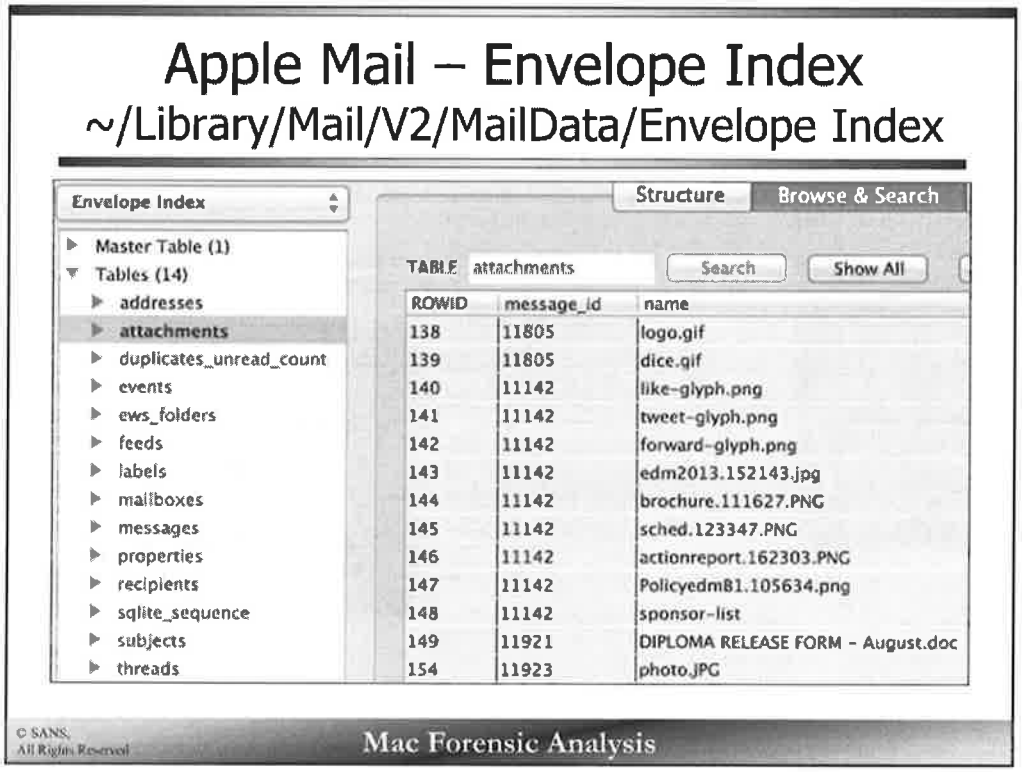
Note: 10.6 does not use a GUID and goes directly to a directory containing the `Info.plist` and a Messages directory.



The Accounts property list contains the e-mail account configuration for the system. This includes:

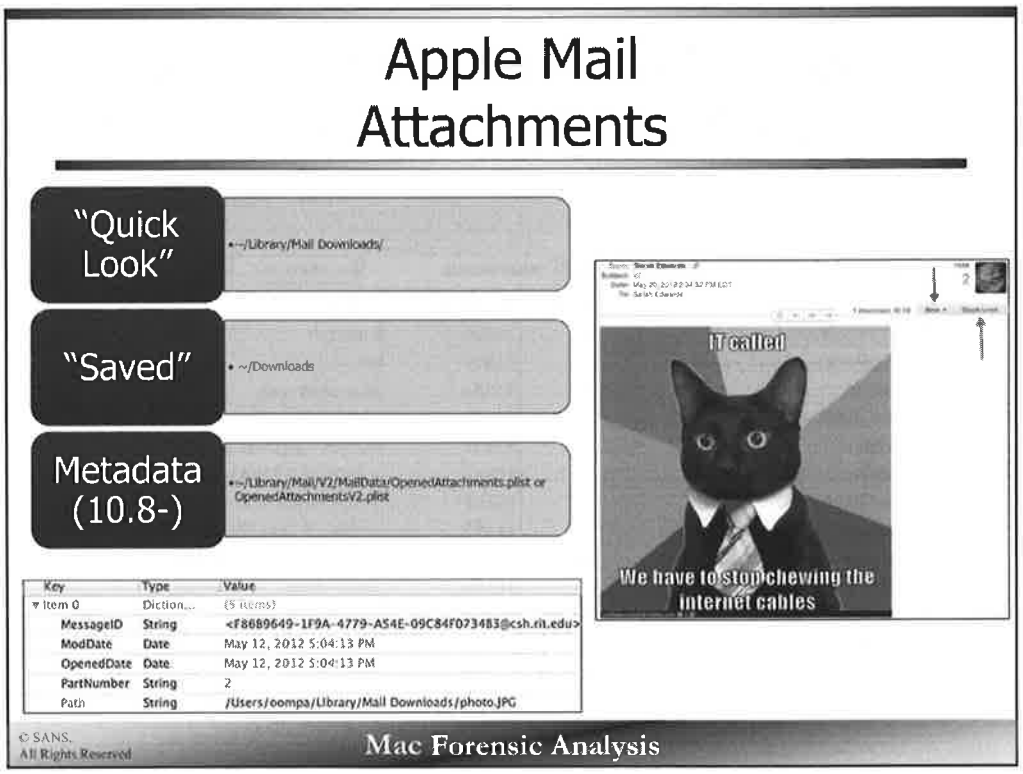
- Account Name
- SMTP Hostname
- Port/SSL/Authentication
- Sync Timestamps
- Mail Ports
- Account Names

▼ Item 3	Dictionary	(27 items)
AccountName	String	elwoodblues531060@gmail.com
AccountPath	String	~/Library/Mail/V2/IMAP-elwoodblues531060@imap.gmail.com
AccountType	String	IMAPAccount
ArchiveMailboxName	String	Archive
ConfigureDynamically	Boolean	NO
DateOfLastSync	Date	Jan 20, 2013 10:19:34 AM
DaysBetweenSyncs	Number	6
DraftsMailboxName	String	Drafts
▼ EmailAddresses	Array	(1 item)
Item 0	String	elwoodblues531060@gmail.com
FullUserName	String	Elwood Blues
Hostname	String	imap.gmail.com
ISPAccountID	String	IMAP
IsSyncable	Boolean	YES
LockDeliveryAccount	String	NO
NotesMailboxName	String	Notes
PortNumber	String	993
SMTPIdentifier	String	smtp.gmail.com:elwoodblues531060@gmail.com
SSLEnabled	String	YES
SecurityLayerType	Number	3
SentMessagesMailboxName	String	Sent Messages
▼ ServerID	Dictionary	(2 items)
name	String	Gimap
vendor	String	Google, Inc.
StoreDraftsOnServer	String	YES
StoreSentMessagesOnServer	String	YES
ToDosMailboxName	String	Apple Mail To Do
TrashMailboxName	String	Deleted Messages
Username	String	elwoodblues531060
uniqueId	String	0798427b-0874-4752-953f-2cd52667afab



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

- The `addresses` table contains all the indexed e-mail addresses and associated contact name.
- The `attachments` table (shown above) 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 e-mail message including; To, From, Subject, e-mail timestamps, if the message was read or not.
- The `subjects` table contains the e-mail subject for each e-mail message.



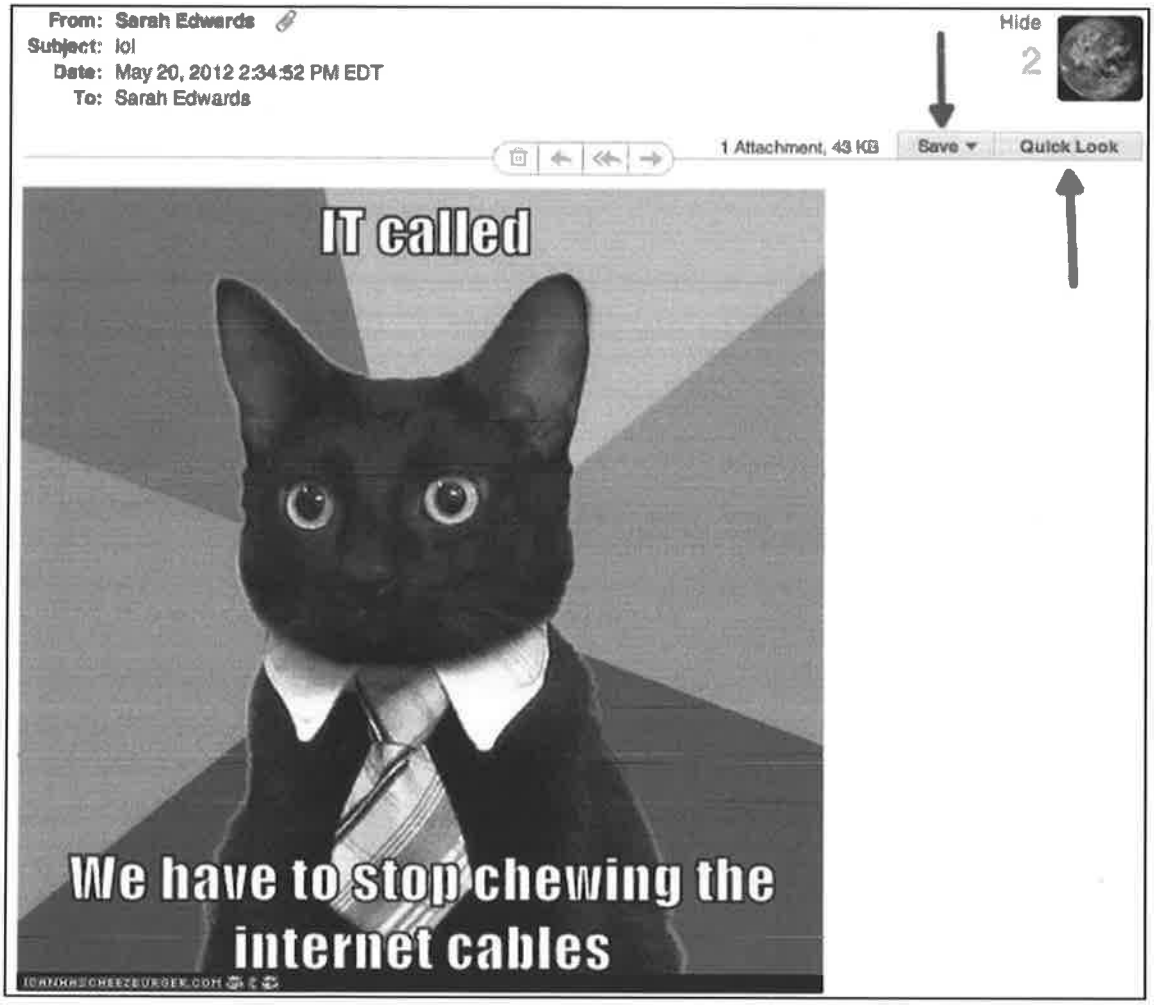
A user can view an attachment in a couple ways:

- The "`Save`" button will save the attachment in the default downloads directory, very likely `~/Downloads`.

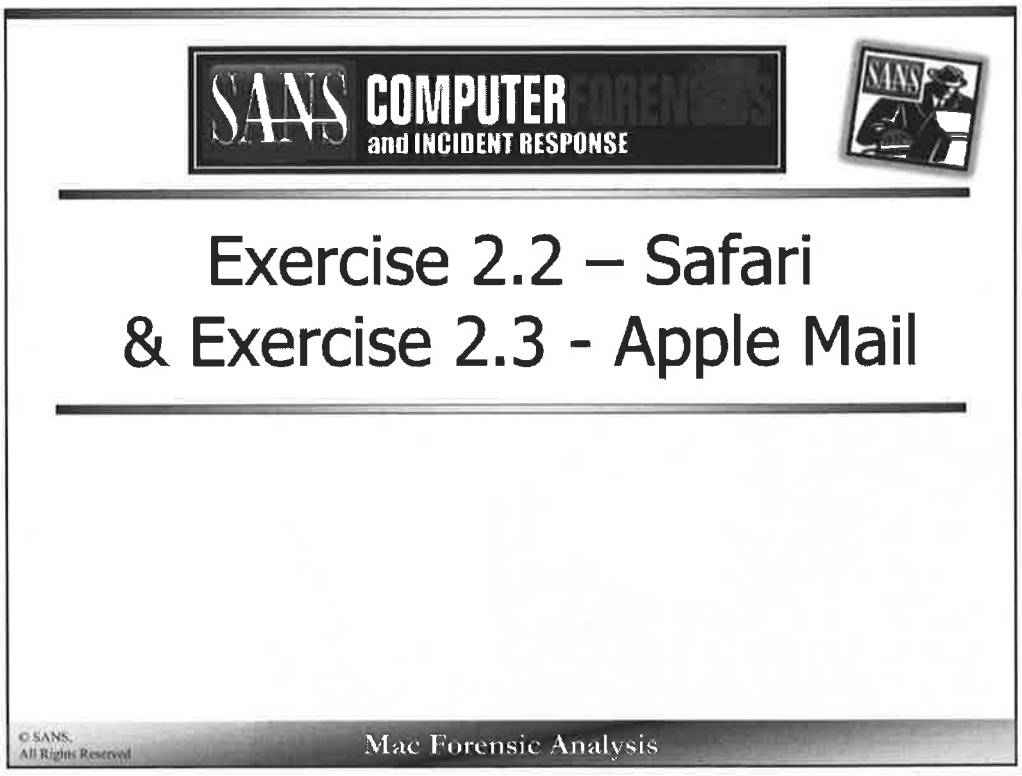
- The "`Quick Look`" button opens the attachment for the viewer to see quickly, and saves the attachment in the `~/Library/Mail Downloads/` directory. The sandbox directory may also be used, `~/Library/Containers/com.apple.mail/Data/Library/Mail Downloads/`.

Each attachment saved is recorded in the `OpenedAttachments.plist` file (`OpenedAttachmentsV2.plist` on 10.8). This property list contains the modification, opened date, and path of the file. These dates are saved in the local time of the system. For systems uses 10.8 and older, if this file does not exist, the user likely did not open an attachment using Apple Mail.

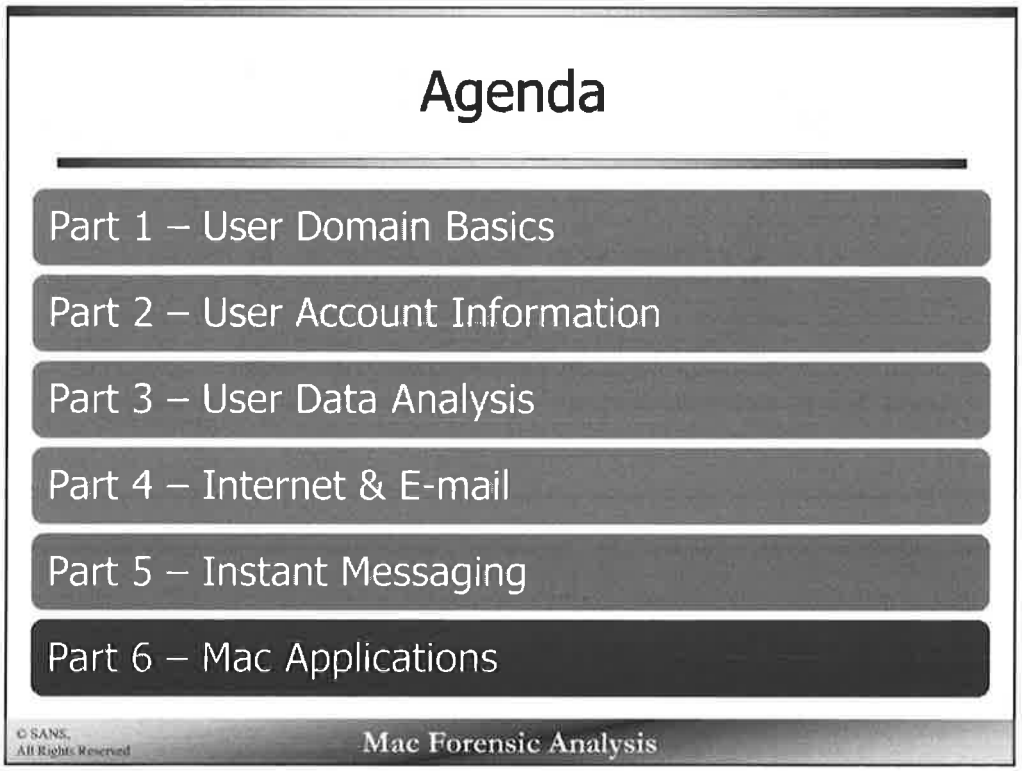
Sadly, This file does not appear to be used on 10.9+ systems.



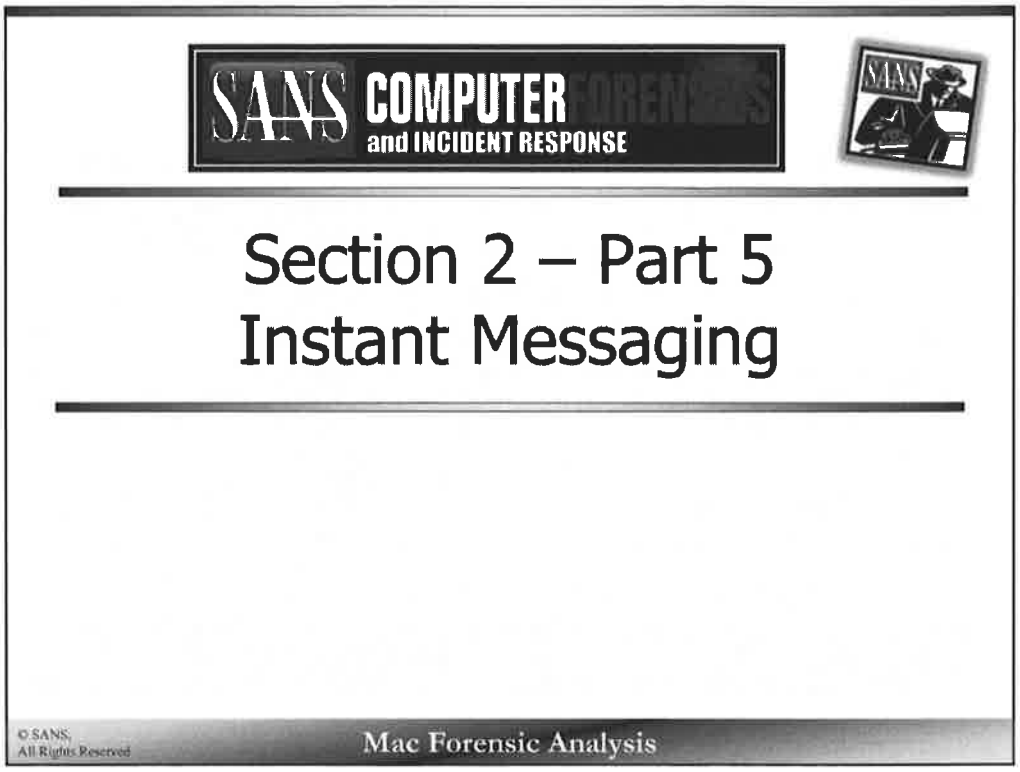
Key	Type	Value
▼ Item 0	Diction...	(5 items)
MessageID	String	<F8689649-1F9A-4779-A54E-09C84F0734B3@csh.rit.edu>
ModDate	Date	May 12, 2012 5:04:13 PM
OpenedDate	Date	May 12, 2012 5:04:13 PM
PartNumber	String	2
Path	String	/Users/oompa/Library/Mail Downloads/photo.JPG



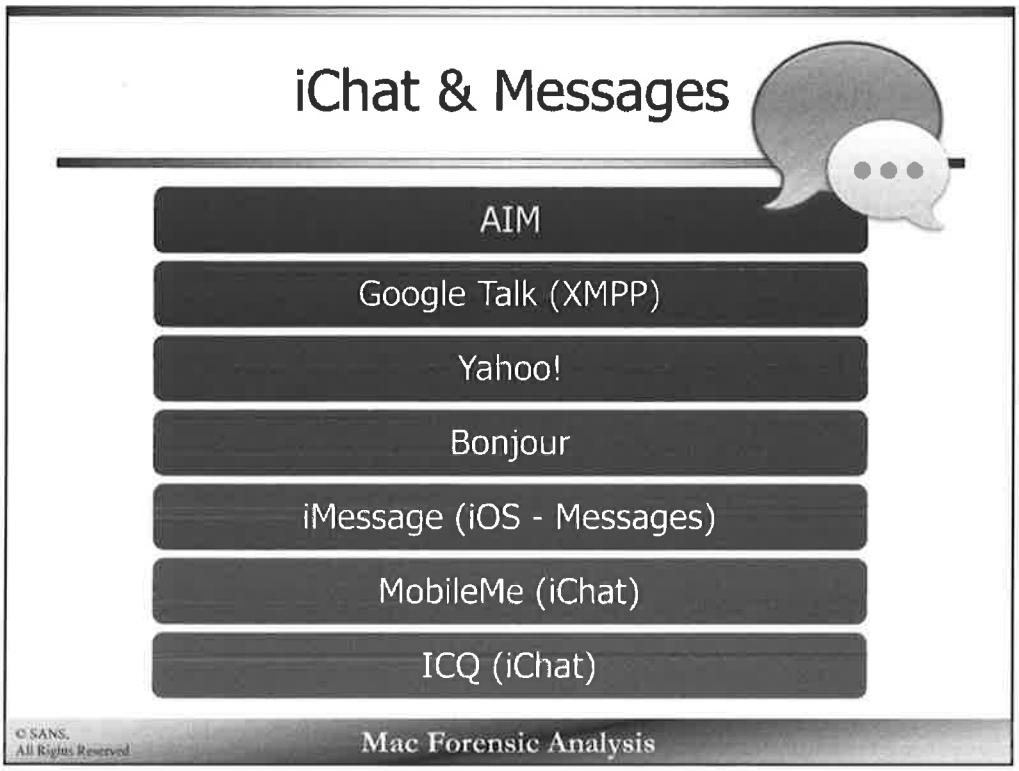
This page intentionally left blank.



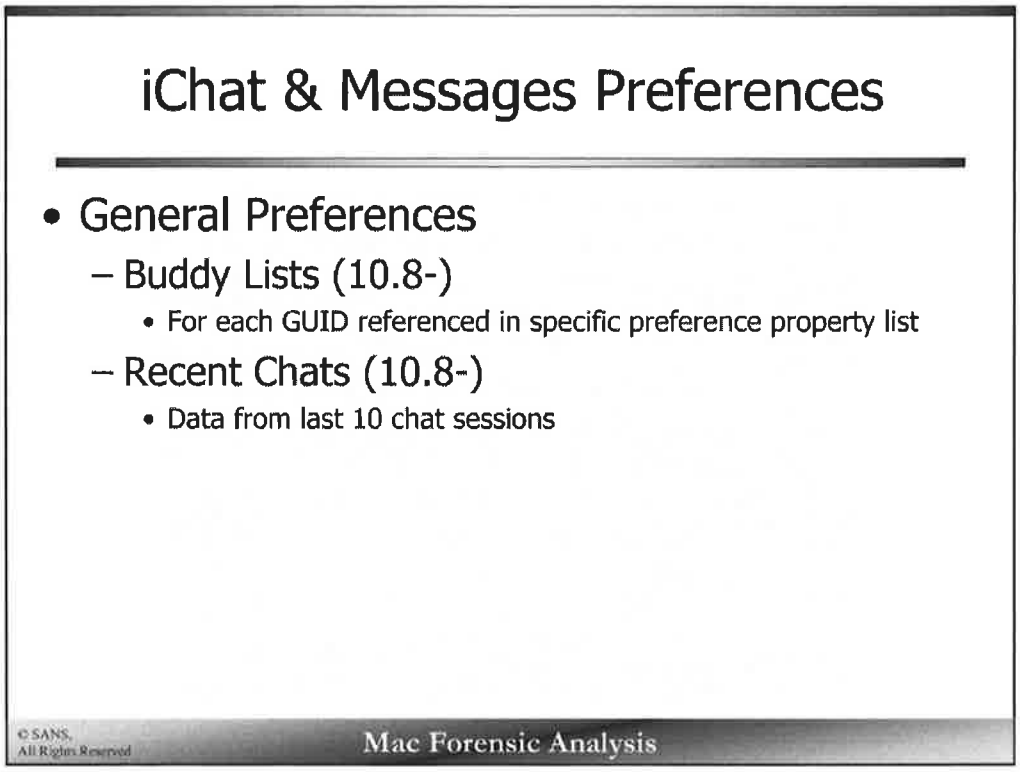
This page intentionally left blank.



This page intentionally left blank.

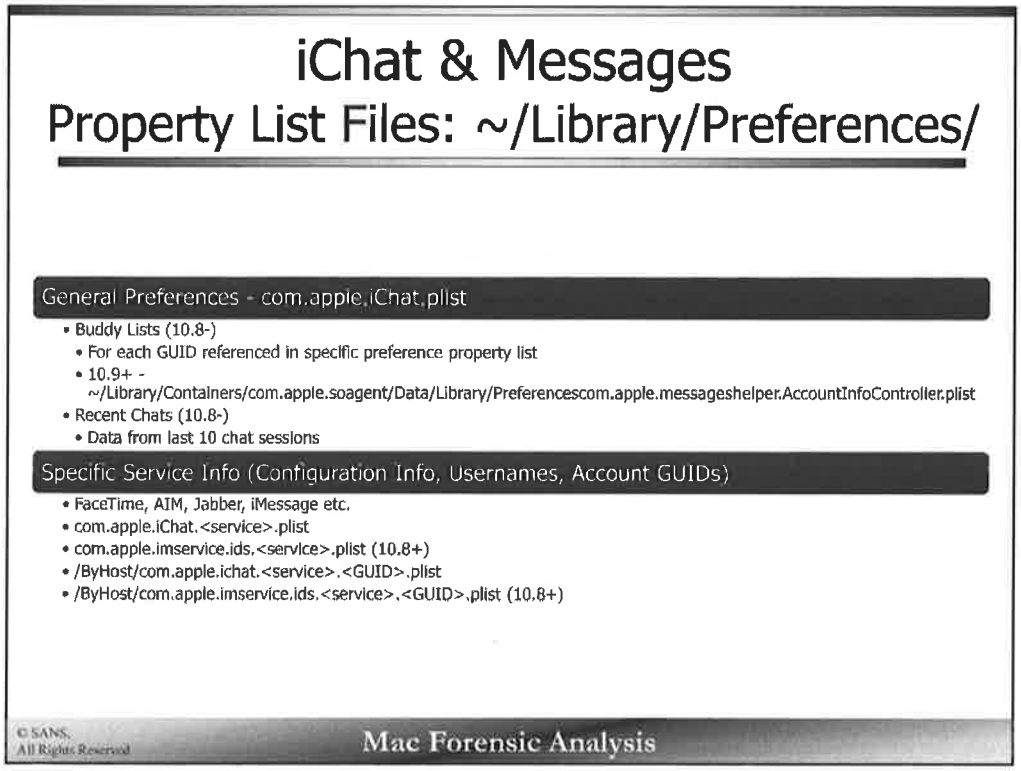


iChat (10.6 & 1.07) and its reincarnation, Messages (10.8) are the native instant messaging applications of OS X. These applications can be used with a variety of instant messaging programs and protocols. Messages replaced iChat in 10.8 (Beta was available earlier) and introduced integration with iMessage (iOS) and FaceTime.



The general preferences for iChat and Messages is located in the `com.apple.iChat.plist`. This property list contains a huge compilation of preferences and configuration data, shown on the next page. Some of the highlights include:

- "Away" & "Available" Messages (Also in `com.apple.iChat.StatusMessages.plist`)
- Account GUIDs
- Buddy Lists (`HandleCompletionCache` Key)
- Recent Chats



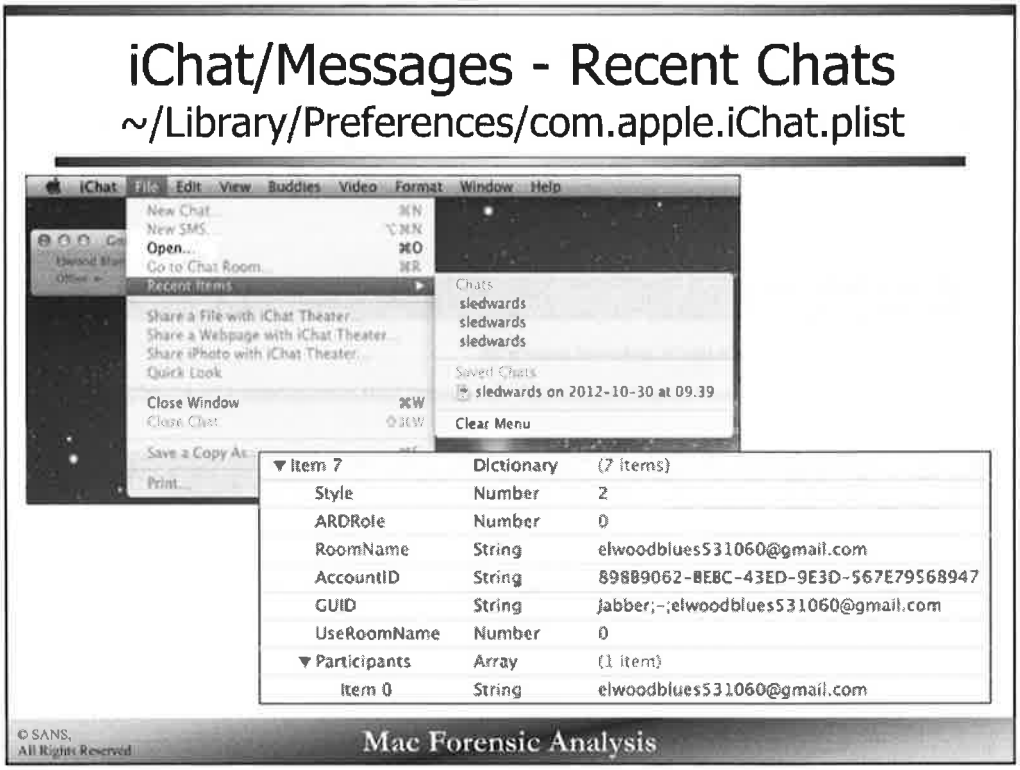
The general preferences for iChat and Messages is located in the `com.apple.iChat.plist`. This property list contains a huge compilation of preferences and configuration data, shown on the next page. Some of the highlights include:

- "Away" & "Available" Messages (Also in `com.apple.iChat.StatusMessages.plist`)
- Account GUIDs
- Buddy Lists (`IHandleCompletionCache` Key)
- Recent Chats

There may be plenty of other files of interest in the `~/Library/Preferences` directory. These files contain account usernames, configuration information and the GUID associated with the account that all account information is associated with.

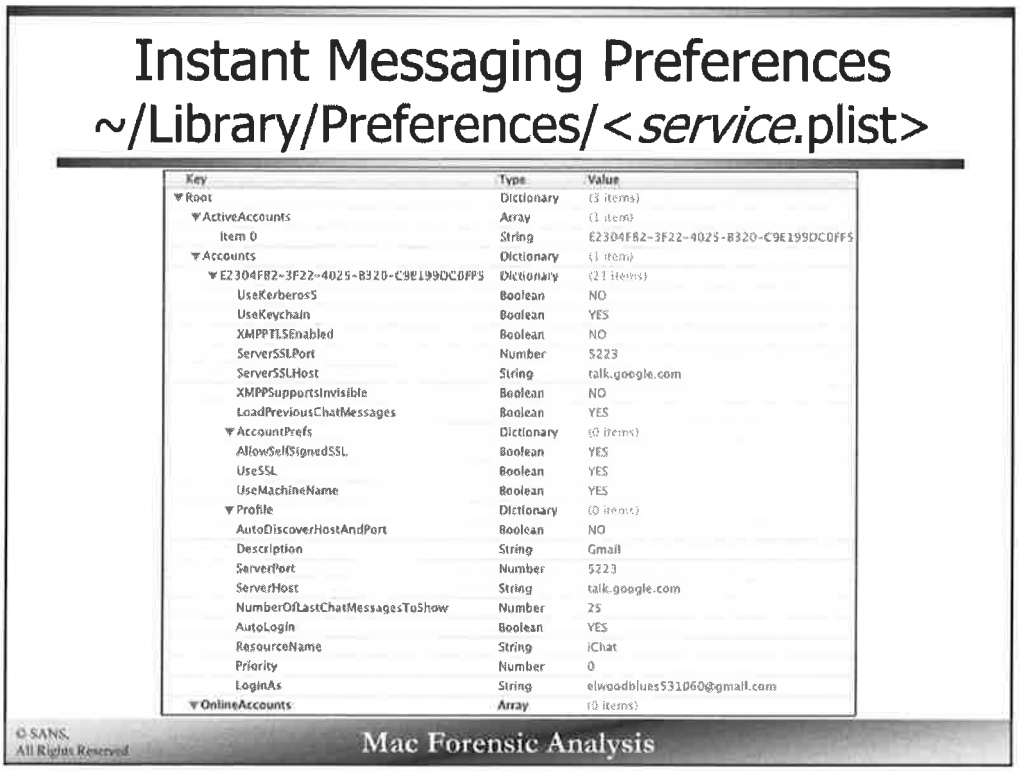
Many of these files are named different depending on the version of OS X the user is using. On newer systems files names `com.apple.iChat.<service>.plist` and `com.apple.imservice.ids.<service>.plist` will include nearly the same information that the same `com.apple.iChat.<service>.plist` files held.

Each service has specific information associated with it, the <service> may be everything from FaceTime, iMessage, Jabber, AIM, or any other supported instant messaging protocol.



The `RecentChats` key in the `com.apple.iChat.plist` property list will contain the last ten chats, and whether the user was using Google Talk, AIM, or another protocol. In the GUI these recent chats are shown in the `File | Recent Items` menu.

Each item in `RecentChats` contains the GUID of the specific account in AccountID (AIM, Google Talk, ICQ, etc.). The room names depends on the protocol used, in Google Chat seen above, it labels the room by the Google account, AIM will label with a random chat number such as "`chat9042687188894940l`" or by the chat recipients handle. The `Participants` key holds the account name or handle of each participant in the chat, usually just the other chat recipient, but could be another contact if it is a group chat.

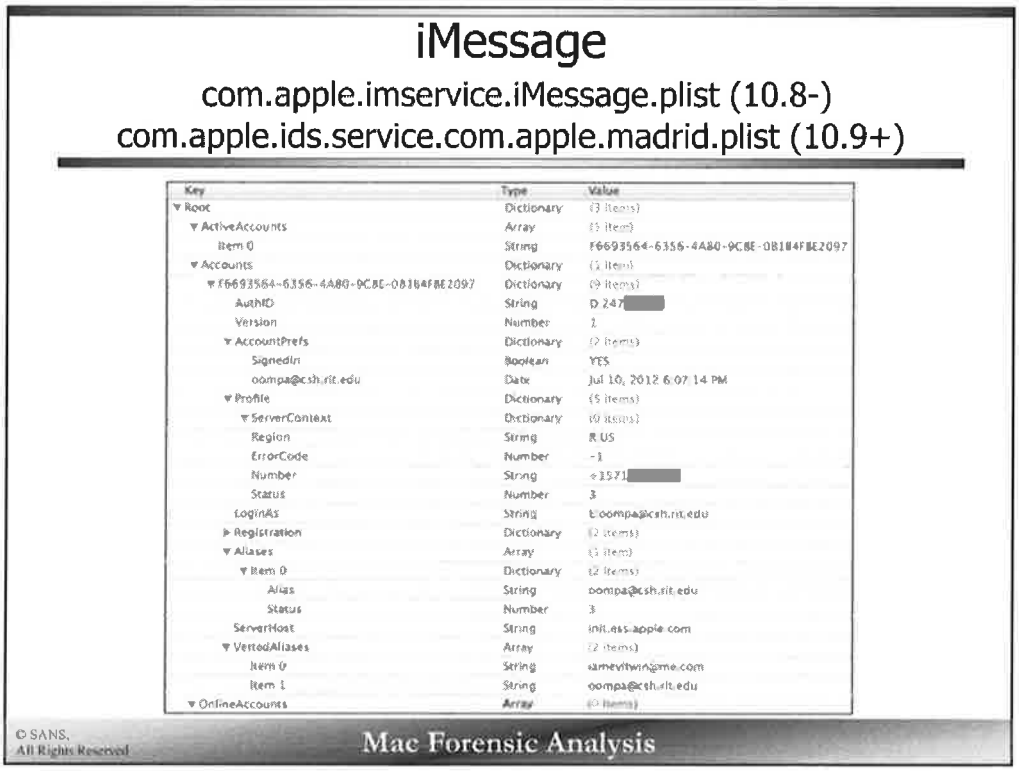


Each instant messaging program or protocol will have its own preferences property list, each sharing the same basic format.

- `com.apple.iChat.AIM.plist`
- `com.apple.iChat.Jabber.plist`
- `com.apple.imservice.iMessage.plist`
- `com.apple.imservice.FaceTime.plist`

The screenshot shows an example of a Jabber/Google Talk account. Contents will include protocol specific connection information and account details which may also include a buddy list.

Key	Type	Value
▼Root	Dictionary	(3 items)
▼ActiveAccounts	Array	(1 item)
Item 0	String	E2304F82-3F22-4025-B320-C9E199DC0FF5
▼Accounts	Dictionary	(1 item)
▼E2304F82-3F22-4025-B320-C9E199DC0FF5	Dictionary	(21 items)
UseKerberos5	Boolean	NO
UseKeychain	Boolean	YES
XMPPTLSEnabled	Boolean	NO
ServerSSLPort	Number	5223
ServerSSLHost	String	talk.google.com
XMPPSupportsInvisible	Boolean	NO
LoadPreviousChatMessages	Boolean	YES
▼AccountPrefs	Dictionary	(0 items)
AllowSelfSignedSSL	Boolean	YES
UseSSL	Boolean	YES
UseMachineName	Boolean	YES
▼Profile	Dictionary	(0 items)
AutoDiscoverHostAndPort	Boolean	NO
Description	String	Gmail
ServerPort	Number	5223
ServerHost	String	talk.google.com
NumberOfLastChatMessagesToShow	Number	25
AutoLogin	Boolean	YES
ResourceName	String	iChat
Priority	Number	0
LoginAs	String	elwoodblues531060@gmail.com
▼OnlineAccounts	Array	(0 items)

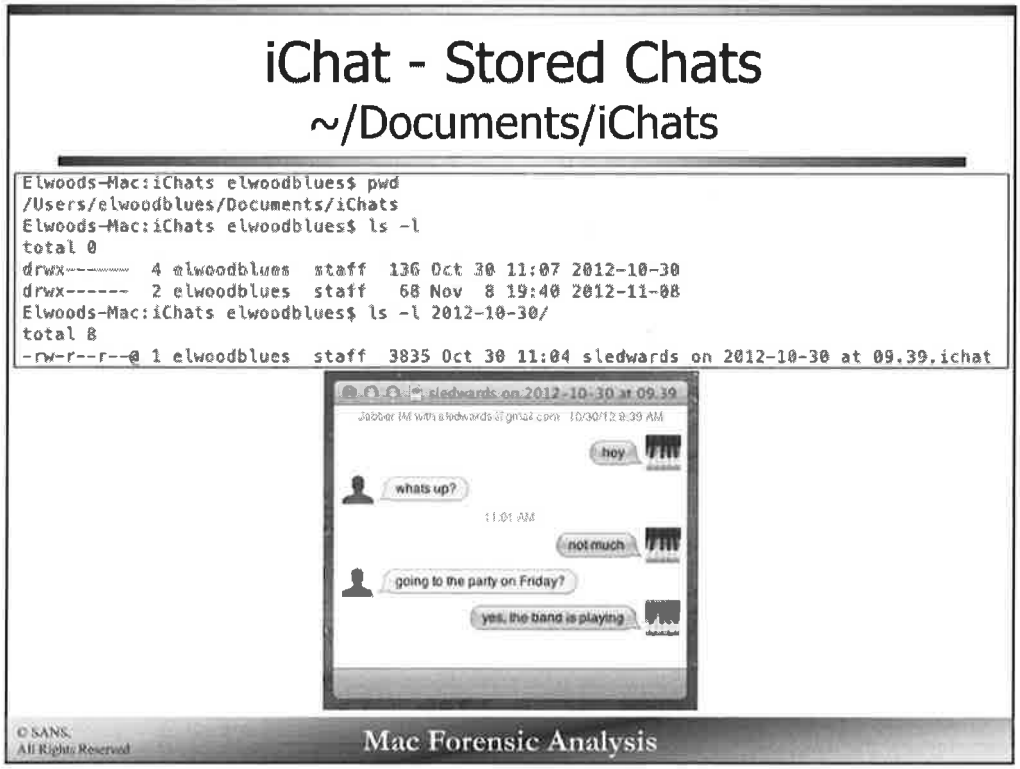


iPhone users can now use their iOS Messages application to message recipients on their iDevices and OS X computers running Messages (10.8, or 10.7 if they are running the Messages Beta software). Messages is the application, iMessage is the protocol. Messages are synced between the associated phone number and iMessage accounts (Apple ID).

iMessage can be used with any account that is registered to it, multiple e-mail accounts, phone numbers, etc. The `com.apple.imservice.iMessage.plist` (10.8-) or the `com.apple.ids.service.com.apple.madrid.plist` (10.9) in the `~/Library/Preferences` directory contains all the accounts that can be used with the iMessage protocol.

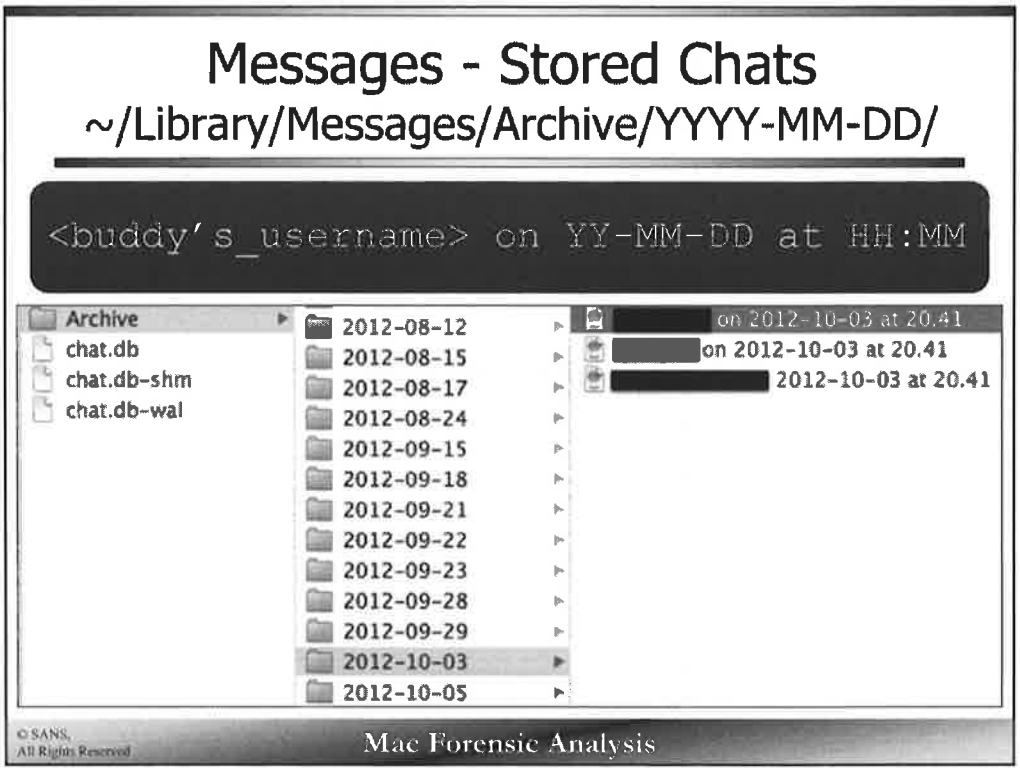
On 10.9+ systems the file `com.apple.imservice.ids.iMessage.<GUID>.plist` located in the `~/Library/Preferences/ByHost/` directory contains more iMessage information including an avatar picture.

Key	Type	Value
▼ Root	Dictionary	(3 items)
▼ ActiveAccounts	Array	(1 item)
Item 0	String	F6693564-6356-4A80-9C8E-081B4FBE2097
▼ Accounts	Dictionary	(1 item)
▼ F6693564-6356-4A80-9C8E-081B4FBE2097	Dictionary	(9 items)
AuthID	String	D:247[redacted]
Version	Number	1
▼ AccountPrefs	Dictionary	(2 items)
SignedIn	Boolean	YES
oompa@csh.rit.edu	Date	Jul 10, 2012 6:07:14 PM
▼ Profile	Dictionary	(5 items)
▼ ServerContext	Dictionary	(0 items)
Region	String	R:US
ErrorCode	Number	-1
Number	String	+1571[redacted]
Status	Number	3
LoginAs	String	E:oompa@csh.rit.edu
▶ Registration	Dictionary	(2 items)
▼ Aliases	Array	(1 item)
▼ Item 0	Dictionary	(2 items)
Alias	String	oompa@csh.rit.edu
Status	Number	3
ServerHost	String	init.ess.apple.com
▼ VettedAliases	Array	(2 items)
Item 0	String	iamevltwin@me.com
Item 1	String	oompa@csh.rit.edu
▼ OnlineAccounts	Array	(0 items)

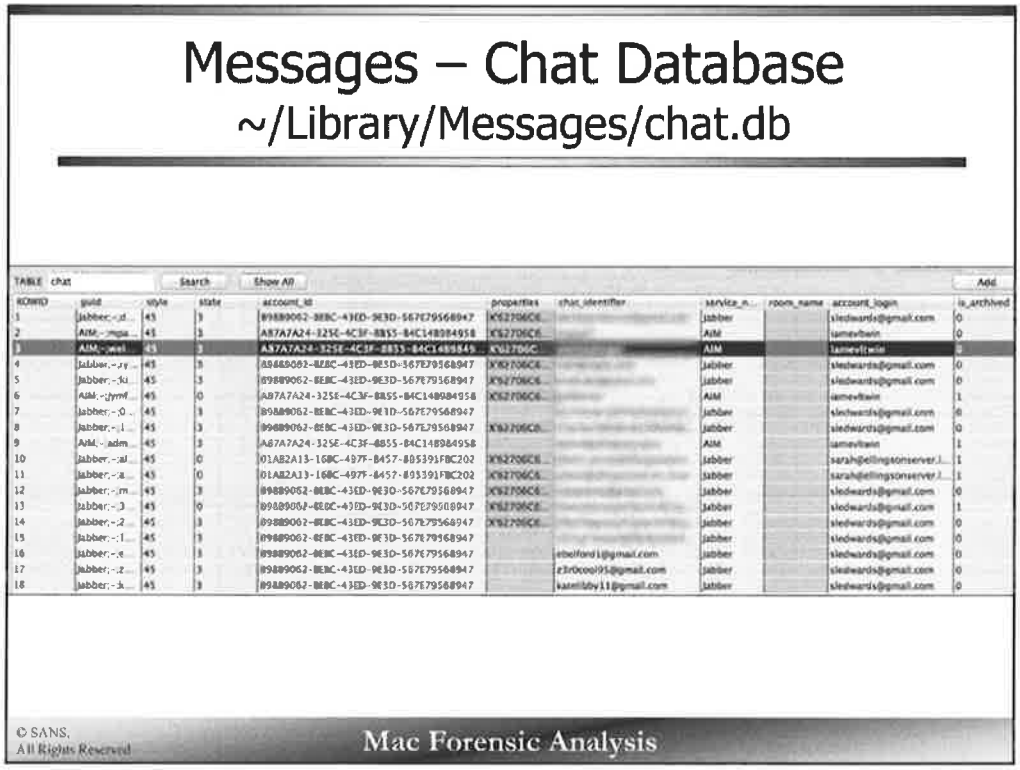


iChat conversations can be saved by checking a box in the Preferences for iChat. These conversation files will be saved in the `~/Documents/iChats` directory. This directory is organized by date, which is then organized by file names that include a date and time (local system time).

Each `*.ichat` file is a binary property list file. This file can be double clicked and viewed with the native program, complete with chat bubbles or any iChat formatting you wish. More detailed information may be found by viewing it in the Xcode property list viewer. This view has more information but it is not as easily read.



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.

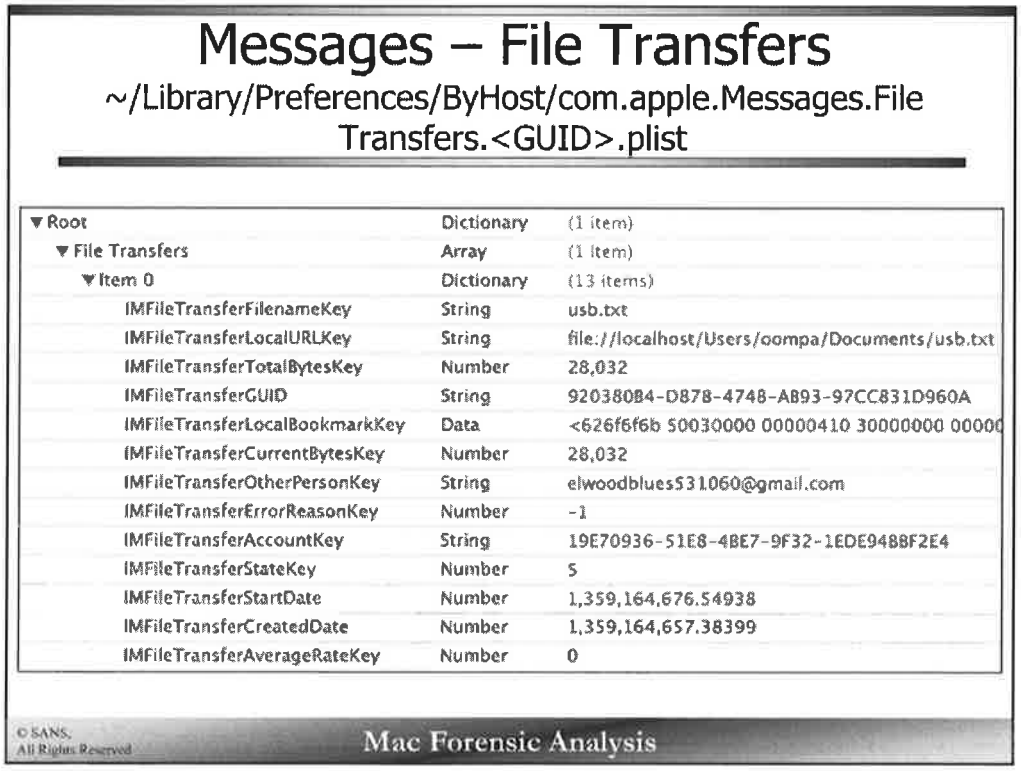


Messages also creates a stored chats SQLite database called `chat.db` located in `~/Library/Messages/`. This database contains information about the chats and messages. Some of the more interesting tables of this database are:

- 'chat'
    - Chat information such as contact and chat protocol
- 'handle'
    - Recent chat contacts
- 'message'
    - iMessage chat contents, including dates, contacts, text and message attributes

TABLE chat | Search | Show All | Add

ROWID	guid	style	state	account_id	properties	chat_identifier	service_n...	room_name	account_login	is_archived
1	jabber;-d...	45	3	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	0
2	AIM;-;mpa...	45	3	A87A7A24-325E-4C3F-8855-B4C14B984958	X'62706C6...		AIM		iamevltwin	0
3	AIM;-;wel...	45	3	A87A7A24-325E-4C3F-8855-B4C1489849...	X'62706C...		AIM		iamevltwin	0
4	jabber;-;y...	45	3	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	0
5	jabber;-;ki...	45	3	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	0
6	AIM;-;jymf...	45	0	A87A7A24-325E-4C3F-8855-B4C14B984958	X'62706C6...		AIM		iamevltwin	1
7	jabber;-;0...	45	3	89889062-8E8C-43ED-9E3D-567E79568947			jabber		sledwards@gmail.com	0
8	jabber;-;1...	45	3	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	0
9	AIM;-;adm...	45	3	A87A7A24-325E-4C3F-8855-B4C14B984958			AIM		iamevltwin	1
10	jabber;-;al...	45	0	01AB2A13-168C-497F-8457-895391FBC202	X'62706C6...		jabber		sarah@ellingsonserver.l...	1
11	jabber;-;a...	45	0	01AB2A13-168C-497F-8457-895391FBC202	X'62706C6...		jabber		sarah@ellingsonserver.l...	1
12	jabber;-;m...	45	3	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	0
13	jabber;-;3...	45	0	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	1
14	jabber;-;2...	45	3	89889062-8E8C-43ED-9E3D-567E79568947	X'62706C6...		jabber		sledwards@gmail.com	0
15	jabber;-;1...	45	3	89889062-8E8C-43ED-9E3D-567E79568947			jabber		sledwards@gmail.com	0
16	jabber;-;e...	45	3	89889062-8E8C-43ED-9E3D-567E79568947		ebelford1@gmail.com	jabber		sledwards@gmail.com	0
17	jabber;-;z...	45	3	89889062-8E8C-43ED-9E3D-567E79568947		z3r0cool95@gmail.com	jabber		sledwards@gmail.com	0
18	jabber;-;k...	45	3	89889062-8E8C-43ED-9E3D-567E79568947		katelibby11@gmail.com	jabber		sledwards@gmail.com	0



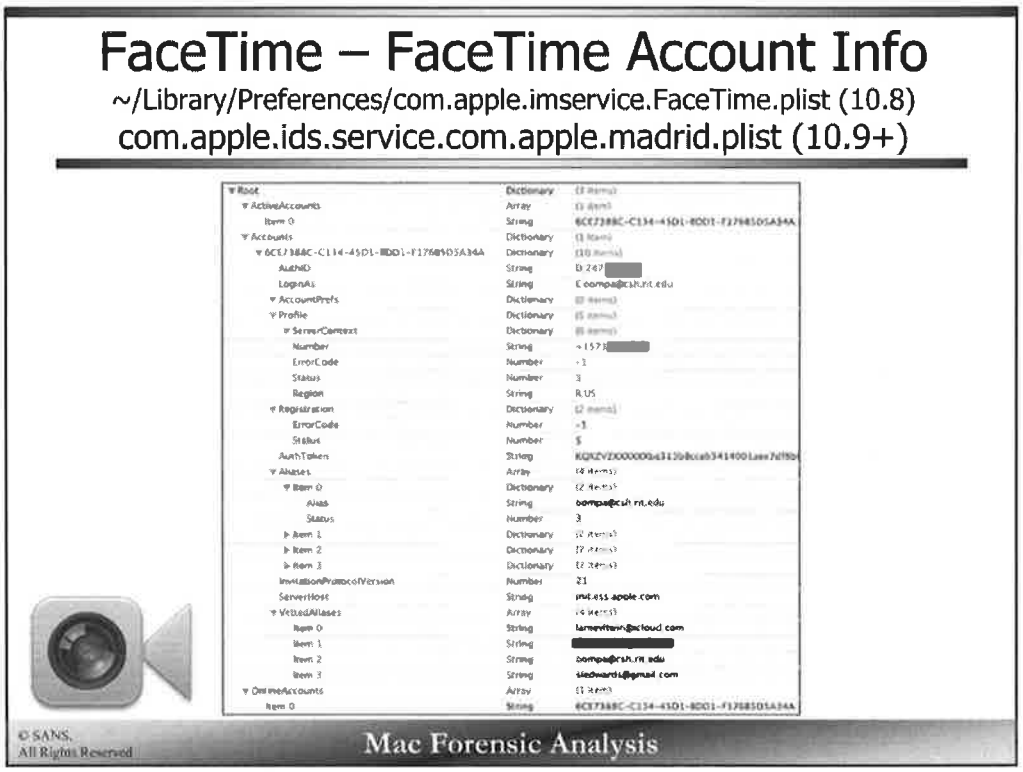
When the Messages file transfer functionality is used, it saves the data in the `com.apple.Messages.FileTransfers.<GUID>.plist` property list file in the `~/Library/Preferences/ByHost/` directory.

Each item contains the following metadata about the transferred file:

- File Name
- Local File Path
- File Size
- Transferred Size
- Bookmark Data
- File Recipient
- IM Account GUID
- Transfer Start Date in Unix Epoch Format
- Transfer Created Date in Unix Epoch Format

Note: If the recipient is using OS X, some file transfer data will be shown in an extended attribute associated with the transferred file.

On 10.9+ Some file transfer information can be found in the `~/Library/Containers/com.apple.soagent/Data/Library/Preferences/com.apple.messageshelper.FileTransferController.plist` file.



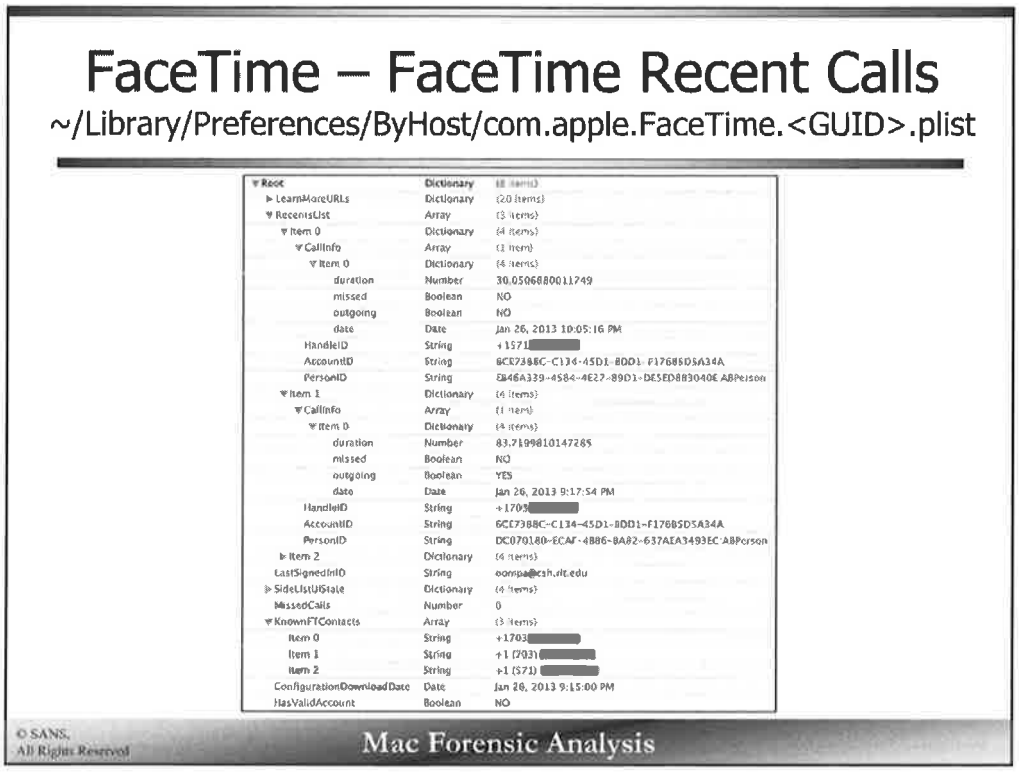
FaceTime is a video chat program that is native to OS X and iOS. The account information for FaceTime can be found in the `com.apple.imservice.FaceTime.plist` (10.8-) or the `com.apple.ids.service.com.apple.madrid.plist` (10.9) file in the `~/Library/Preferences/` directory. This property list shows the phone number and aliased e-mail addresses associated with the FaceTime account.

On 10.9+ systems the file `com.apple.imservice.ids.FaceTime.<GUID>.plist` located in the `~/Library/Preferences/ByHost/` directory contains more iMessage information including an avatar picture.

▼ Root	Dictionary	(3 items)
▼ ActiveAccounts	Array	(1 item)
Item 0	String	6CE7388C-C134-45D1-BDD1-F176B5D5A34A
▼ Accounts	Dictionary	(1 item)
▼ 6CE7388C-C134-45D1-BDD1-F176B5D5A34A	Dictionary	(10 items)
AuthID	String	D:247[redacted]
LoginAs	String	E:oompa@csh.rit.edu
▼ AccountPrefs	Dictionary	(0 items)
▼ Profile	Dictionary	(5 items)
▼ ServerContext	Dictionary	(0 items)
Number	String	+1571[redacted]
ErrorCode	Number	-1
Status	Number	3
Region	String	R:US
▼ Registration	Dictionary	(2 items)
ErrorCode	Number	-1
Status	Number	5
AuthToken	String	KQXZV2XXXXXXba313b8ccab3414001aee7df8b[redacted]
▼ Aliases	Array	(4 items)
▼ Item 0	Dictionary	(2 items)
Alias	String	oompa@csh.rit.edu
Status	Number	3
▶ Item 1	Dictionary	(2 items)
▶ Item 2	Dictionary	(2 items)
▶ Item 3	Dictionary	(2 items)
InvitationProtocolVersion	Number	21
ServerHost	String	init.ess.apple.com
▼ VettedAliases	Array	(4 items)
Item 0	String	iamevitwin@icloud.com
Item 1	String	[redacted]
Item 2	String	oompa@csh.rit.edu
Item 3	String	sledwards@gmail.com
▼ OnlineAccounts	Array	(1 item)
Item 0	String	6CE7388C-C134-45D1-BDD1-F176B5D5A34A

The `com.apple.Facetime.plist` contains the favorite contacts for FaceTime. These favorites are user selected, and each contain a phone number or e-mail address and the associated GUID of the account.

On 10.9+ systems, this information can be found in `~/Library/Containers/com.apple.soagent/Data/Library/Preferences/com.apple.messageshelper.FavoritesController.plist`.

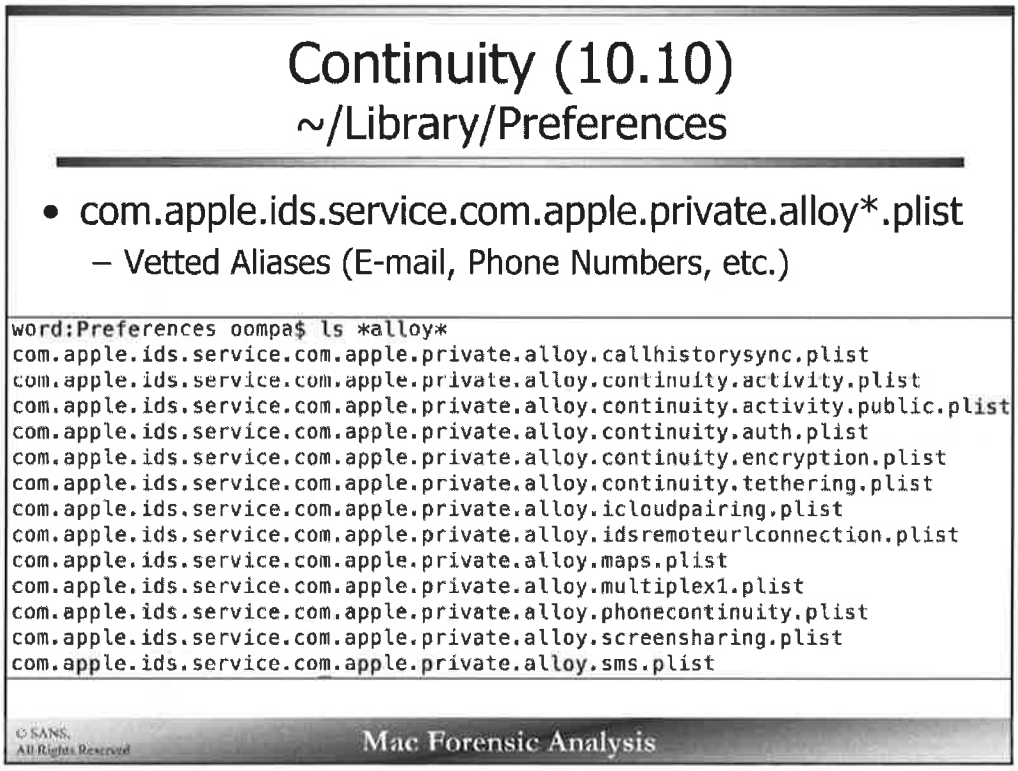


FaceTime records the recent calls in the `com.apple.FaceTime.<GUID>.plist` found in the `~/Library/Preferences/ByHost/` directory. The information found in this property list makes it a good argument not to take the plist files in the `ByHost` directory for granted.

Each recent call contains:

- Call Duration
- Missed Call
- Incoming or Outgoing
- Call Date
- FaceTime Phone Number or E-mail
- Account GUID
- Recipient Contact GUID in Address Book

▼ Root	Dictionary	(8 items)
▶ LearnMoreURLs	Dictionary	(20 items)
▼ RecentsList	Array	(3 items)
▼ Item 0	Dictionary	(4 items)
▼ CallInfo	Array	(1 item)
▼ Item 0	Dictionary	(4 items)
duration	Number	30.0506680011749
missed	Boolean	NO
outgoing	Boolean	NO
date	Date	Jan 26, 2013 10:05:16 PM
HandleID	String	+1571[redacted]
AccountID	String	6CE7388C-C134-45D1-8DD1-F176B5D5A34A
PersonID	String	EB46A339-4584-4E27-89D1-DE5ED8B3040E:ABPerson
▼ Item 1	Dictionary	(4 items)
▼ CallInfo	Array	(1 item)
▼ Item 0	Dictionary	(4 items)
duration	Number	83.7199810147285
missed	Boolean	NO
outgoing	Boolean	YES
date	Date	Jan 26, 2013 9:17:54 PM
HandleID	String	+1703[redacted]
AccountID	String	6CE7388C-C134-45D1-8DD1-F176B5D5A34A
PersonID	String	DC070180-ECAF-4B86-BA82-637AEA3493EC:ABPerson
▶ Item 2	Dictionary	(4 items)
LastSignedInID	String	oompa@csh.rit.edu
▶ SideListUIState	Dictionary	(4 items)
MissedCalls	Number	0
▼ KnownFTContacts	Array	(3 items)
Item 0	String	+1703[redacted]
Item 1	String	+1 (703) [redacted]
Item 2	String	+1 (571) [redacted]
ConfigurationDownloadDate	Date	Jan 26, 2013 9:15:00 PM
HasValidAccount	Boolean	NO



In OS X 10.10 introduced Continuity, a technology that allows a user to accept or make a call through their laptop, send a text message to someone, continue editing a document, or tether you Internet connection.

On the `~/Library/Preferences/` directory, many property lists contain information related to this technology including vetted aliases. Vetted aliases are e-mails and phone numbers which the user setup to use the continuity service.

The SQLite database `CallHistory.storedata` located in the `~/Library/Application Support/CallHistoryDB/` directory contains the calls made using the Continuity technology.

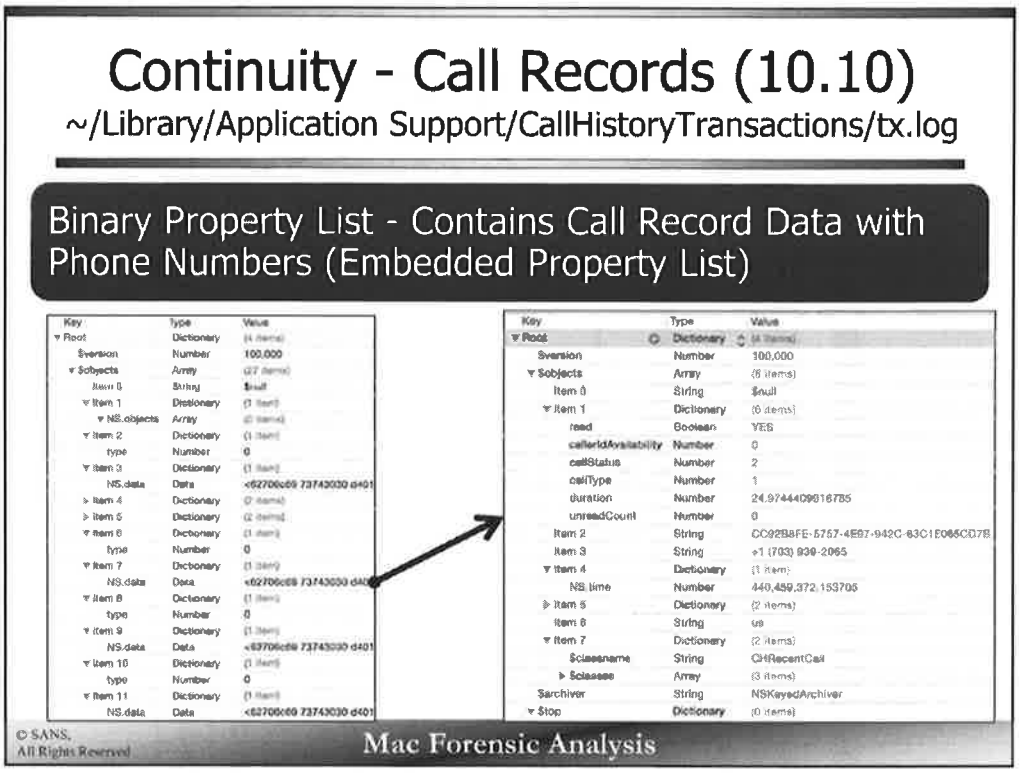
The screenshot in the upper right-hand corner contains the notification that the user will see when taking a call using their OS X system. This call data is then written to the database.

Each record in the database contains the following:

- If it was an incoming or outgoing call (`ZORIGINATED`)
- If the call was answered or missed (`ZANSWERED`)
- The timestamp of the call in Mac epoch (`ZDATE`)
- The duration of the call (`ZDURATION`)
- Country Code (`ZISO_COUNTRY_CODE`)
- Unique GUID of the account used (`ZUNIQUE_ID`)

Table: ZCALLRECORD

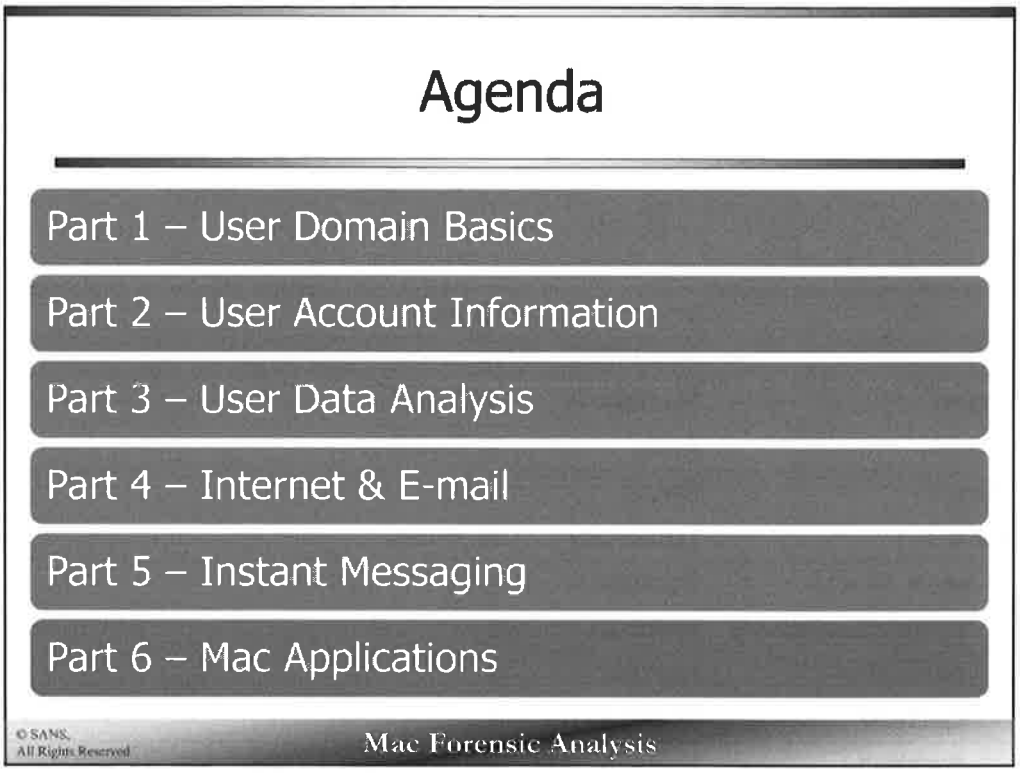
	Z_PK	ZANSWERED	ZCALLTYPE	ZORIGINATED	ZDATE	ZDURATION	ZDEVICE_ID	D_COUNTRY_CC	ZUNIQUE_ID
	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter	Filter
1	1	1	1	0	438363274.9...	0.0		us	9EF87337-9130-462A-...
2	2	0	1	1	440458696.4...	0.0		us	C825F950-7533-4BCF...
3	3	0	1	1	440458733.1...	0.0		us	A3537721-C957-471C-BBEB-E6386E0FD39F
4	4	0	1	1	440458733.1...	0.0		us	A3537721-C957-471C-BBEB-E6386E0FD39F
5	5	1	1	0	440458968.1...	31.86522...		us	D6E5AE96-B87A-4DA6-AA31-25C42F132BF9
6	6	1	1	0	440458968.1...	31.86604...		us	D6E5AE96-B87A-4DA6-AA31-25C42F132BF9
7	7	0	16	1	440459141.4...	7.0		us	0D8704708BCFF8075...
8	8	0	1	1	440459176.8...	18.23335...		us	D768CD22-1000-4ED...
9	9	0	1	1	440459176.8...	18.23631...		us	D768CD22-1000-4ED...



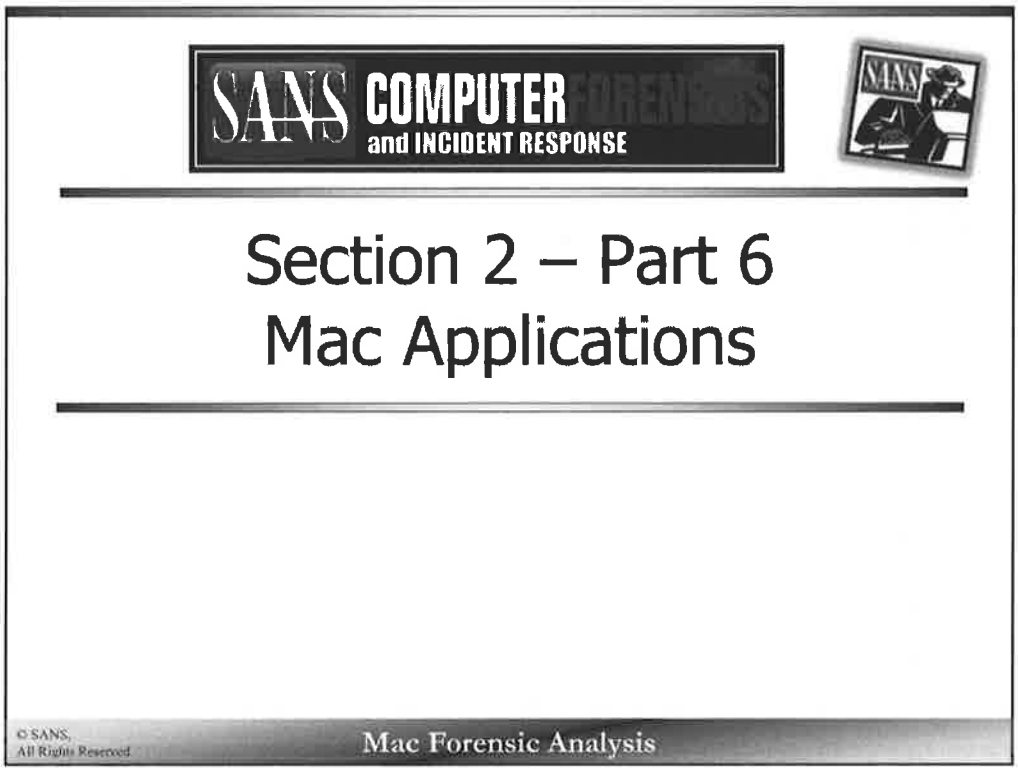
The `tx.log` is a binary property list located in the `~/Library/Application Support/CallHistoryTransactions/` directory. While it uses the log extension, renaming the file to a plist and viewing it in Xcode it makes it slightly easier to read.

This "log" file contains similar information to that found in the `CallHistory.storedata` database stored in embedded property lists. Phone numbers are not stored in the database, however the contact phone numbers are stored in these embedded property lists.

Key	Type	Value
▼ Root	Dictionary	(4 items)
$version	Number	100,000
▼ $objects	Array	(8 items)
Item 0	String	$null
▼ Item 1	Dictionary	(6 items)
read	Boolean	YES
callerIdAvailability	Number	0
callStatus	Number	2
callType	Number	1
duration	Number	24.9744409918785
unreadCount	Number	0
Item 2	String	CC92B8FE-5757-4E97-942C-63C1E065CD7B
Item 3	String	+1 (703) 939-2065
▼ Item 4	Dictionary	(1 item)
NS.time	Number	440,459,372.153705
▶ Item 5	Dictionary	(2 items)
Item 6	String	us
▼ Item 7	Dictionary	(2 items)
$classname	String	CHRecentCall
▶ $classes	Array	(3 items)
$archiver	String	NSKeyedArchiver
▼ $top	Dictionary	(0 items)

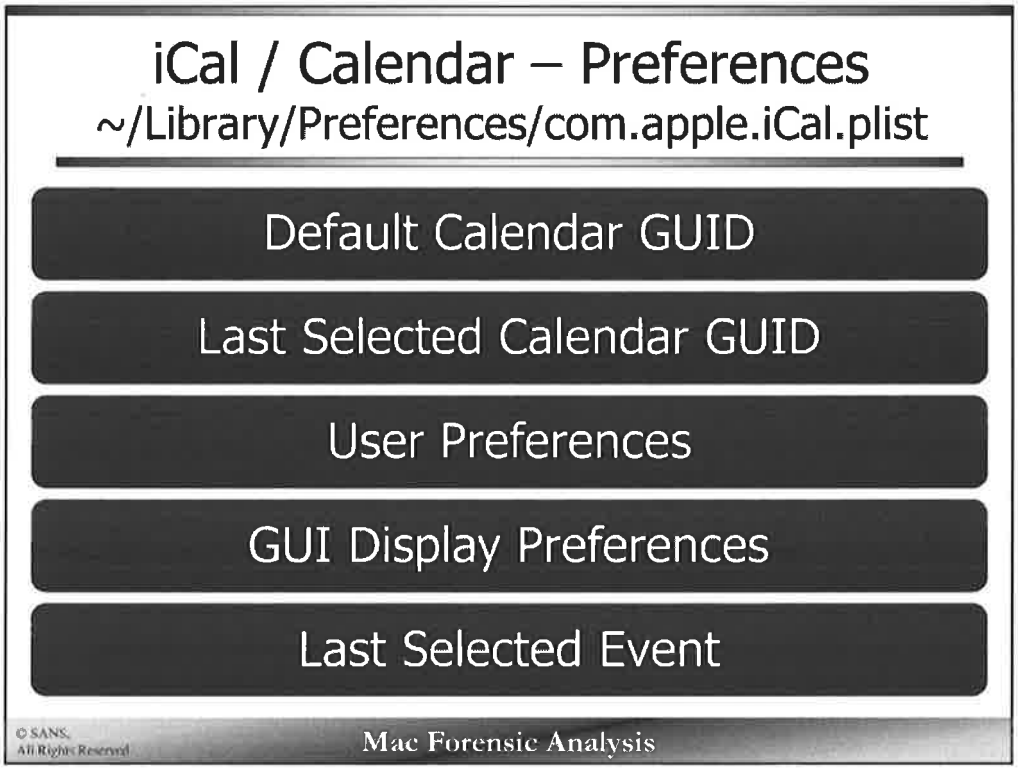


This page intentionally left blank.



This page intentionally left blank.

The native OS X calendar application is iCal or Calendar. The name changed to Calendar in Mountain Lion (10.8). The default calendars provided, shown in the screenshot above, are Home and Work.

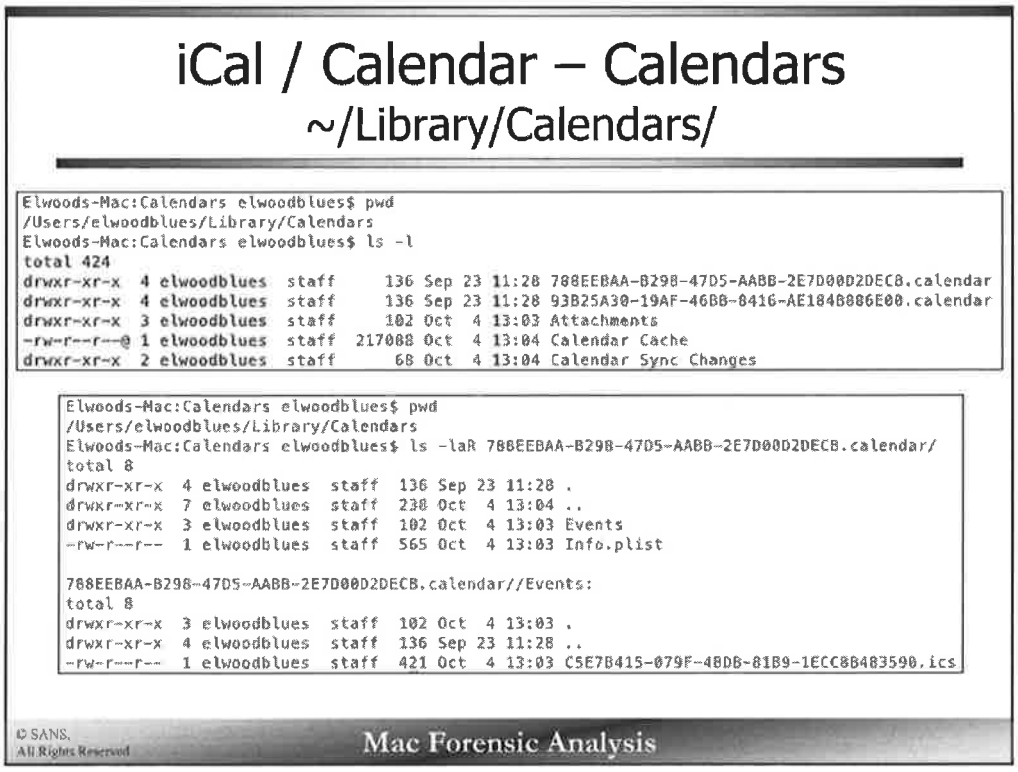


The preferences for iCal or Calendar are both found in the `com.apple.iCal.plist` property list in the user's preferences directory.

The example on the next page shows the property list containing the Default and Last Selected Calendar GUIDs, each calendar has its own GUID.

The property list also shows how the user last had formatted the calendar, in the example the calendar was last viewed by month (versus weekly, daily, yearly). Other preferences include hidden tasks, show birthday calendar, task deletion, day start/end times, minutes in the day, and calendar subscriptions.

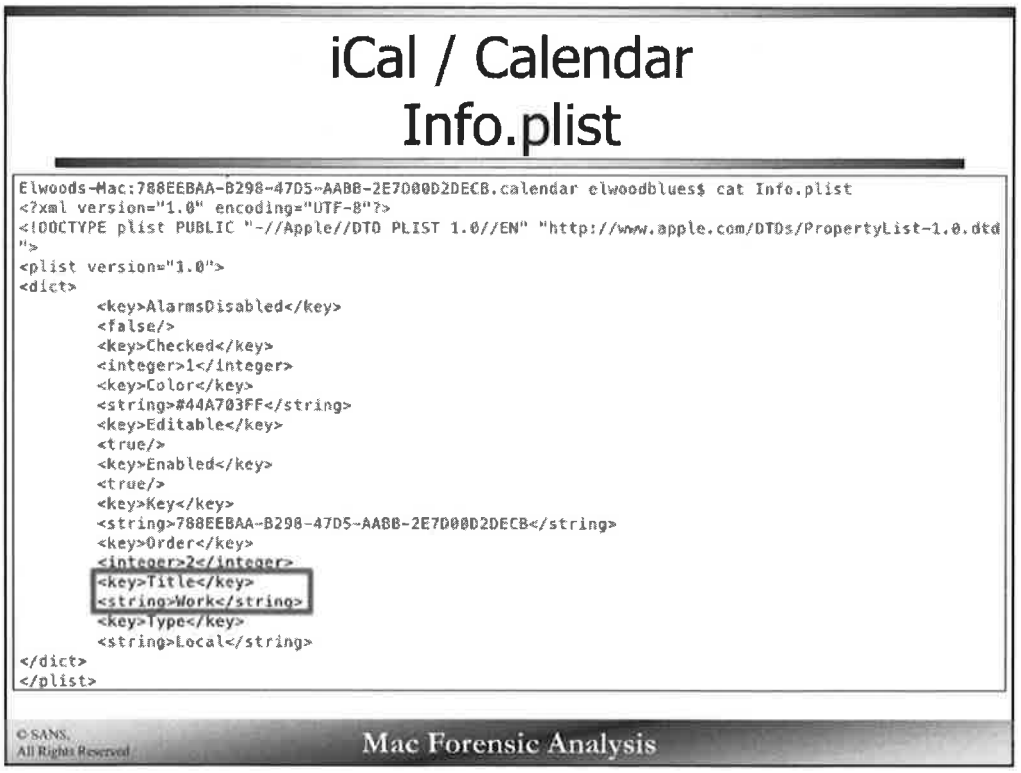
▼ Root	Dictionary	(30 items)
first minute of day time range	Number	0
Tasks should be shown at launch	Boolean	YES
▶ NSSplitView Subview Frames Calendar List	Array	(2 items)
▶ NSSplitView Subview Frames Calendar List	Array	(2 items)
CalDefaultCalendarSelectedByUser	Boolean	NO
▶ NSSplitView Subview Frames Calendar To Do List	Array	(2 items)
▶ NSTableView Sort Ordering SearchView	Array	(4 items)
▶ NSTableView Hidden Columns SearchView	Array	(0 items)
DeleteExpiredTodos	Boolean	NO
last selected calendar list item	String	8797AF44-99E5-4D66-98F4-962A64853C4B
CalDefaultReminderList	String	DF4C8A30-B3F2-4102-B817-6E4FB01E6935
iCal version	Number	100,663,296
last calendar view description	String	Monthly
first shown minute of day	Number	451
▶ DelegatesInSeparateWindows	Dictionary	(1 item)
NSWindow Frame iCal	String	-1389 53 959 1021 -1920 0 1920 1080
DisableEmphasizedViews	Boolean	YES
syncingDisabled	Boolean	NO
NSWindow Frame	String	-1158 315 304 366 -1920 0 1920 1080
last minute of day time range	Number	1,440
CalSuccessfulLaunchTimestampPreferenceKey	Number	3.760837E+08
▶ view rects	Dictionary	(9 items)
CalDefaultPrincipal	String	DDB6E01E-4792-49B6-844B-132BAE56D594
NSDontMakeMainWindowKey	String	NO
CalDefaultCalendar	String	3442B41C-16F2-418E-8C8D-47868613B7B2
▶ NSTableView Columns SearchView	Array	(8 items)
Subscription Calendar Keychains Migrated	Number	2
display birthdays calendar	Boolean	YES
delete todos after	Number	30
Hide completed tasks	Boolean	YES



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`.

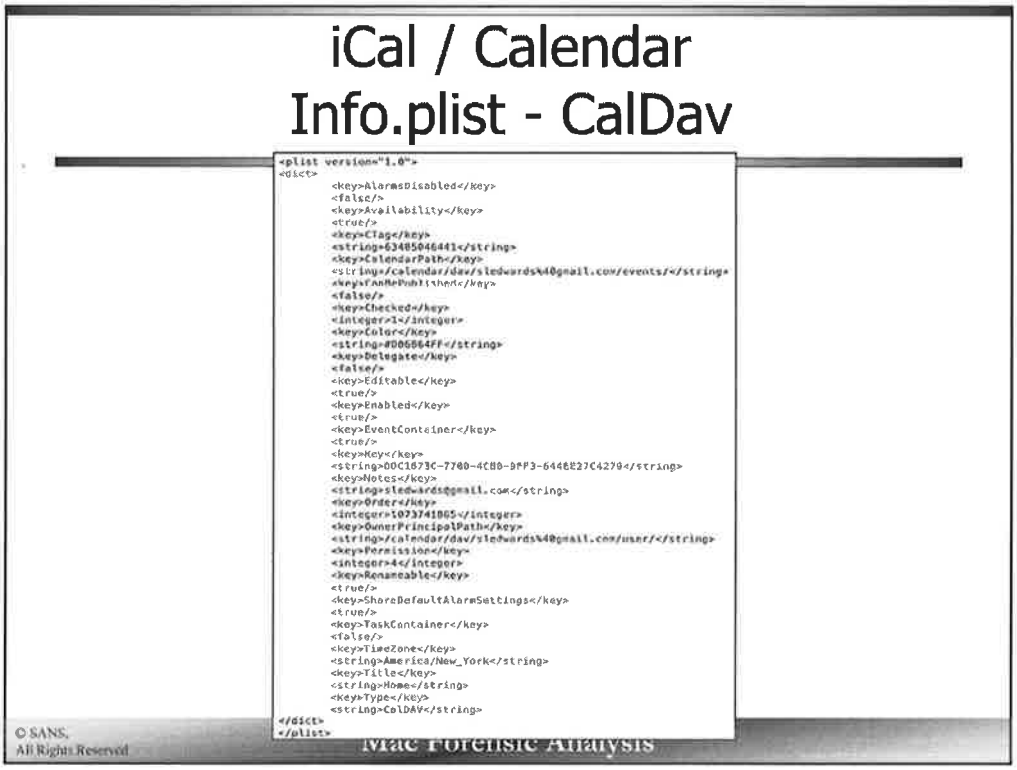
The `Info.plist` contains information about the calendar such as calendar name, GUI preferences, and other configurable items.

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



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.

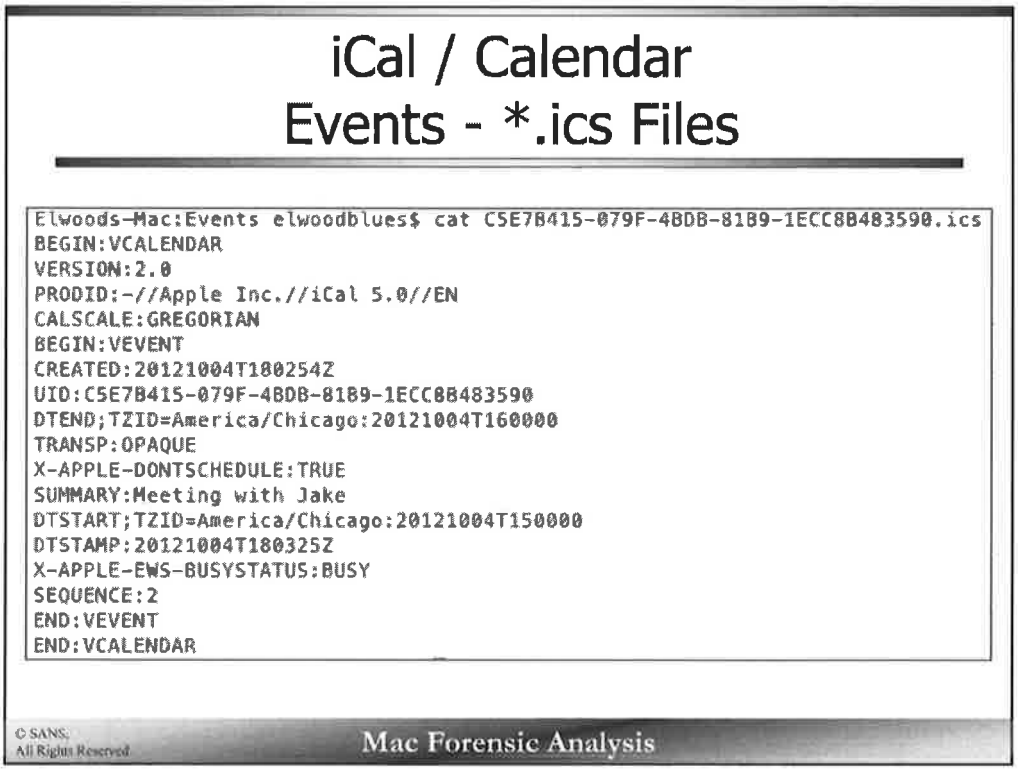


This slide shows an example of a CalDAV calendar `Info.plist`. CalDAV is a standard that allows multiple applications such as iCal or Google Calendar to save the same information and keep it synced between clients.

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

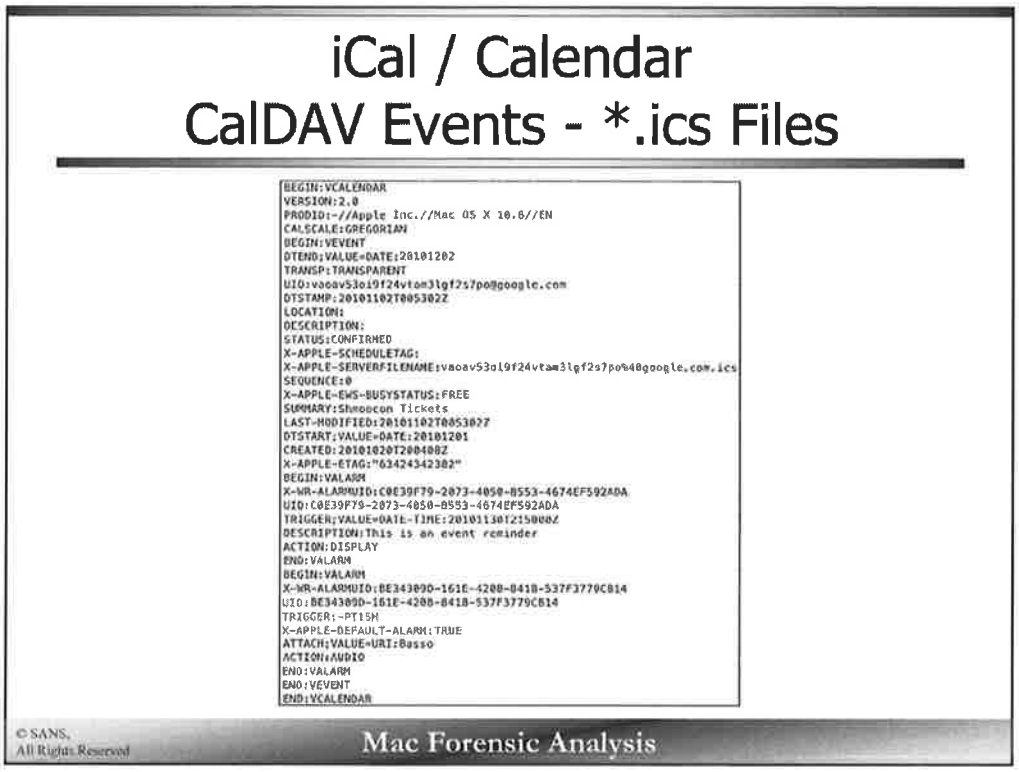
```
<plist version="1.0">
<dict>
        <key>AlarmsDisabled</key>
        <false/>
        <key>Availability</key>
        <true/>
        <key>CTag</key>
        <string>63485046441</string>
        <key>CalendarPath</key>
        <string>/calendar/dav/sledwards%40gmail.com/events/</string>
        <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>
```



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

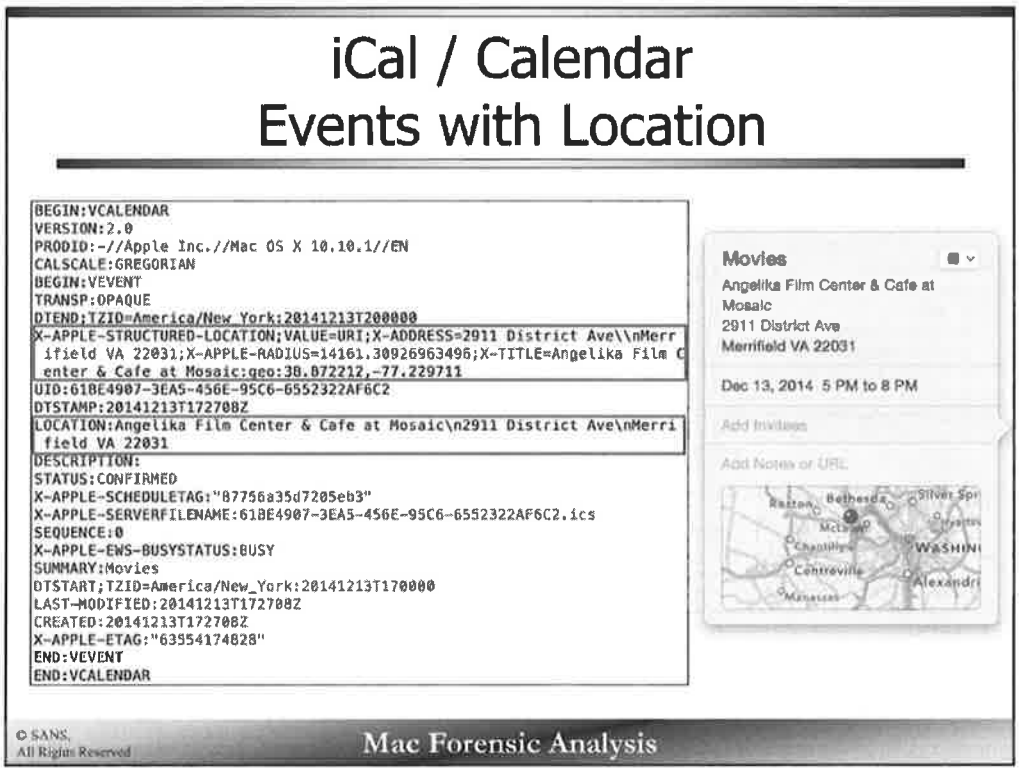
- Create Date
- Event Start Time
- Event End Time
- Unique ID
- Time Zone Information
- Event Summary

```
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
```



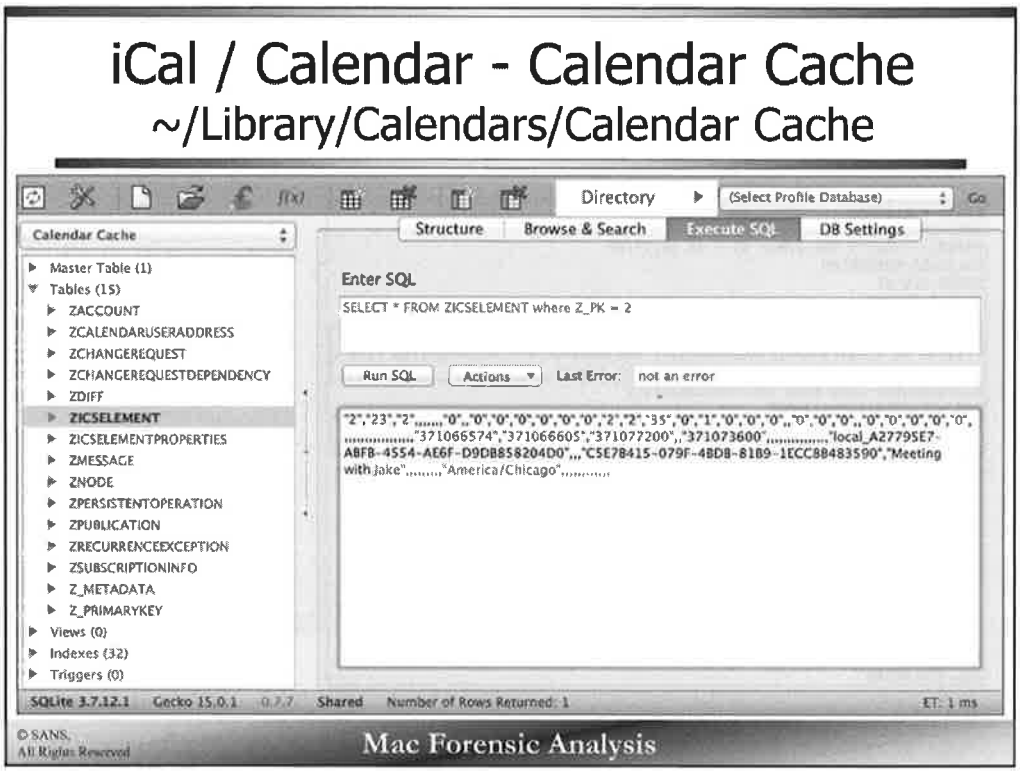
CalDAV calendar events may have additional information depending on the client the event was created with. The example above shows a Google Calendar event.

```
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:Shmoocon 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
```



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.



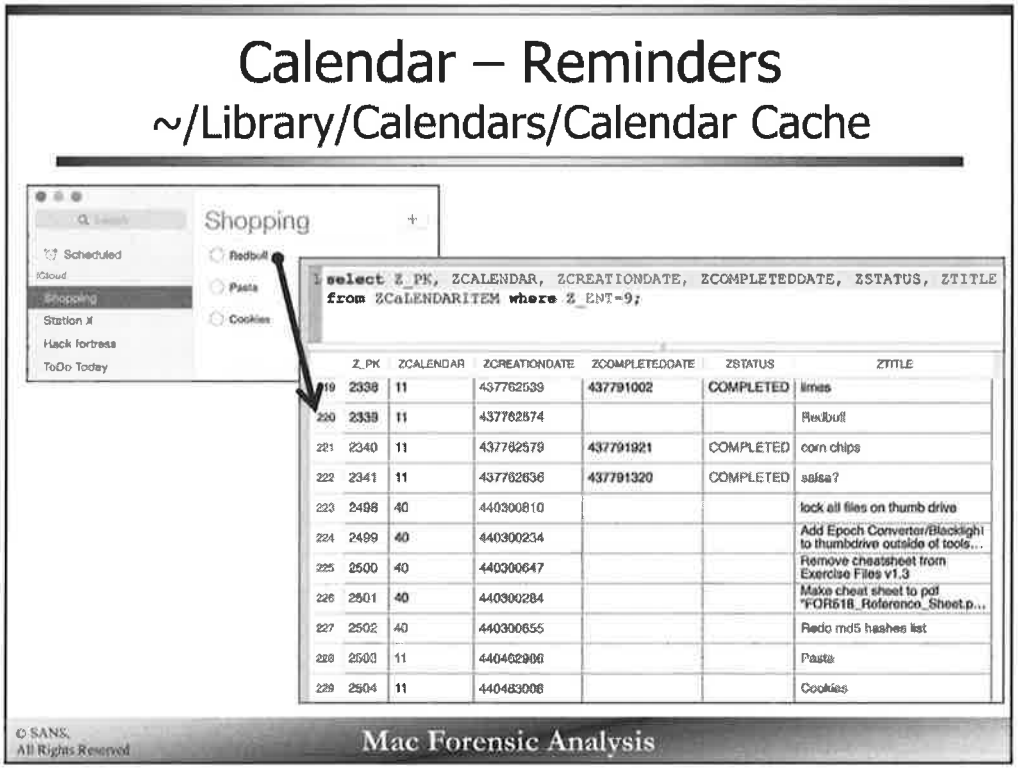
The Calendar Cache SQLite database found in the `~/Library/Calendars` directory contains information for the Calendars.

The `ZICSELEMENT` table contains calendar event information. The screenshot example shows a SQL query for the record with the primary key of 2. Each comma separated element comes from a column of the `ZICSELEMENT` table. This data includes much of the same information found in the `.ics` file.

The `ZACCOUNT` table shows various Calendar Accounts such as iCloud or Google Calendar.

On newer systems, the same information is found in this database with a few updates:

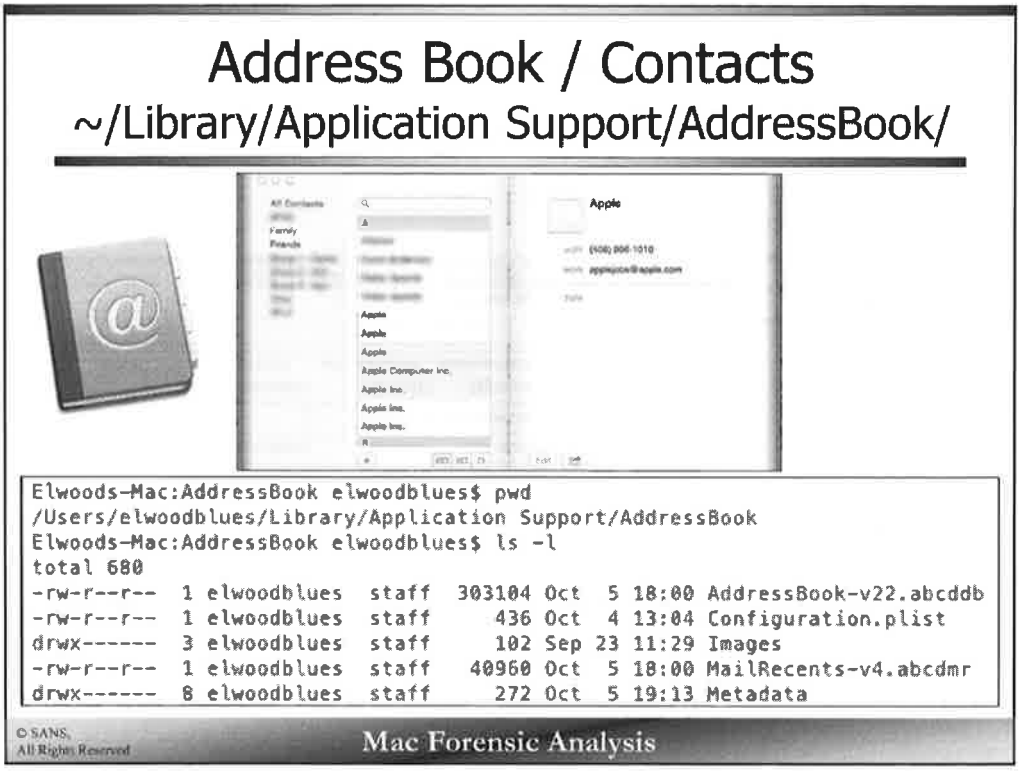
- `ZICSELEMENT` is now `ZCALENDARITEM`
- `ZLOCATION` contains locational information (similar to that found in the ICS files)



The Reminders application also uses the Calendar Cache file. When the `Z_ENT` = 9 in the `ZCALENDARITEM` table, this is a reminder item.

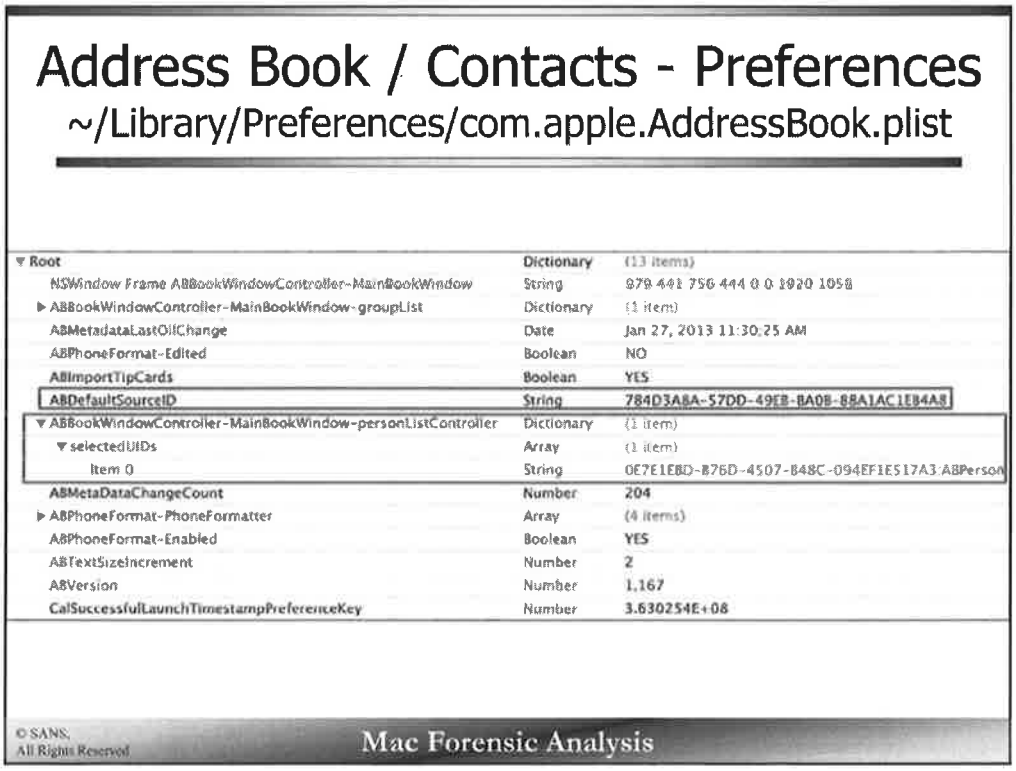
Each reminder may be associated with a specific list, in this example – Shopping. This list names can be correlated with data found in the `ZNODE` table of this database.

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



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

The Address Book or Contacts data is stored in `~/Library/Application Support/AddressBook/` directory.

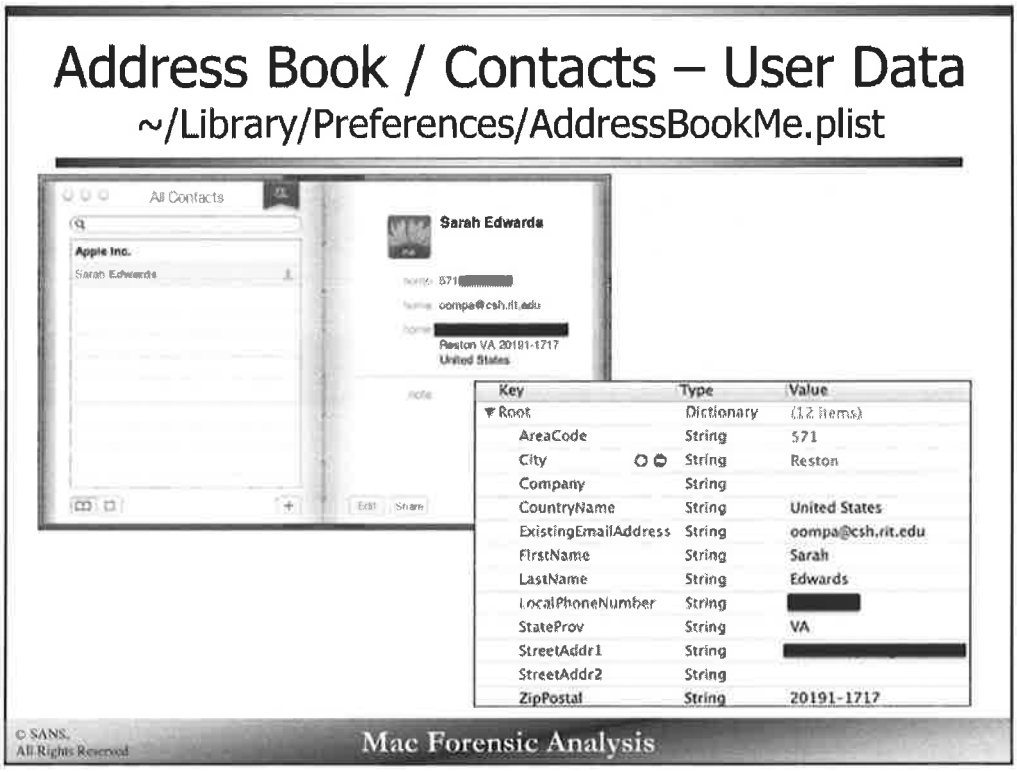


The Address Book/Contacts preferences are stored in the `com.apple.AddressBook.plist` property list.

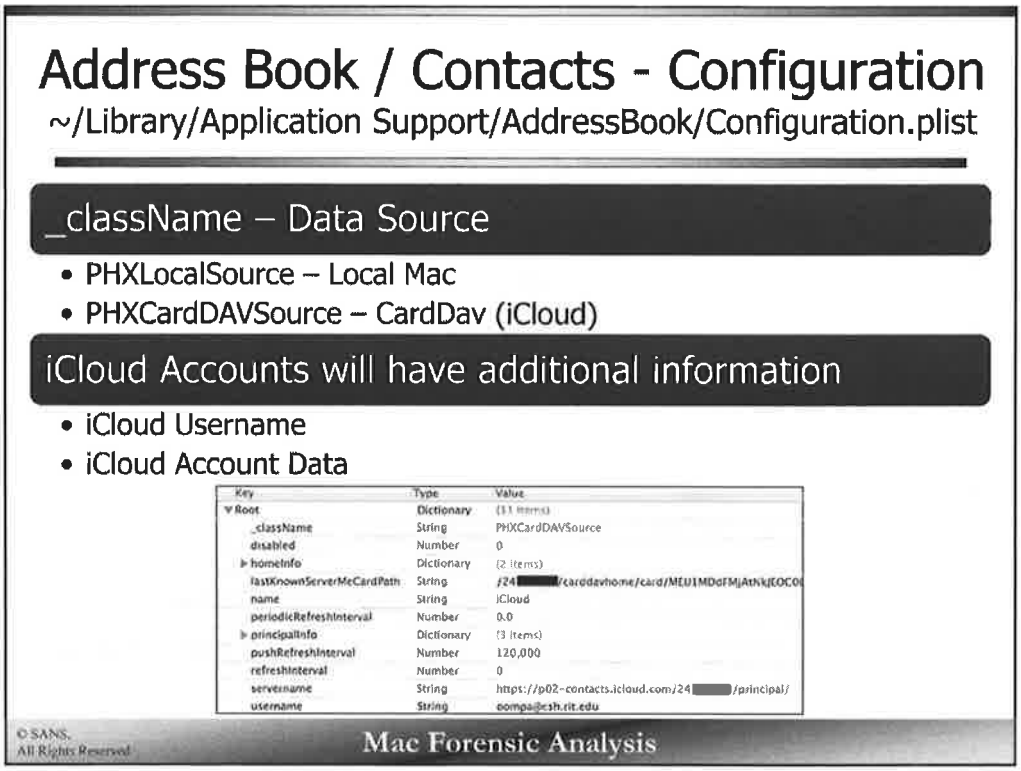
This file contains information such as the last selected contact (`ABBookWindowController-MainBookWindow-personListController/selectedUIDs`) and the GUID for the source, found in the `~/Application Support/AddressBook/Sources/` directory.

▼Root	Dictionary	(13 items)
NSWindow Frame ABBookWindowController-MainBookWindow	String	979 441 756 444 0 0 1920 1058
▶ABBookWindowController-MainBookWindow-groupList	Dictionary	(1 item)
ABMetadataLastOilChange	Date	Jan 27, 2013 11:30:25 AM
ABPhoneFormat-Edited	Boolean	NO
ABImportTipCards	Boolean	YES
ABDefaultSourceID	String	784D3A8A-57DD-49EB-BA0B-88A1AC1EB4A8
▼ABBookWindowController-MainBookWindow-personListController	Dictionary	(1 item)
▼selectedUIDs	Array	(1 item)
Item 0	String	0E7E1EBD-B76D-4507-B48C-094EF1E517A3:ABPerson
ABMetaDataChangeCount	Number	204
▶ABPhoneFormat-PhoneFormatter	Array	(4 items)
ABPhoneFormat-Enabled	Boolean	YES
ABTextSizeIncrement	Number	2
ABVersion	Number	1,167
CalSuccessfulLaunchTimestampPreferenceKey	Number	3.630254E+08

176

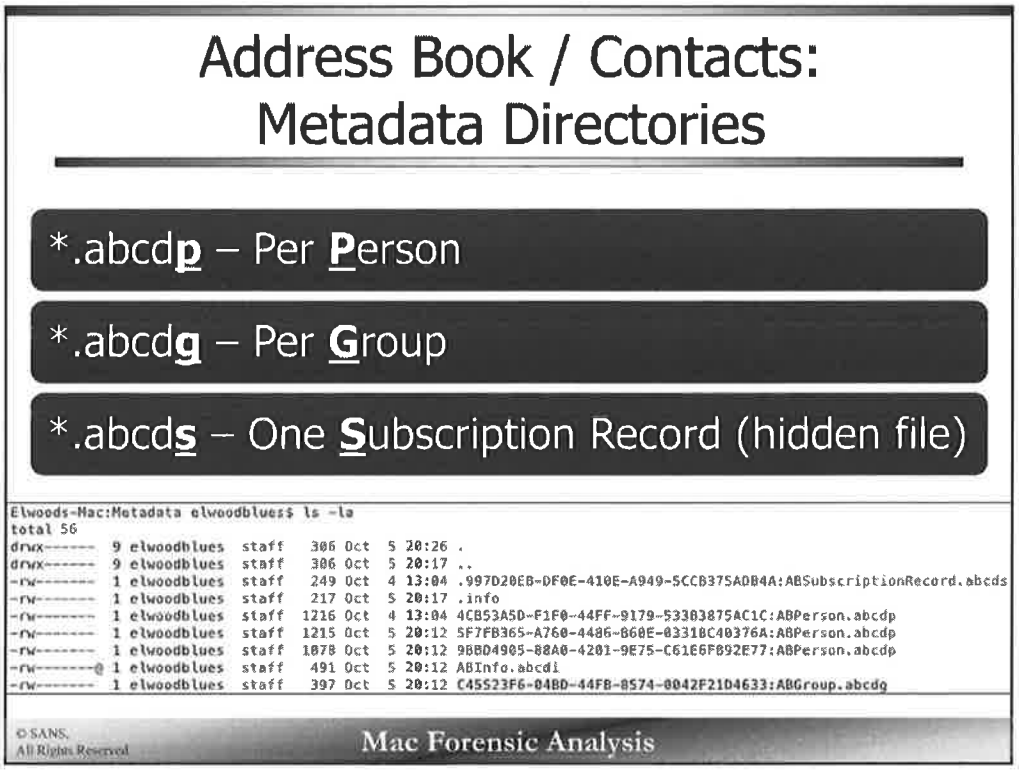


A user may have a default entry for themselves in their Address Book. This information can be found in the `AddressBookMe.plist` file in the `~/Librarary/Preferences/` directory.



The OS X Address Book can be populated from the local Mac or via iCloud. The `Configuration.plist` will show you how Address Book or Contacts were populated.

The configuration data may also be found in another directory called Sources, i.e., (`~/Library/Application Support/AddressBook/Sources/<GUID>/Configuration.plist`)

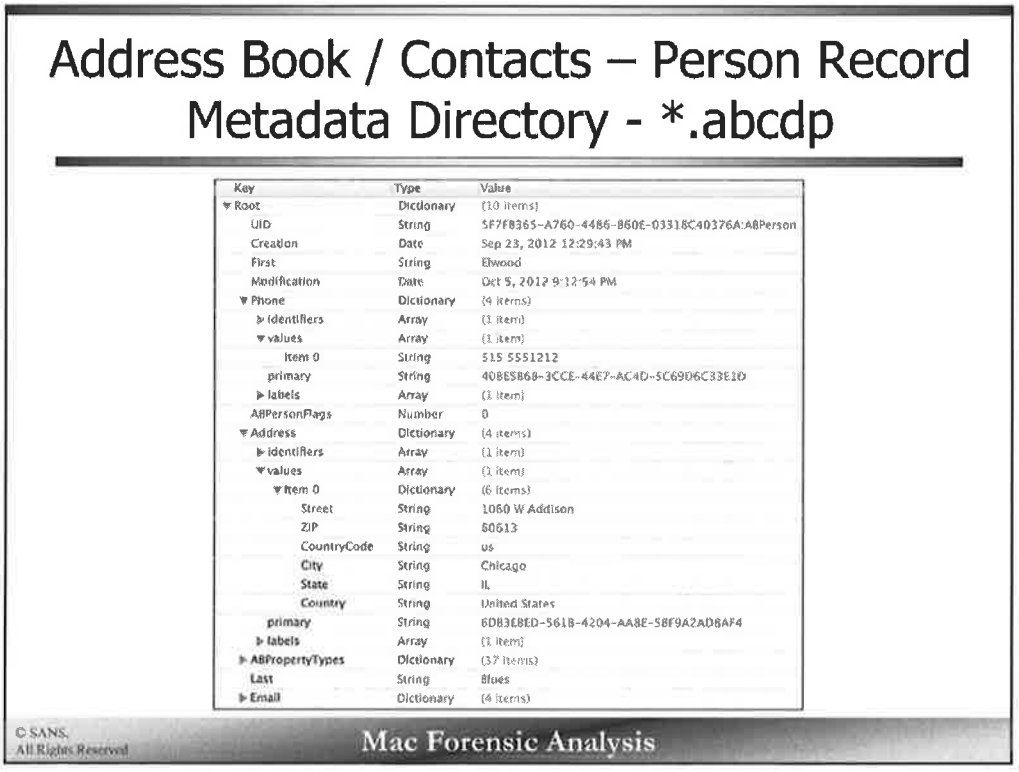


The Metadata directory contains a file for each person, group, or subscription. Each file is named accordingly; 'p' for person, 'g' for group, and 's' for subscription. Each file is a binary property list containing the information for that particular Address Book entry.

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

Newer versions of OS X has a Sources directories that contains the GUID of the specific Address Book source, you will find the Metadata directory here.

```
Elwoods-Mac:Metadata elwoodblues$ ls -la
total 56
drwx------  9 elwoodblues  staff   306 Oct  5 20:26 .
drwx------  9 elwoodblues  staff   306 Oct  5 20:17 ..
-rw-------  1 elwoodblues  staff   249 Oct  4 13:04 .997D20EB-DF0E-410E-A949-5CCB375ADB4A:ABSubscriptionRecord.abcds
-rw-------  1 elwoodblues  staff   217 Oct  5 20:17 .info
-rw-------  1 elwoodblues  staff  1216 Oct  4 13:04 4CB53A5D-F1F0-44FF-9179-533B3875AC1C:ABPerson.abcdp
-rw-------  1 elwoodblues  staff  1215 Oct  5 20:12 5F7FB365-A760-4486-B60E-03318C40376A:ABPerson.abcdp
-rw-------  1 elwoodblues  staff  1078 Oct  5 20:12 9BBD4905-88A0-4201-9E75-C61E6FB92E77:ABPerson.abcdp
-rw-------@ 1 elwoodblues  staff   491 Oct  5 20:12 ABInfo.abcdi
-rw-------  1 elwoodblues  staff   397 Oct  5 20:12 C45523F6-04BD-44FB-8574-0042F21D4633:ABGroup.abcdg
```



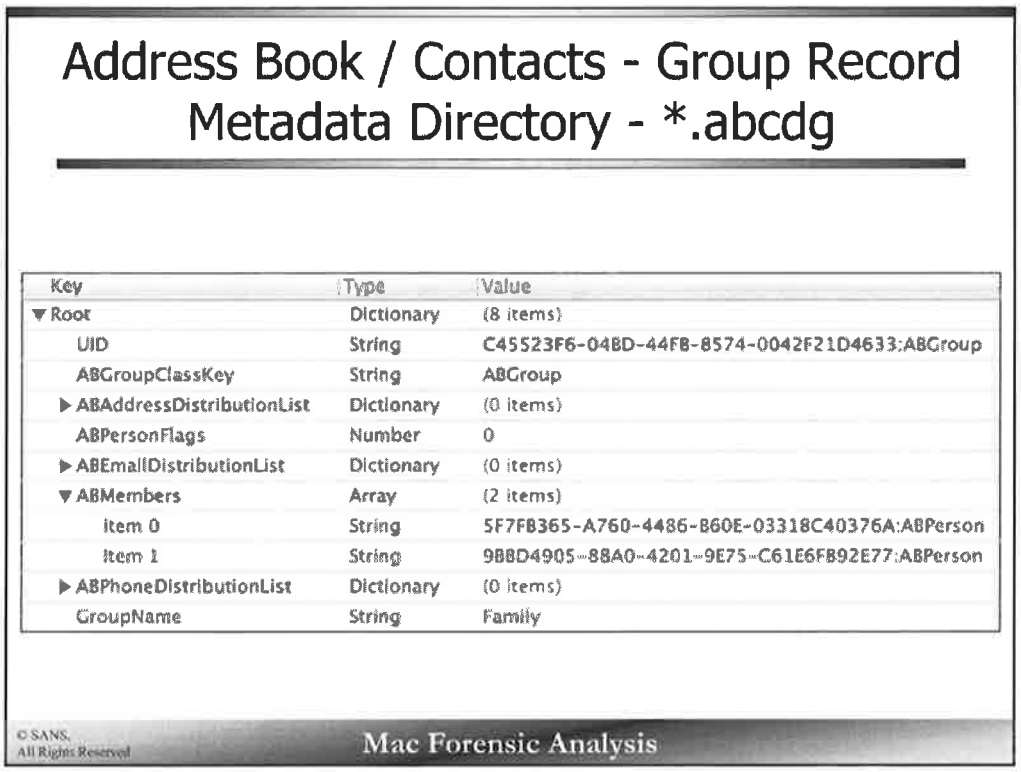
To view the raw Address Book records with Xcode, you will need to rename them as `.plist`, otherwise they will show up in a view similar to what is found in the Address Book/Contacts application.

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, e-mail, 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.

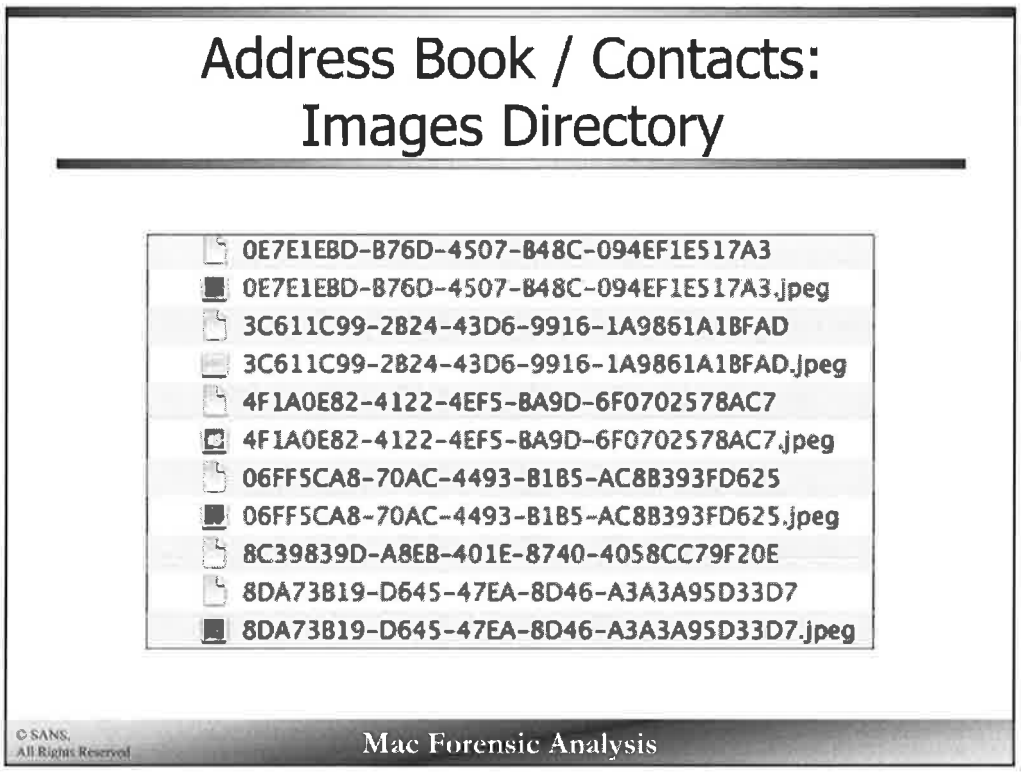
Key	Type	Value
▼ Root	Dictionary	(10 items)
UID	String	5F7FB365-A760-4486-B60E-03318C40376A:ABPerson
Creation	Date	Sep 23, 2012 12:29:43 PM
First	String	Elwood
Modification	Date	Oct 5, 2012 9:12:54 PM
▼ Phone	Dictionary	(4 items)
▶ identifiers	Array	(1 item)
▼ values	Array	(1 item)
Item 0	String	515 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)



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 e-mail, phone, or mailing distribution list.

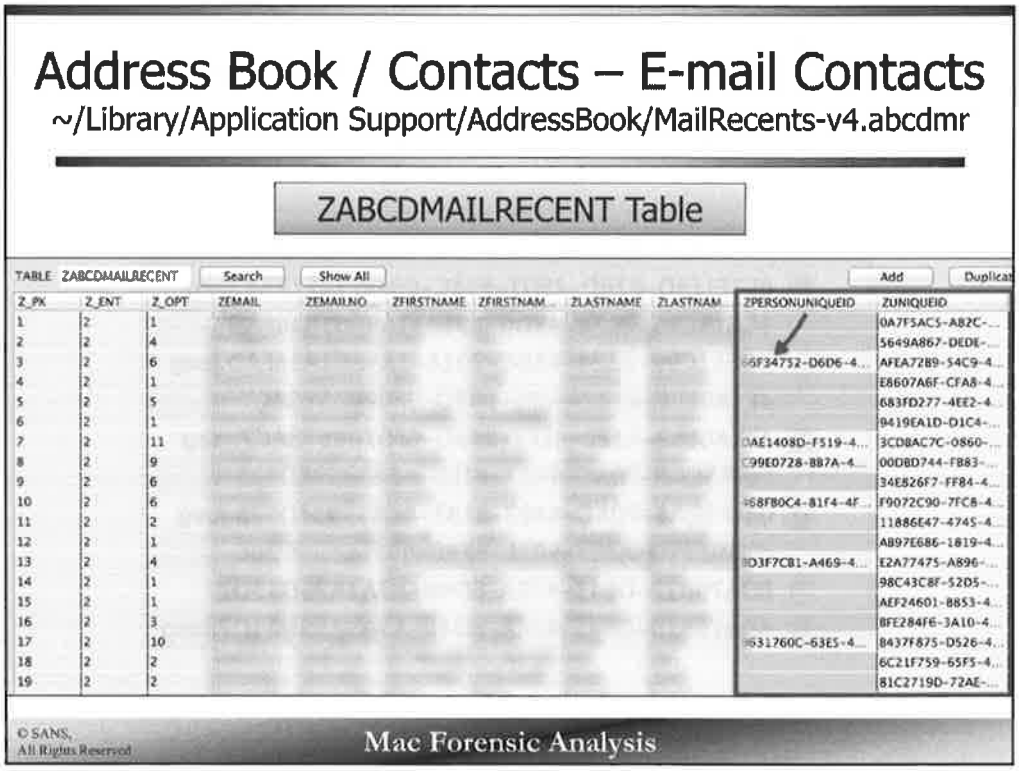
Key	Type	Value
▼ Root	Dictionary	(8 items)
UID	String	C45523F6-04BD-44FB-8574-0042F21D4633:ABGroup
ABGroupClassKey	String	ABGroup
▶ ABAddressDistributionList	Dictionary	(0 items)
ABPersonFlags	Number	0
▶ ABEmailDistributionList	Dictionary	(0 items)
▼ ABMembers	Array	(2 items)
Item 0	String	5F7FB365-A760-4486-B60E-03318C40376A:ABPerson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



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 has a `Sources` directories that contains the GUID of the specific Address Book source, you will find the `Images` directory here.



The `MailRecents-v4.abcdmr` SQLite database in the `~/Library/Application Support/AddressBook` directory contains information about e-mail contacts. This data may contain other contacts not in the user's Address Book or Contacts application.

The `ZABCDMAILRECENT` table, shown above, contains the contact information including e-mail address, first and last names. If there is a GUID in the `ZPERSONUNIQUEID` column, there is a corresponding Address Book entry in the `Metadata` directory. The value in the `Z_PK` column is a unique number, each contact will be referenced by this number.

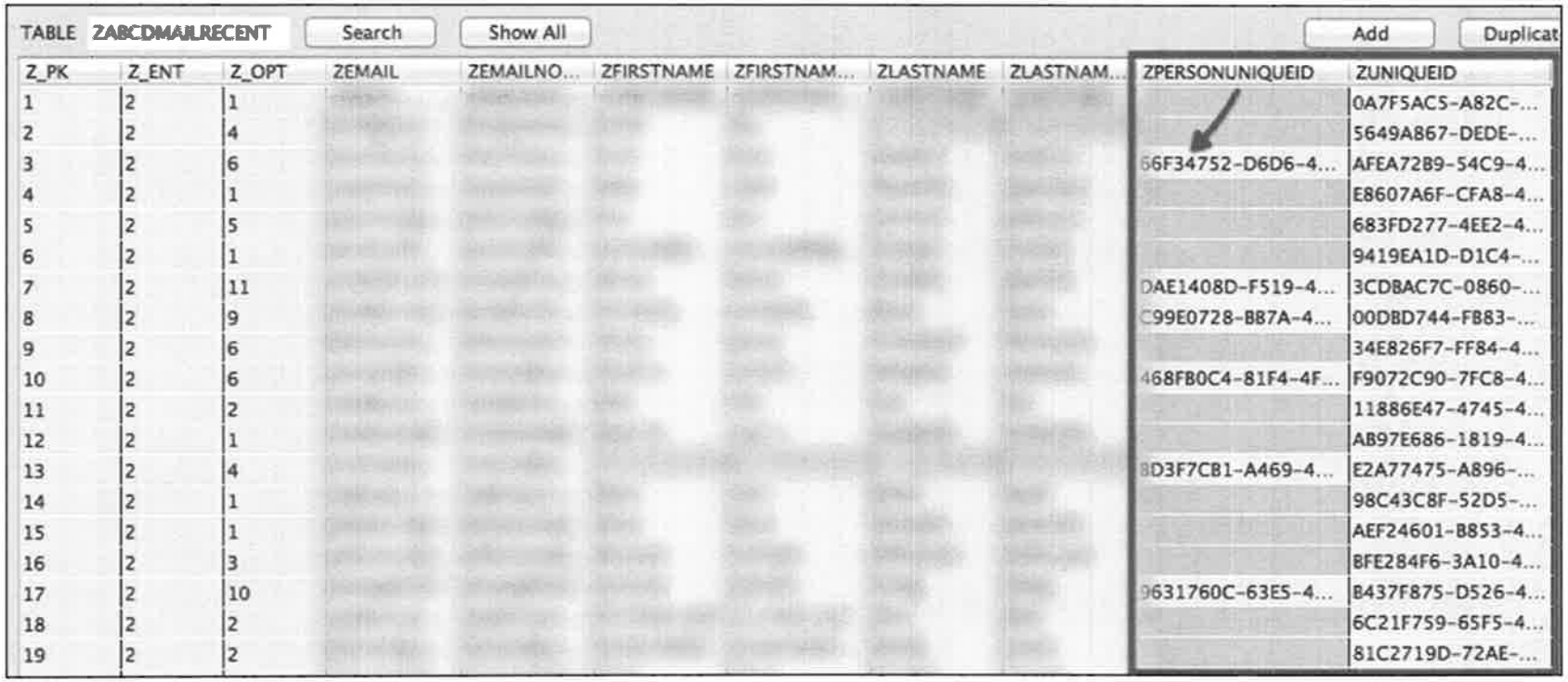
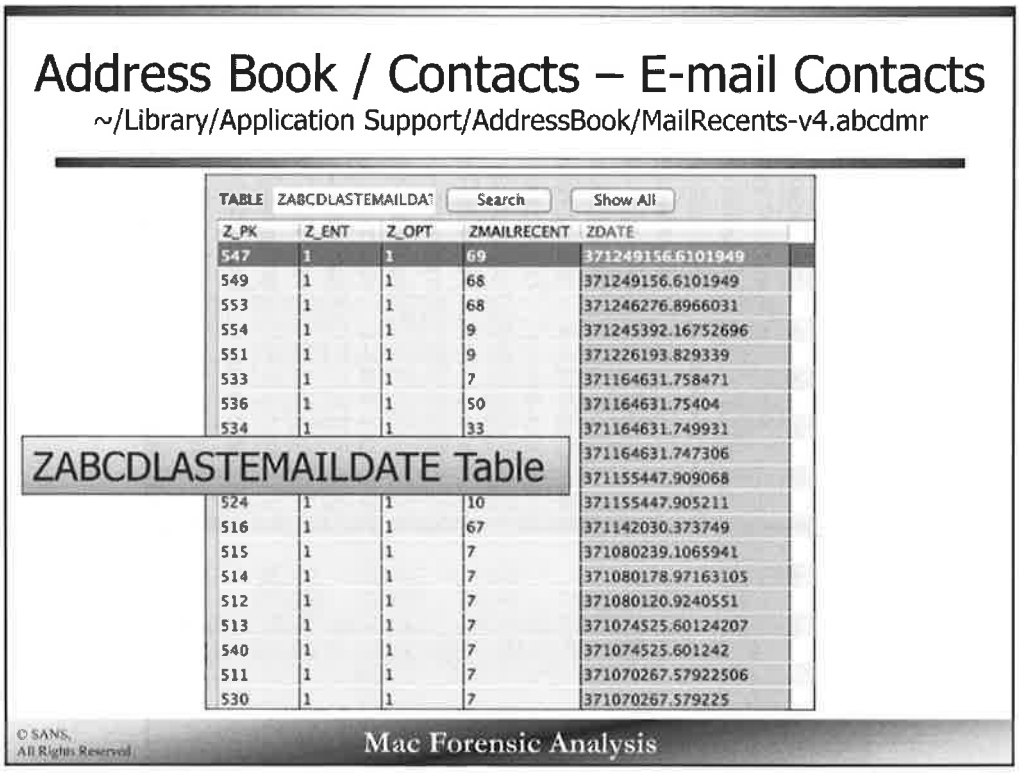


TABLE ZABCDMAILRECENT | Search | Show All | Add | Duplicat

Z_PK	Z_ENT	Z_OPT	ZEMAIL	ZEMAILNO...	ZFIRSTNAME	ZFIRSTNAM...	ZLASTNAME	ZLASTNAM...	ZPERSONUNIQUEID	ZUNIQUEID
1	2	1								0A7F5AC5-A82C-...
2	2	4								5649A867-DEDE-...
3	2	6							66F34752-D6D6-4...	AFEA72B9-54C9-4...
4	2	1								E8607A6F-CFA8-4...
5	2	5								683FD277-4EE2-4...
6	2	1								9419EA1D-D1C4-...
7	2	11							DAE1408D-F519-4...	3CD8AC7C-0860-...
8	2	9							C99E0728-B87A-4...	00DBD744-FB83-...
9	2	6								34E826F7-FF84-4...
10	2	6							468FB0C4-81F4-4F...	F9072C90-7FC8-4...
11	2	2								11886E47-4745-4...
12	2	1								AB97E686-1819-4...
13	2	4							8D3F7CB1-A469-4...	E2A77475-A896-...
14	2	1								98C43C8F-52D5-...
15	2	1								AEF24601-B853-4...
16	2	3								BFE284F6-3A10-4...
17	2	10							9631760C-63E5-4...	B437F875-D526-4...
18	2	2								6C21F759-65F5-4...
19	2	2								81C2719D-72AE-...

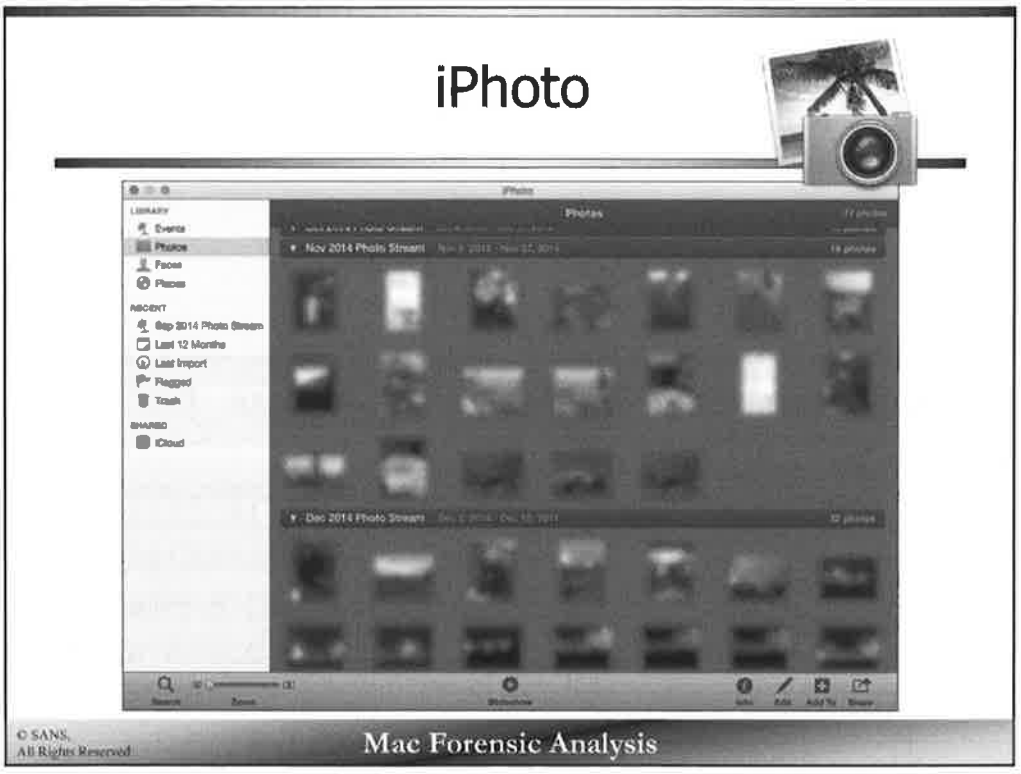


The `ZABCDLASTEMAILDATE` table contains a WebKit/Mac Absolute timestamp which corresponds to an e-mail message with a contact. This can be a way to determine when an e-mail contact was last communicated with.

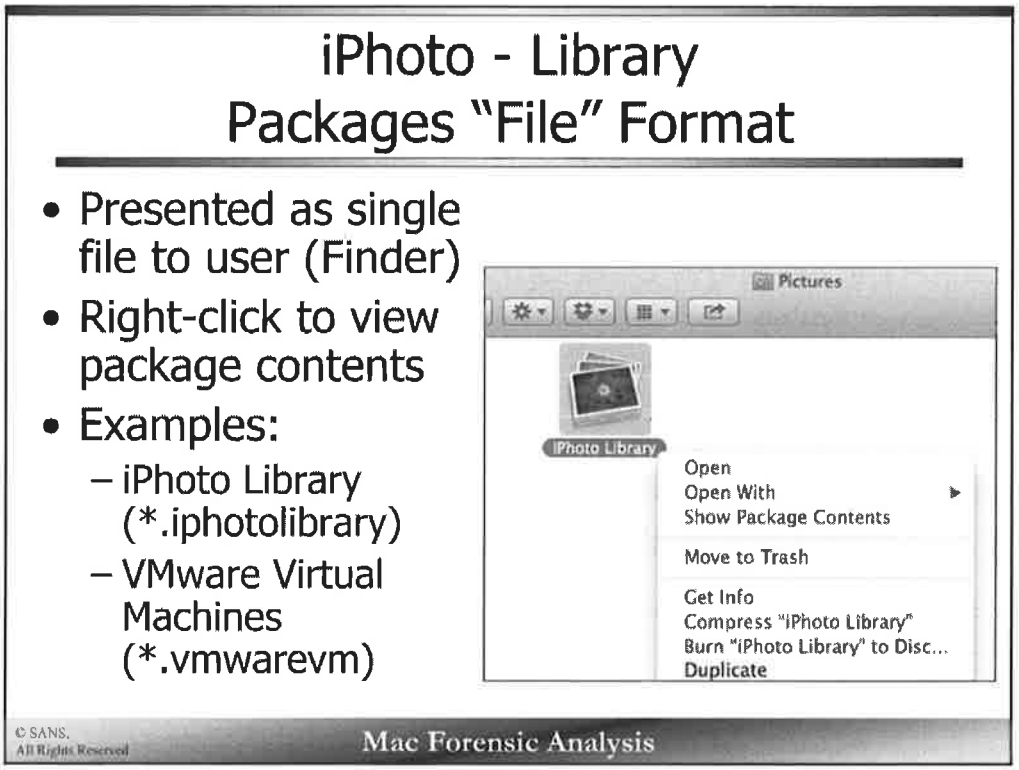
The contact information can be correlated by associating the `ZMAILRECENT` value in the `ZABCDMAILRECENT` table.

TABLE ZABCDLASTEMAILDAT Search Show All

Z_PK	Z_ENT	Z_OPT	ZMAILRECENT	ZDATE
547	1	1	69	371249156.6101949
549	1	1	68	371249156.6101949
553	1	1	68	371246276.8966031
554	1	1	9	371245392.16752696
551	1	1	9	371226193.829339
533	1	1	7	371164631.758471
536	1	1	50	371164631.75404
534	1	1	33	371164631.749931
538	1	1	17	371164631.747306
519	1	1	8	371155447.909068
524	1	1	10	371155447.905211
516	1	1	67	371142030.373749
515	1	1	7	371080239.1065941
514	1	1	7	371080178.97163105
512	1	1	7	371080120.9240551
513	1	1	7	371074525.60124207
540	1	1	7	371074525.601242
511	1	1	7	371070267.57922506
530	1	1	7	371070267.579225



iPhoto is the native photo and movie organizer application on OS X.



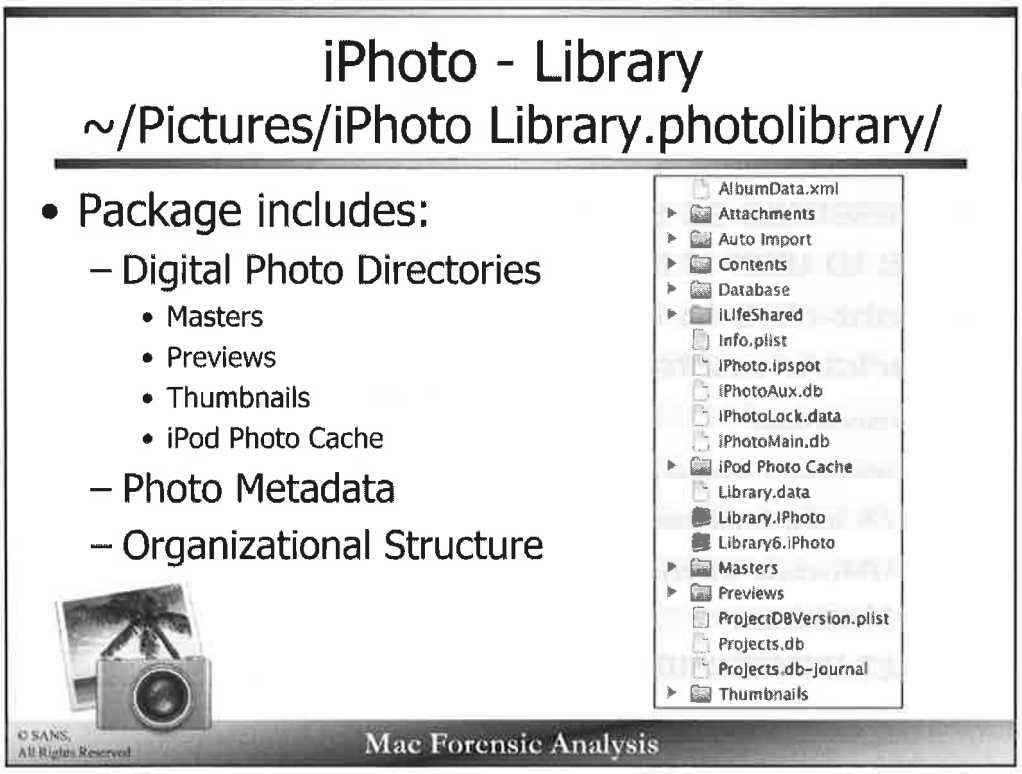
Packages are directories that are presented to the user as a single file in Finder. To view the contents of a package, right-click and choose "`Show Package Contents`", or view it in a terminal window.

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

References:

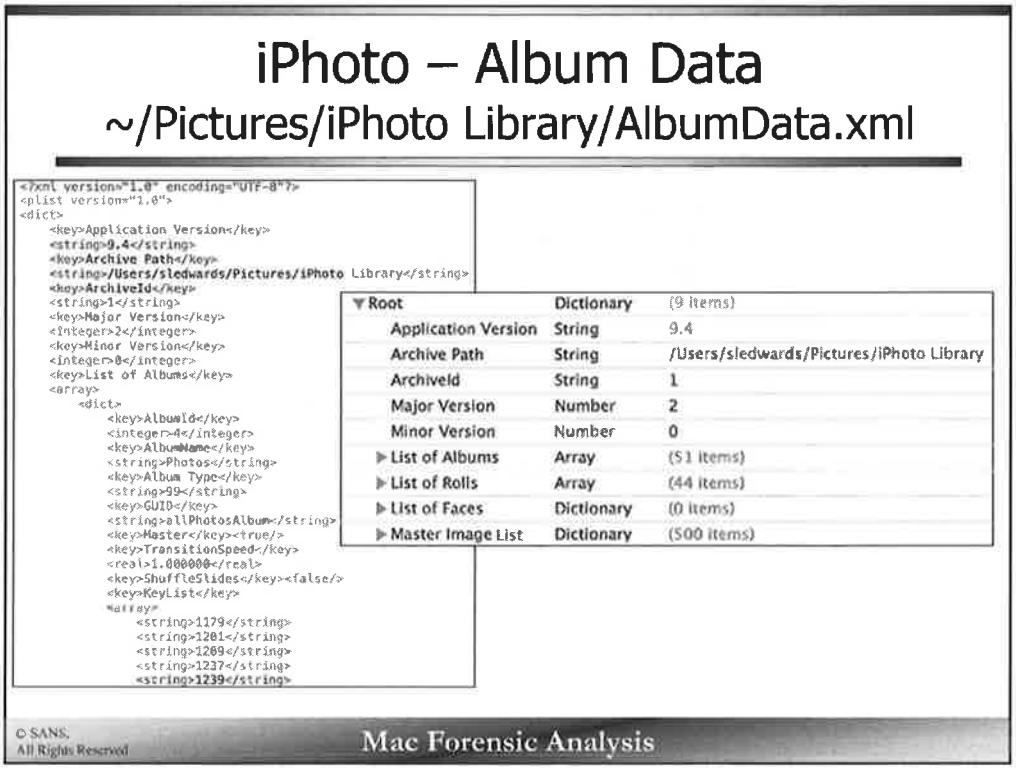
Apple Developer Website: About Bundles

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



The iPhoto library is located in the `~/Pictures/iPhoto Library` directory, which is actually a package. This package contains all the data that makes up the iPhoto Library:

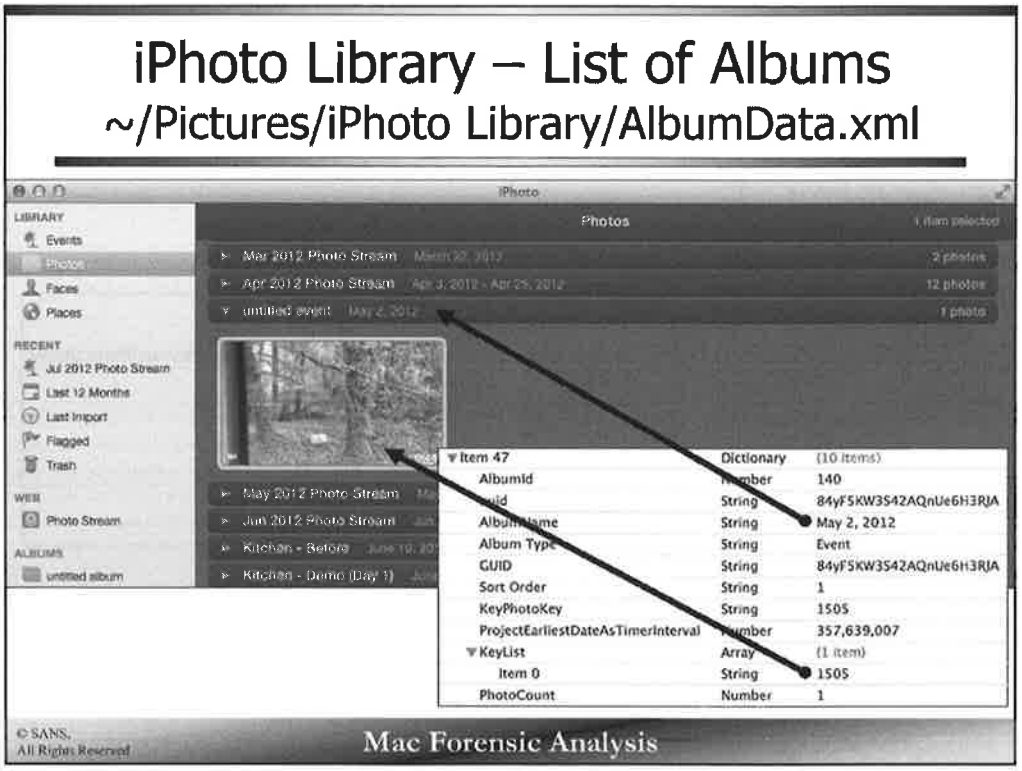
- Photos – Masters, Preview, Thumbnails
- Metadata Databases and Property Lists
- Project Databases



Each album in the iPhoto library stores its metadata in the `AlbumData.xml` file located in the iPhoto Library package. This XML file can be changed to a property list by changing the file extensions from `.xml` to `.plist`.

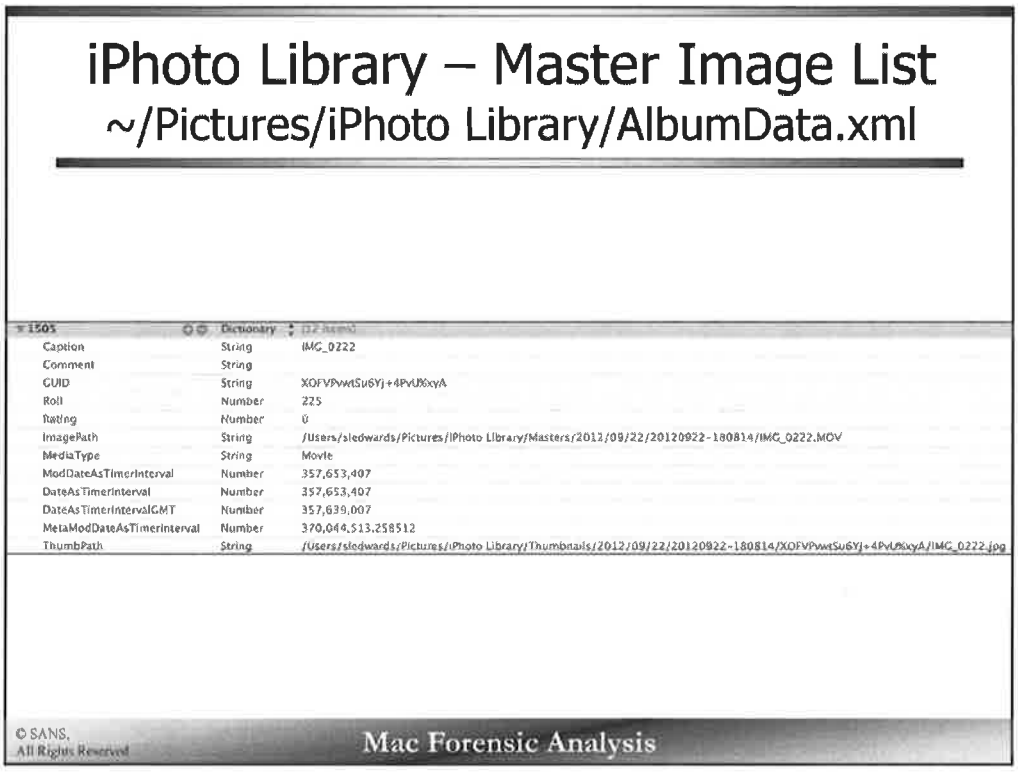
The slide shows an example of the XML file on the left, and the same file in property list format on the right.

This file contains data about each album, not just the user created albums but also the "default" albums.



The `List of Albums` keys in the `AlbumData.xml` file contains the information for each album including:

- Album ID
- Album Name (Often labeled by the "event date" by default.)
- Album Creation Date in WebKit date format
- The `KeyList` key contains the items in the Album, referenced by a number



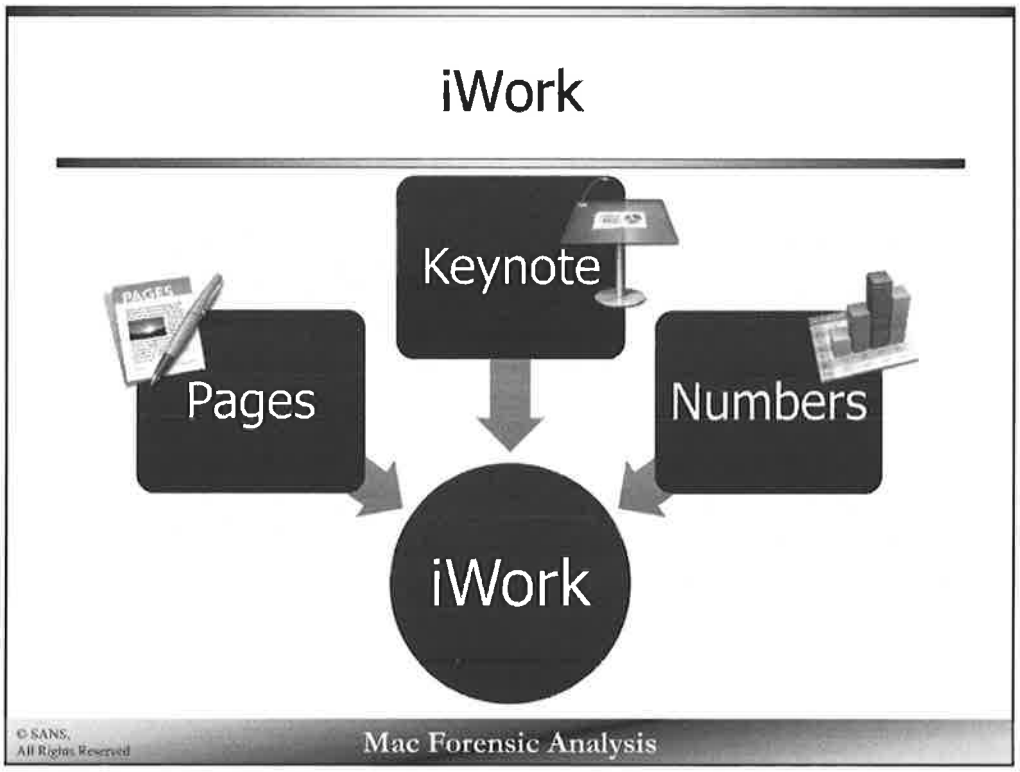
The image number from the album listed in the last slide can be correlated to a number in the Master Image List key in the same `AlbumData.xml` file.

Xcode makes searching for specific keys and data items easy in very lengthy XML/property list files. Highlighted in yellow in the screenshot above is the search term "`1505`", the image number from the May 2nd Album.

In the example above, the photo contains metadata items such as:

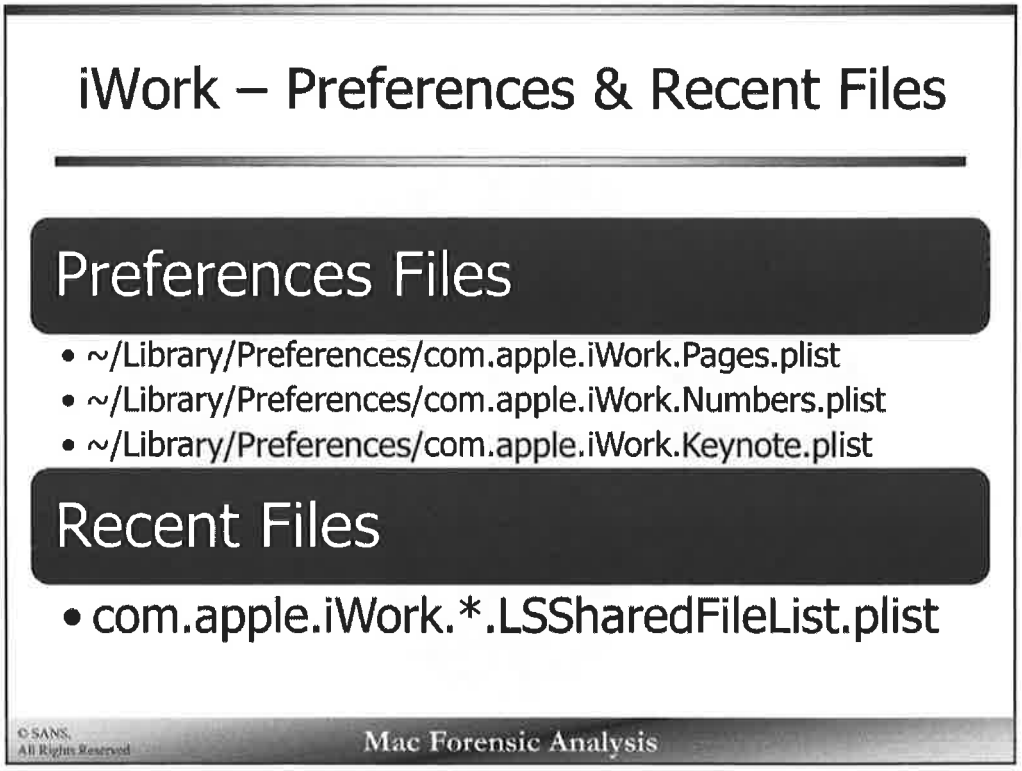
- Photo Name (`IMG_0222`)
- User Comments
- Path to the photo/movie file on disk
- Path to thumbnail on disk
- Media Type
- Dates in WebKit Format
    - Modification Date (`ModDateAsTimerInterval`)
    - Created Date (`DateAsTimerInterval`)
    - GMT Created Date (`DateAsTimerIntervalGMT`)
    - Metadata Modification Date (`MetaModDateAsTimerInterval`)

▼1505	Dictionary	(12 items)
Caption	String	IMG_0222
Comment	String	
GUID	String	XOFVPvwtSu6Yj+4PvUXxyA
Roll	Number	225
Rating	Number	0
ImagePath	String	/Users/sledwards/Pictures/iPhoto Library/Masters/2012/09/22/20120922-180814/IMG_0222.MOV
MediaType	String	Movie
ModDateAsTimerInterval	Number	357,653,407
DateAsTimerInterval	Number	357,653,407
DateAsTimerIntervalGMT	Number	357,639,007
MetaModDateAsTimerInterval	Number	370,044,513.258512
ThumbPath	String	/Users/sledwards/Pictures/iPhoto Library/Thumbnails/2012/09/22/20120922-180814/XOFVPvwtSu6Yj+4PvUXxyA/IMG_0222.jpg



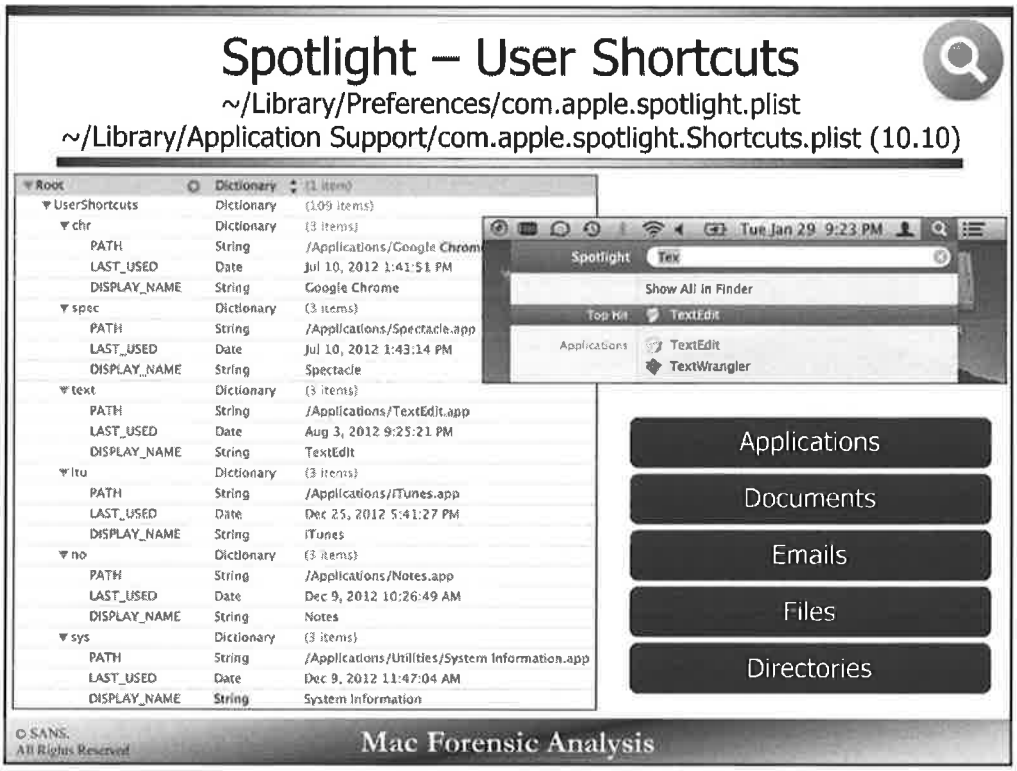
iWork is comprised of three applications; Pages, Keynote, and Numbers.

Pages – Similar to Microsoft Word, it is a document editor.
Keynote – Analogous to Microsoft PowerPoint, slide presentations.
Numbers – Like Microsoft Excel, spreadsheets editor.

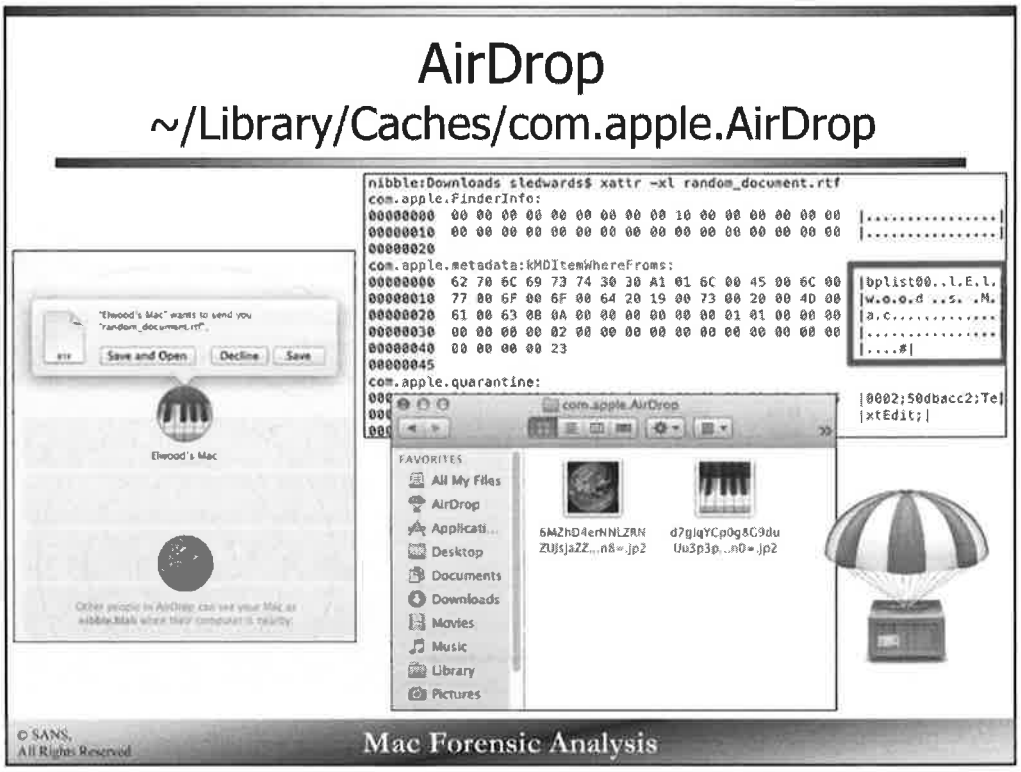


Each iWork application has its own preferences property list in the `~/Library/Preferences/` directory. Each iWork Preferences file may contain "last saved" directories and application specific configurations.

The `LSSharedFileList` property lists (one for each application, per user) contain the recently opened files for the specified application.



Spotlight is an application that is used to index and create a searchable database of the OS X system. Users can search for items using the Spotlight magnifying glass icon in the top-right of the GUI, shown above in the screenshot. These search terms are saved in the `com.apple.spotlight.plist` property list in the `~/Library/Preferences/` directory.



AirDrop is available on newer 10.7+ systems. This application allows users to "drop" files to other users in their vicinity. Note: this does not necessarily mean the user has to be on the same network.

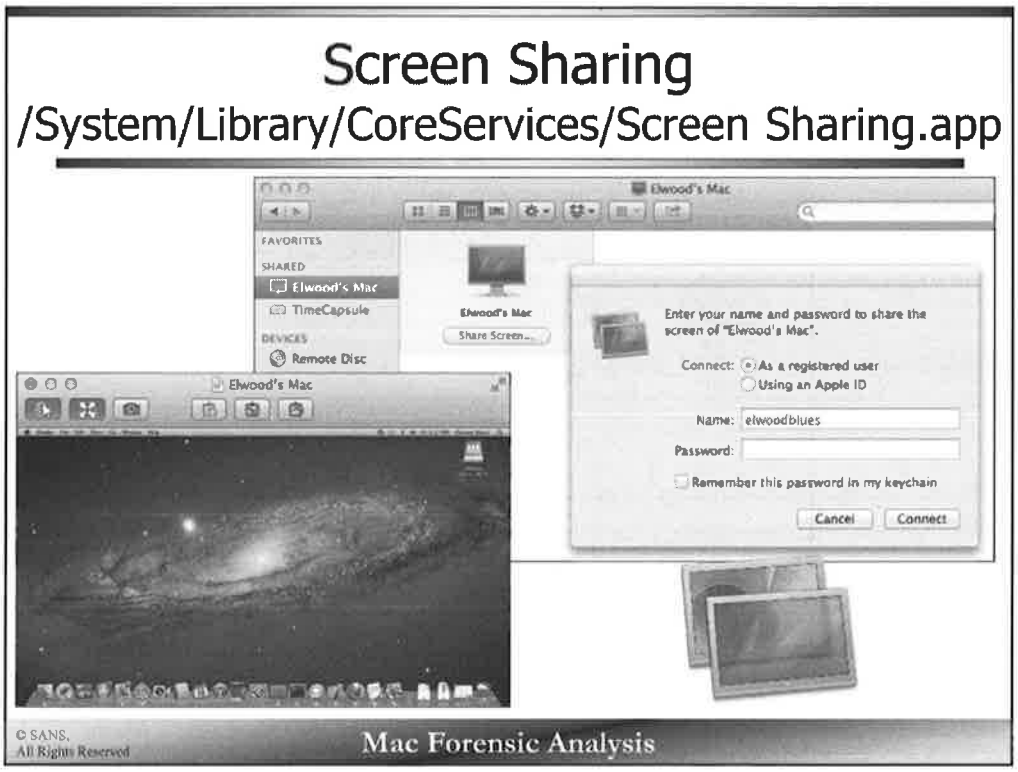
In the example on the left, Elwood's Mac wants to send a document. If this file is saved, it will, by default, be saved to the `~/Downloads` directory.

Extended attributes for the file on the recipient's system show the file was from Elwood's Mac, shown in the top screenshot.

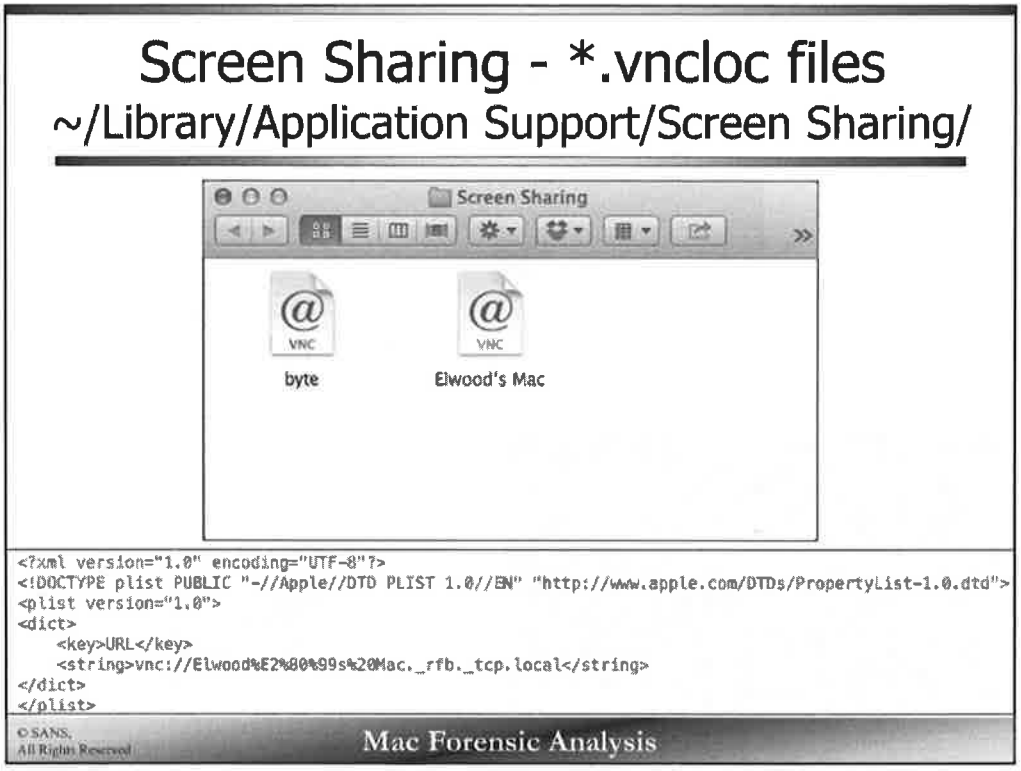
The `com.apple.AirDrop` directory located in `~/Library/Caches/` directory contains icons with a base64 formatted filename, this filename is unique to each system.

As of 10.10, this AirDrop directory no longer exists. However be on the lookout for extended attributes with iPhone or iPad designators. With 10.10, users are now able to AirDrop with their iDevices.

```
word:Downloads oompa$ xattr -xl IMG_1626.JPG
com.apple.metadata:kMDItemWhereFroms:
00000000  62 70 6C 69 73 74 30 30 A2 01 02 56 69 50 68 6F  |bplist00...ViPho|
00000010  6E 65 58 6D 69 50 68 6F 6E 65 36 08 0B 12 00 00  |neXmiPhone6.....|
00000020  00 00 00 00 01 01 00 00 00 00 00 00 00 03 00 00  |................|
00000030  00 00 00 00 00 00 00 00 00 00 00 00 00 00 1B     |...............|
0000003e
```



The Screen Sharing application, located in `/System/Library/CoreServices/` (rather than `/Applications`) is used to connect to other systems using a VNC protocol.

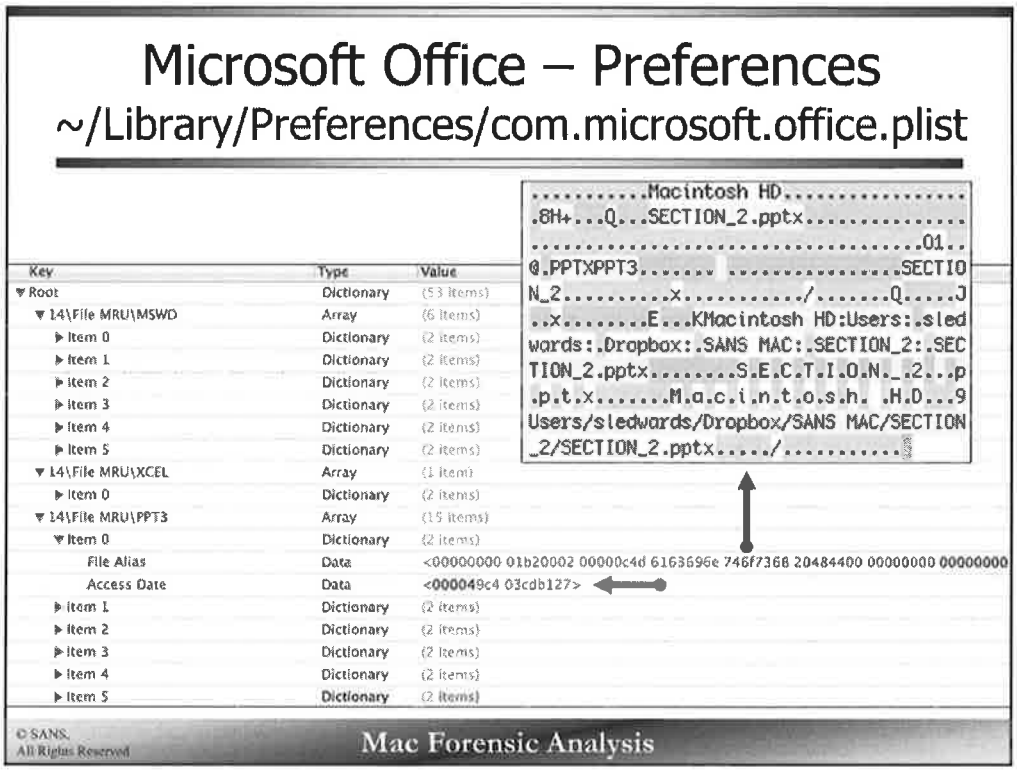


A `.vncloc` file will be automatically created for each system when an outgoing screen sharing connection has been made. These files are an XML property list file that contains the connection address (like the one shown in the example above).

A list of recent connections can be found in the `com.apple.ScreenSharing.LSSharedFileList.plist` property list file.

References:

RFB = Remote Framebuffer Protocol - http://en.wikipedia.org/wiki/RFB_protocol

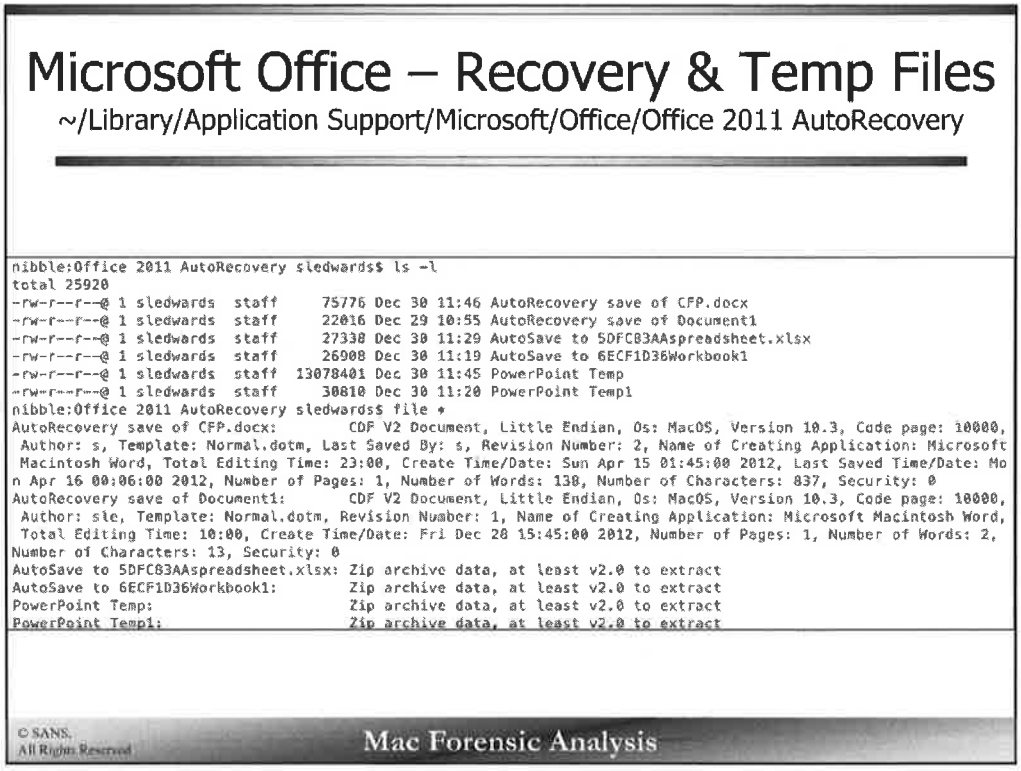


The always ubiquitous Microsoft Office uses the `com.microsoft.office.plist` property list to store it's MRU files. Each application in Office (Word, Excel, PowerPoint, etc.) has a separate MRU key containing the most recently used files for that application.

Each MRU has Alias data that can be extracted to view additional data (shown above) and an access date in Mac OS format.

The access date can be interpreted as follows:

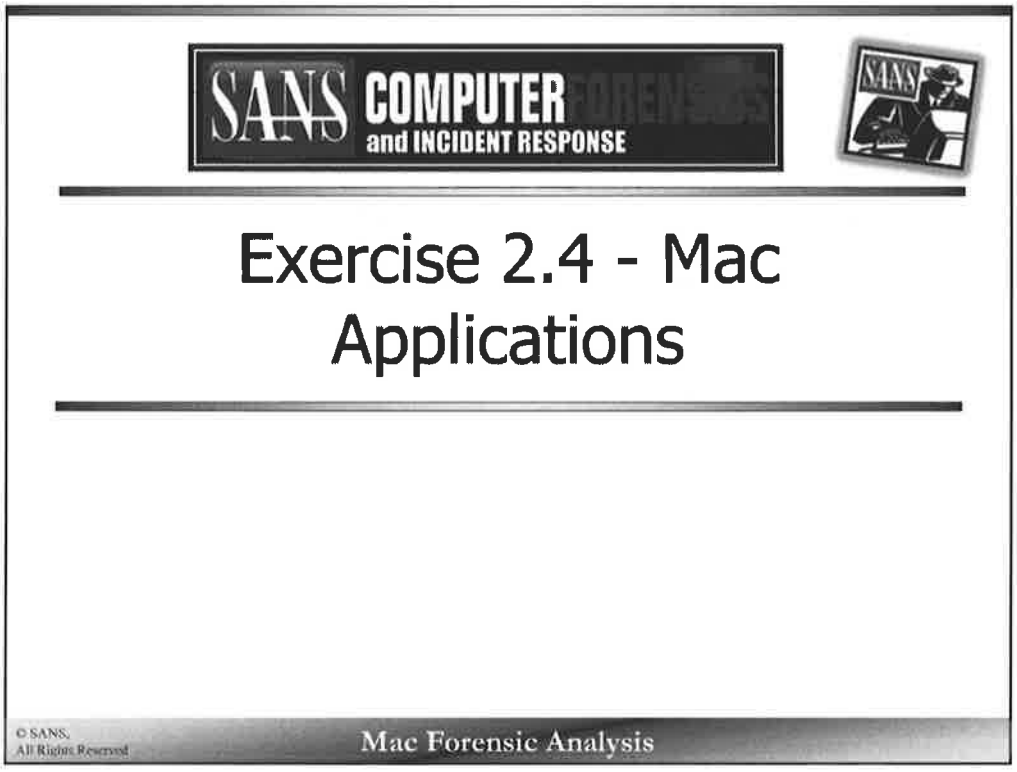
- Select the middle four bytes (i.e.: 0x49C403CD)
- Put these bytes into the Calculator.app
- Select "Byte Flip" to change it to little endian. (0xCD03C449)
- Calculate the numeric Mac OS date (ie: 3439576137)
- Use Epoch Converter to make the date human readable (2012-12-28 21:48:57 Fri UTC)



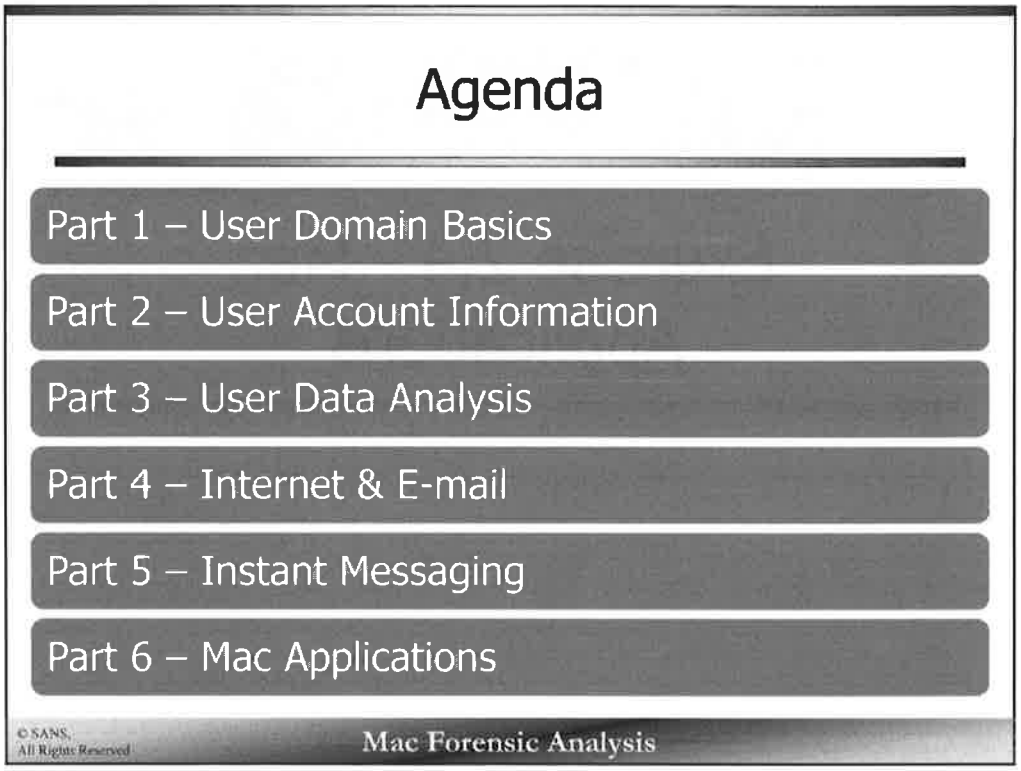
Each Microsoft Office 2011 application stores recovery and temporary files, each with a slightly different format or naming scheme, in the `Office 2011 AutoRecovery` directory.

The screenshot shows Word documents are named with "`AutoRecovery`", Excel documents use "`AutoSave`", and PowerPoint presentations use "`Temp`".

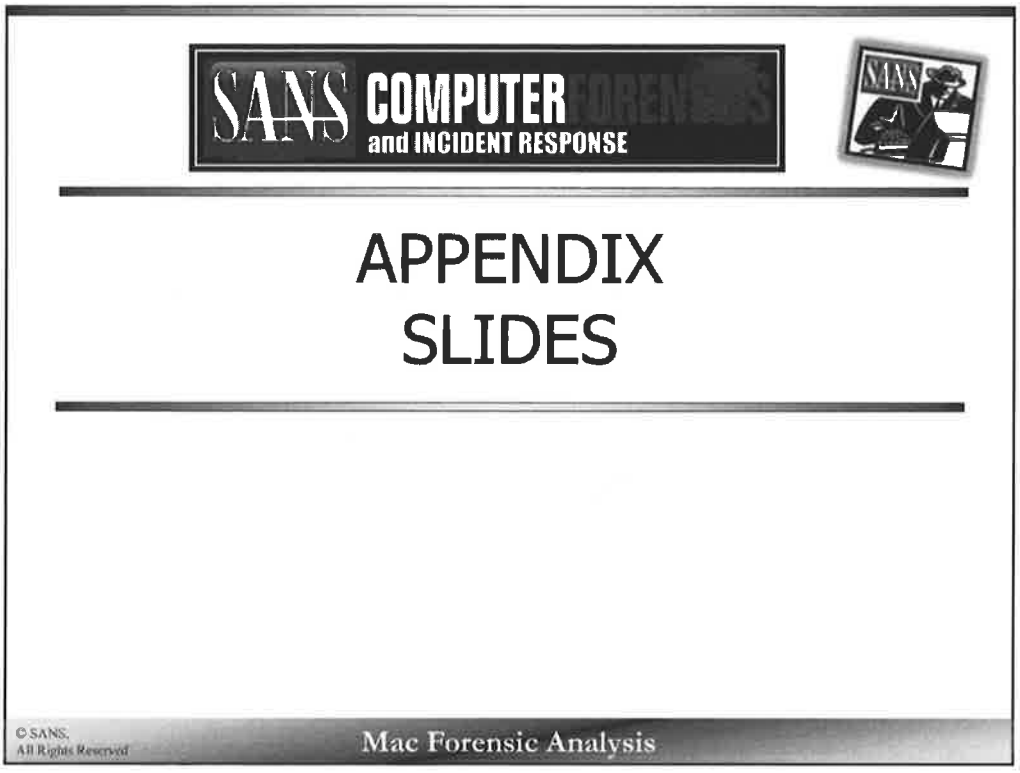
Excel spreadsheets and PowerPoint presentations are saved in a ZIP archive format, while Word documents are saved as documents.



This page intentionally left blank.

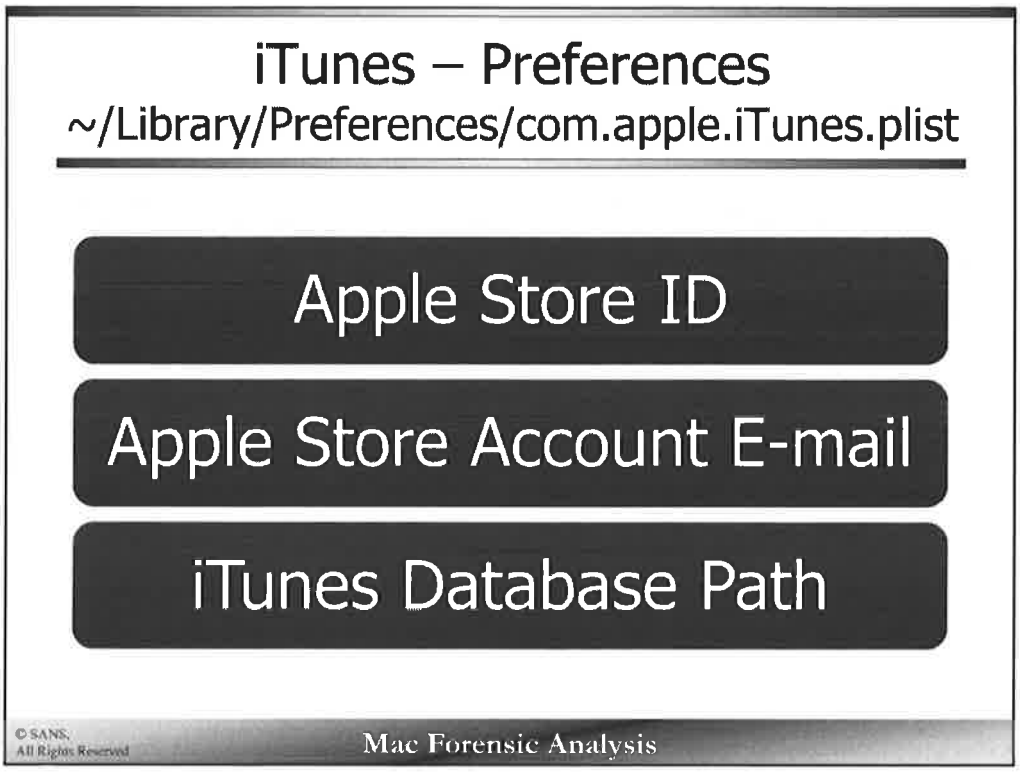


This page intentionally left blank.



This page intentionally left blank.

iTunes is the OS X native application for media organization. iTunes is the default application to play music, listen to podcasts, download iPhone applications, and watch movies and TV shows. It is also used to purchase and download these media types.



The iTunes preferences are stored in the `com.apple.iTunes.plist` file. This file may include information related to the various synced iDevices, Apple Account IDs and e-mail address, and the path to the iTunes database.

The iTunes music library is stored as an XML file called `iTunes Music Library.xml` located in the `~/Music/iTunes/` directory. While this is a plaintext XML file, it is much easier to read as a property list file – just change the file extension to `.plist` rather than `.xml` to open it in Xcode.

Note: iTunes Libraries before 10.4 did not have the XML file extension.

The `Date` key in the file contains the time the library was last updated. The `Application Version` key keeps track of the iTunes version that is being used, while the `Music Folder` key keeps track of where the iTunes media is located.

The screenshot on the left shows the Tracks key in the iTunes Library XML file contains each "`track`", each music file, TV show, Movie, etc.

Each track is labeled by a Track ID number and contains all the metadata of the item, including:

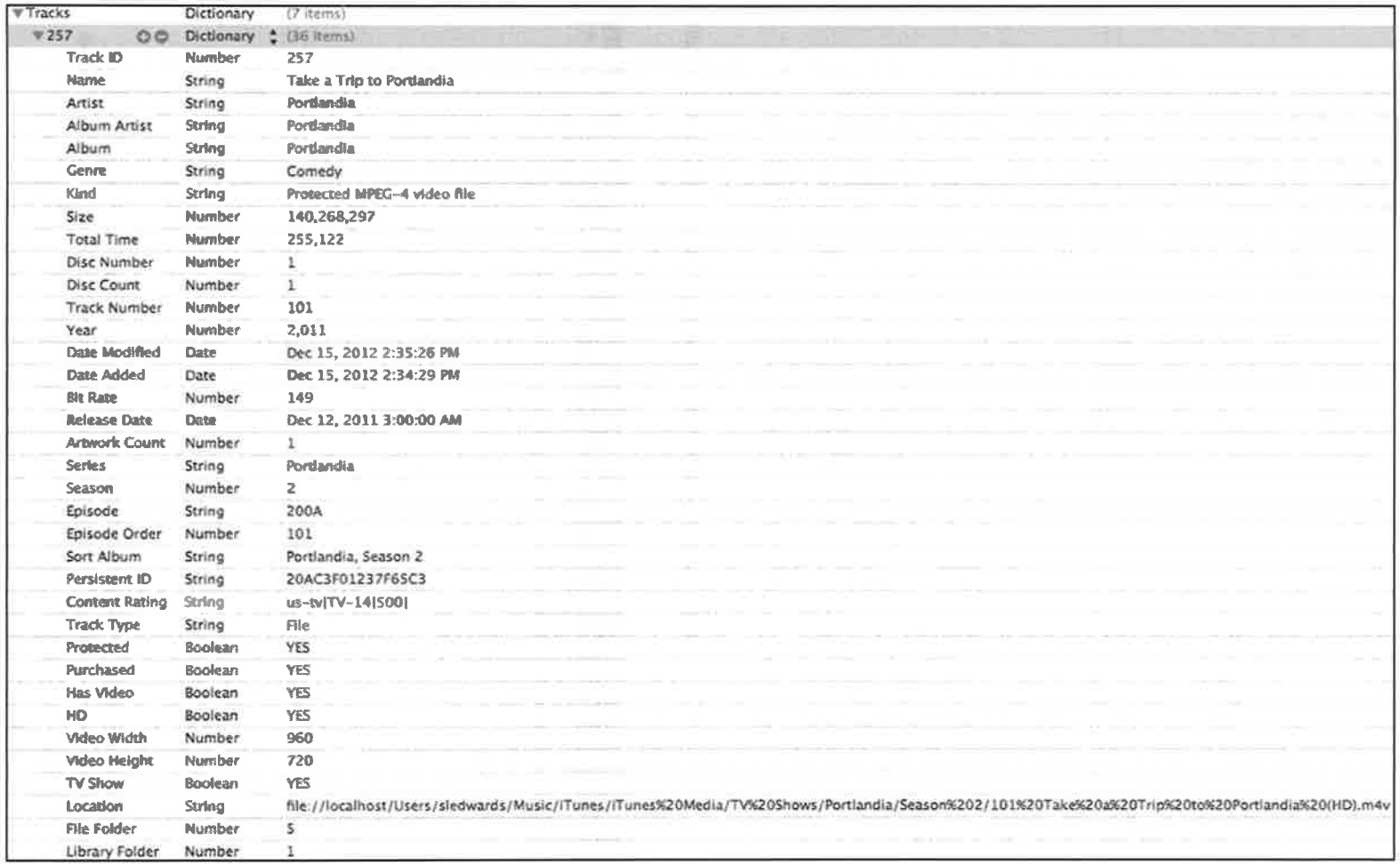
- Track Name
- Artist Information
- Album Information
- Size
- Length
- Release Dates
- Purchase Dates
- Bit/Sample Rates
- Purchase Information

Note: The Date Modified and Date Added dates may also be the original purchase dates of the media, not necessarily the date the media was added to this specific library.

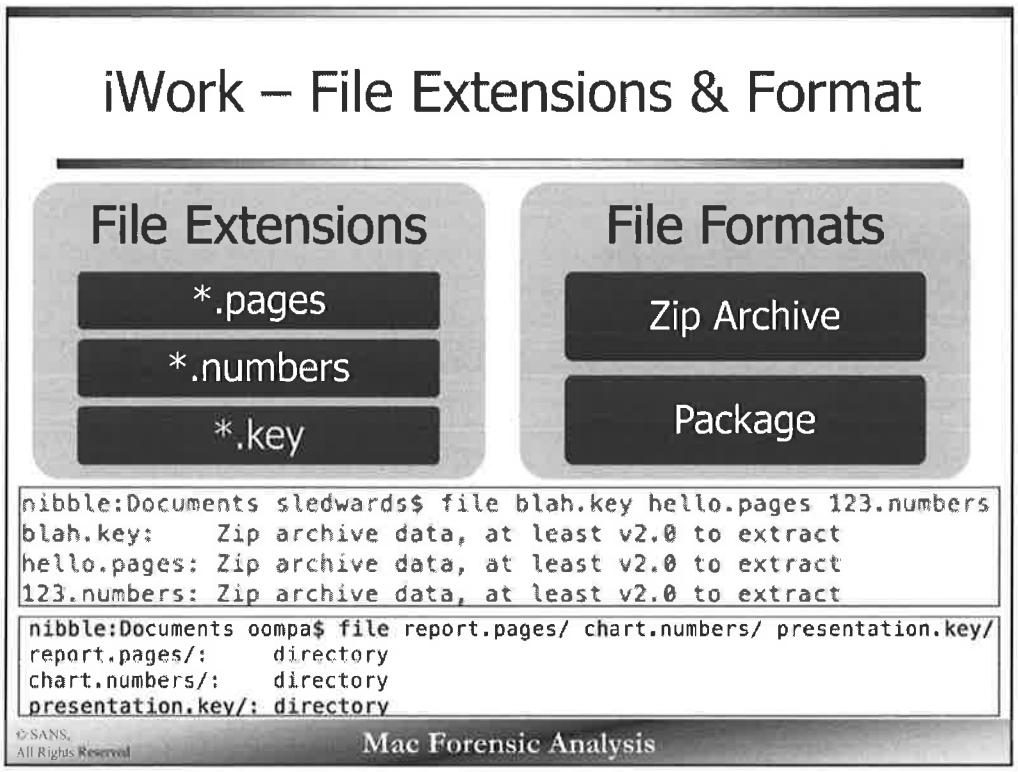
The screenshot on the right shows the iTunes purchase information from the EXIF data from the media file. Information includes the purchasers name, Apple Store account, and the purchase date. This data can be extracted using exiftool (available at http://www.sno.phy.queensu.ca/~phil/exiftool/).

▼Tracks	Dictionary	(64 items)
▼206	Dictionary	(27 items)
Track ID	Number	206
Name	String	Falling for You
Artist	String	Jem
Album Artist	String	Jem
Composer	String	Brian Higgins, Jem Griffiths & Nick Coler
Album	String	Finally Woken
Genre	String	Electronic
Kind	String	Purchased AAC audio file
Size	Number	4,160,323
Total Time	Number	256,974
Disc Number	Number	1
Disc Count	Number	1
Track Number	Number	9
Track Count	Number	11
Year	Number	2,004
Date Modified	Date	May 12, 2005 8:16:07 PM
Date Added	Date	May 12, 2005 8:16:07 PM
Bit Rate	Number	128
Sample Rate	Number	44,100
Release Date	Date	Mar 23, 2004 7:00:00 PM
Artwork Count	Number	1
Sort Album	String	Finally Woken
Sort Artist	String	Jem
Sort Name	String	Falling for You
Persistent ID	String	02A182C9A11C3BC2
Track Type	String	Remote
Purchased	Boolean	YES

Another example of an entry in Tracks – This one is an example of a TV show.



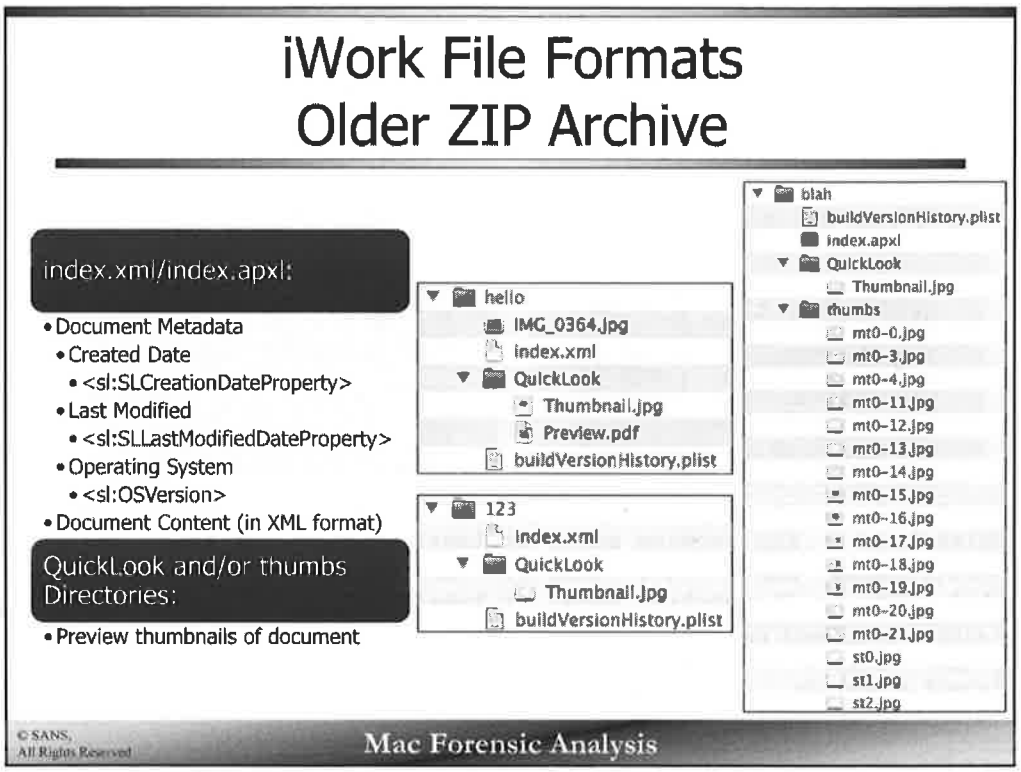
Tracks	Dictionary	(7 items)
257	Dictionary	(36 items)
Track ID	Number	257
Name	String	Take a Trip to Portlandia
Artist	String	Portlandia
Album Artist	String	Portlandia
Album	String	Portlandia
Genre	String	Comedy
Kind	String	Protected MPEG-4 video file
Size	Number	140,268,297
Total Time	Number	255,122
Disc Number	Number	1
Disc Count	Number	1
Track Number	Number	101
Year	Number	2,011
Date Modified	Date	Dec 15, 2012 2:35:26 PM
Date Added	Date	Dec 15, 2012 2:34:29 PM
Bit Rate	Number	149
Release Date	Date	Dec 12, 2011 3:00:00 AM
Artwork Count	Number	1
Series	String	Portlandia
Season	Number	2
Episode	String	200A
Episode Order	Number	101
Sort Album	String	Portlandia, Season 2
Persistent ID	String	20AC3F01237F6SC3
Content Rating	String	us-tv|TV-14|500|
Track Type	String	File
Protected	Boolean	YES
Purchased	Boolean	YES
Has Video	Boolean	YES
HD	Boolean	YES
Video Width	Number	960
Video Height	Number	720
TV Show	Boolean	YES
Location	String	file://localhost/Users/sledwards/Music/iTunes/iTunes%20Media/TV%20Shows/Portlandia/Season%202/101%20Take%20a%20Trip%20to%20Portlandia%20(HD).m4v
File Folder	Number	5
Library Folder	Number	1



Each iWork application saves its associated files with a different file extension.

- Pages saves its files with a `.pages` extension.
- Numbers saves its files with a `.numbers` extension.
- Keynote saves its files with a `.key` extension.

iWork files may be formatted as a ZIP archive or the Package format as shown in the screenshots above. Apple has switched between these formats depending on the version of iWork used.

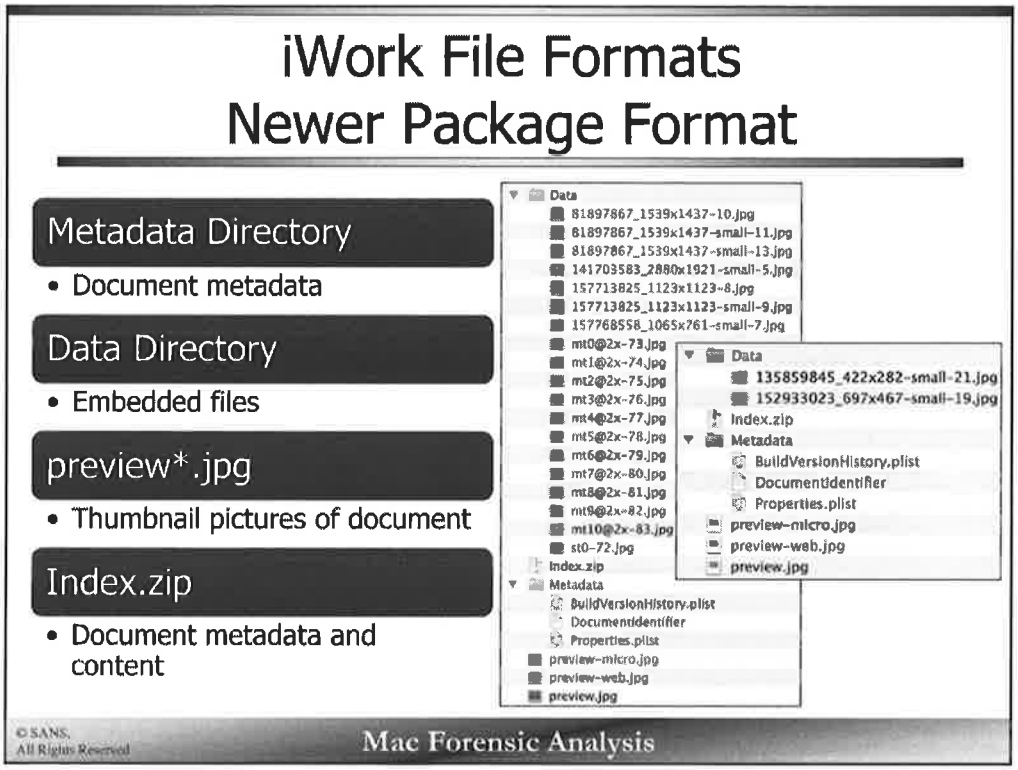


A Pages document (`hello.pages`), a Numbers spreadsheet (`123.numbers`), and a Keynote presentation (`blah.keynote`) were unzipped and shown in the Finder application in the screenshots above.

Each ZIP file will consist of an `index.xml/index.xpxl` XML file containing the metadata and file content for the iWork document. Files embedded into the document (`IMG_0364.jpg`) are in the root of the directory. Similar to Pages and Numbers, Keynote stores its data in an XML file called `index.apxl`. Although it uses a different file extension, this file is still an XML-based file.

The `QuickLook` and `thumbs` directories contain a PDF and/or JPG thumbnail pictures of the document. These files are used in various views of the OS X Finder, to show what the document looks like to the user.

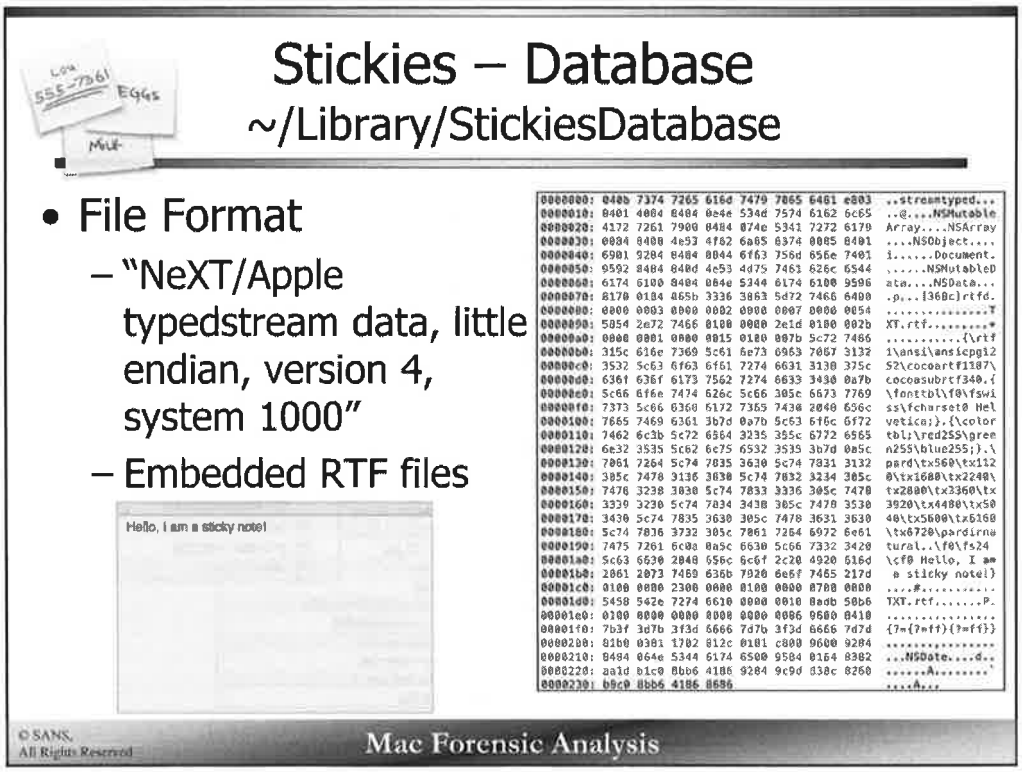
Keynote stores individual JPGs for each of the slides named `st<somenumber>.jpg` in the `thumbs` directory. The files starting with `mt`, contain previews for the slide templates.



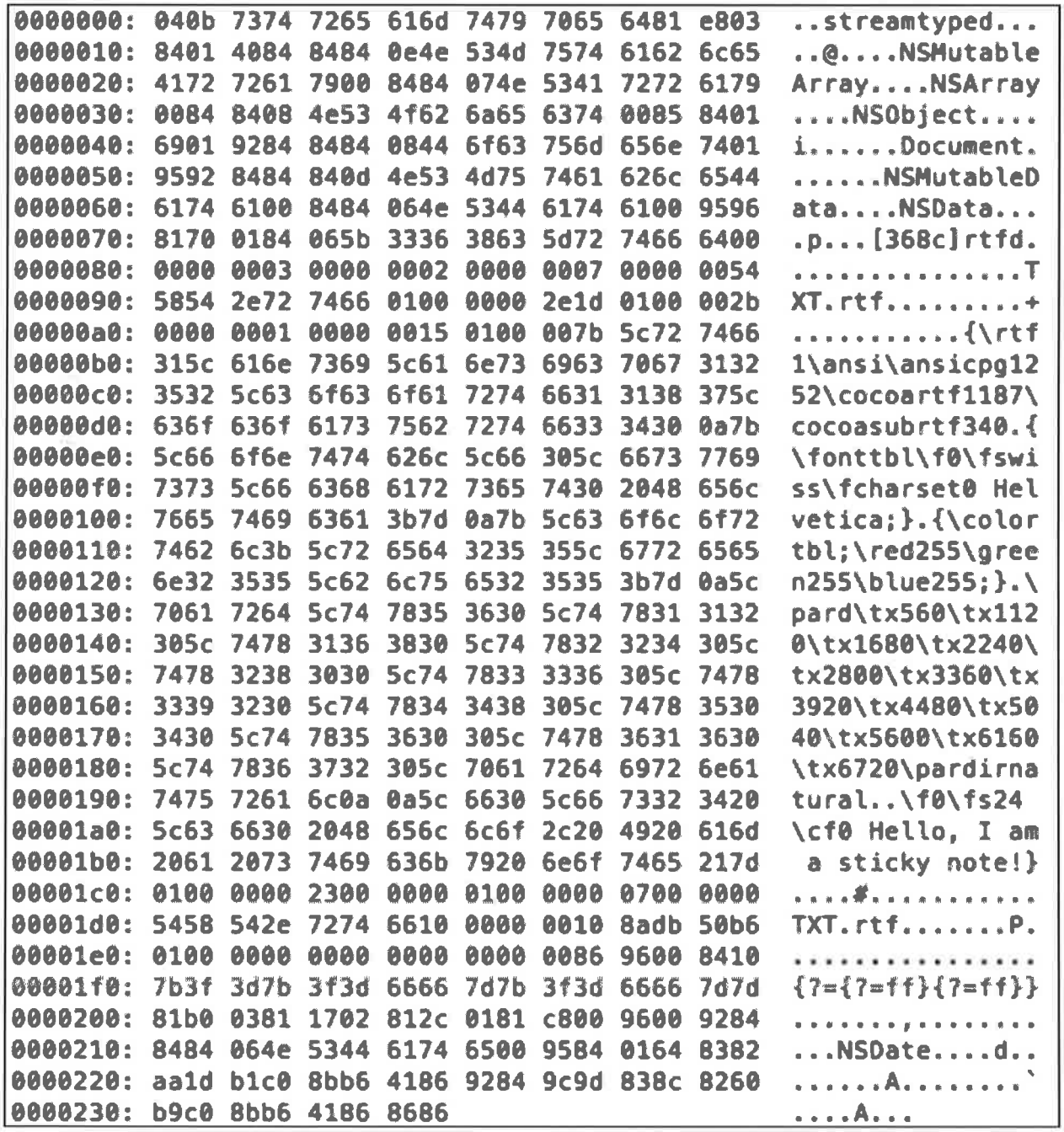
The newer package iWork format contains similar data for each document. Document metadata can be found in the Metadata directory as well as in the proprietary files located in the `Index.zip` file.

The document content is also stored in the `Index.zip` file in a proprietary format, this should be noted as a simple keyword search may not find the information you are looking for. Be sure to expand all archives!

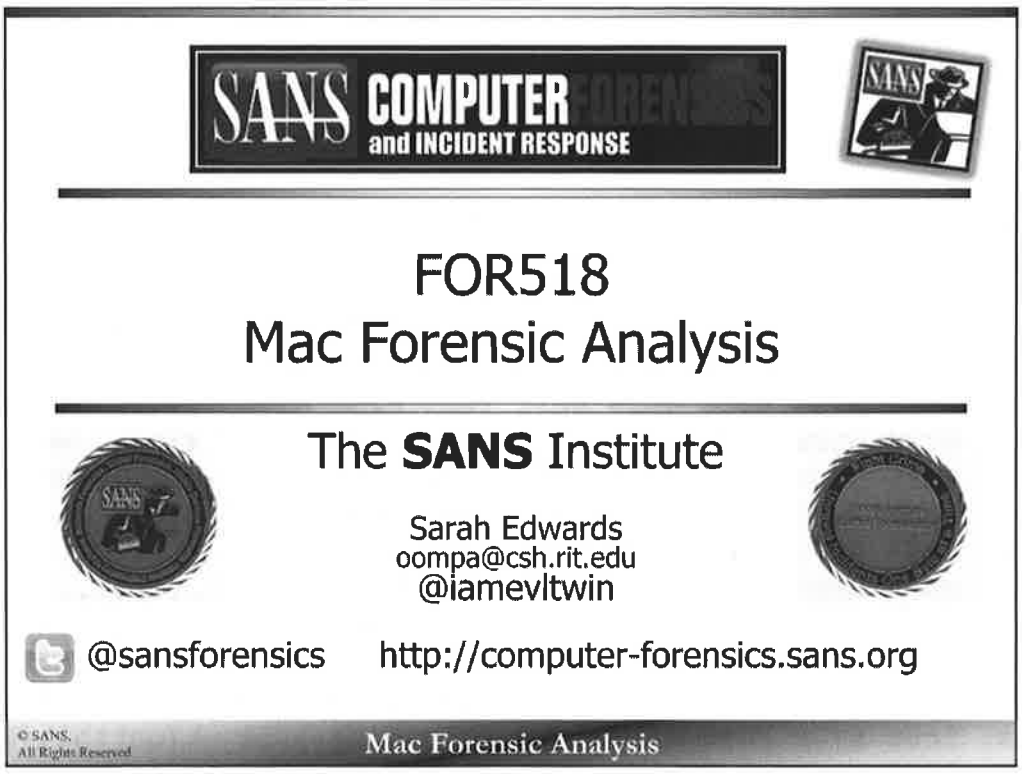
Thumbnail pictures (`preview*.jpg`) of the documents can be found in various formats in the root of the directory.



Stickies is a native application that is analogous to those yellow sticky notes that often populate our desks. These notes are stored in the `StickiesDatabase` in the `~/Library` directory. The file format for this database is a throw back to the NeXT days, while not immediately parse-able you can extract the Rich Text Files (RTF) from the file.



```
0000000: 040b 7374 7265 616d 7479 7065 6481 e803  ..streamtyped...
0000010: 8401 4084 8484 0e4e 534d 7574 6162 6c65  ..@....NSMutable
0000020: 4172 7261 7900 8484 074e 5341 7272 6179  Array....NSArray
0000030: 0084 8408 4e53 4f62 6a65 6374 0085 8401  ....NSObject....
0000040: 6901 9284 8484 0844 6f63 756d 656e 7401  i......Document.
0000050: 9592 8484 840d 4e53 4d75 7461 626c 6544  ......NSMutableD
0000060: 6174 6100 8484 064e 5344 6174 6100 9596  ata....NSData...
0000070: 8170 0184 065b 3336 3863 5d72 7466 6400  .p...[368c]rtfd.
0000080: 0000 0003 0000 0002 0000 0007 0000 0054  ...............T
0000090: 5854 2e72 7466 0100 0000 2e1d 0100 002b  XT.rtf.........+
00000a0: 0000 0001 0000 0015 0100 007b 5c72 7466  ...........{\rtf
00000b0: 315c 616e 7369 5c61 6e73 6963 7067 3132  1\ansi\ansicpg12
00000c0: 3532 5c63 6f63 6f61 7274 6631 3138 375c  52\cocoartf1187\
00000d0: 636f 636f 6173 7562 7274 6633 3430 0a7b  cocoasubrtf340.{
00000e0: 5c66 6f6e 7474 626c 5c66 305c 6673 7769  \fonttbl\f0\fswi
00000f0: 7373 5c66 6368 6172 7365 7430 2048 656c  ss\fcharset0 Hel
0000100: 7665 7469 6361 3b7d 0a7b 5c63 6f6c 6f72  vetica;}.{\color
0000110: 7462 6c3b 5c72 6564 3235 355c 6772 6565  tbl;\red255\gree
0000120: 6e32 3535 5c62 6c75 6532 3535 3b7d 0a5c  n255\blue255;}.\
0000130: 7061 7264 5c74 7835 3630 5c74 7831 3132  pard\tx560\tx112
0000140: 305c 7478 3136 3830 5c74 7832 3234 305c  0\tx1680\tx2240\
0000150: 7478 3238 3030 5c74 7833 3336 305c 7478  tx2800\tx3360\tx
0000160: 3339 3230 5c74 7834 3438 305c 7478 3530  3920\tx4480\tx50
0000170: 3430 5c74 7835 3630 305c 7478 3631 3630  40\tx5600\tx6160
0000180: 5c74 7836 3732 305c 7061 7264 6972 6e61  \tx6720\pardirna
0000190: 7475 7261 6c0a 0a5c 6630 5c66 7332 3420  tural..\f0\fs24
00001a0: 5c63 6630 2048 656c 6c6f 2c20 4920 616d  \cf0 Hello, I am
00001b0: 2061 2073 7469 636b 7920 6e6f 7465 217d   a sticky note!}
00001c0: 0100 0000 2300 0000 0100 0000 0700 0000  ....#...........
00001d0: 5458 542e 7274 6610 0000 0010 8adb 50b6  TXT.rtf.......P.
00001e0: 0100 0000 0000 0000 0000 0086 9600 8410  ................
00001f0: 7b3f 3d7b 3f3d 6666 7d7b 3f3d 6666 7d7d  {?={?=ff}{?=ff}}
0000200: 81b0 0381 1702 812c 0181 c800 9600 9284  .......,........
0000210: 8484 064e 5344 6174 6500 9584 0164 8382  ...NSDate....d..
0000220: aa1d b1c0 8bb6 4186 9284 9c9d 838c 8260  ......A........`
0000230: b9c0 8bb6 4186 8686                      ....A...
```



Author: Sarah Edwards
oompa@csh.rit.edu
http://twitter.com/iamevltwin
http://twitter.com/sansforensics