Directory:

/bin	Contains command utilities, just like Linux.
/sbin	Holds system binaries for managing the OS
/dev	Device files live here, as in most Unix-like systems
/opt	For optional software.
/private/var	Stores variable data like logs and system audits
/private/etc	System configuration files, such as hosts and passwd.
/private/tmp	Temporary files directory (deleted automatically over time).

MacOS Timestamps:

stat -x <filename>	# Shows Access, Modify, and Change timestamps in seconds
For nanosecond accuracy, use:	
stat -f %Fa <filename>	# Access time
stat -f %Fm <filename>	# Modification time
stat -f %Fc <filename>	# Change time
GetFileInfo <filename>	Command gives you additional details about the file,

Command to Mount in macOS

```
sudo su
mkdir /Volumes/apfs_images
mkdir /Volumes/apfs_mounts
xmount -- in ewf evidencecapture.E01 -- out dmg /Volumes/apfs_images
hdiutil attach -nomount /Volumes/apfs_images/evidencecapture.dmg
diskutil ap list
diskutil ap unlockvolume <Disk GUID> -nomount
mount_apfs -o rdonly,noexec,noowners /dev/disk# /Volumes/apfs_mounts/
```

Command to Mount in Linux

1. **Install APFS FUSE Drivers**: First, you'll need to install the necessary dependencies and clone the APFS FUSE repository from GitHub.

```
sudo apt update
sudo apt install libicu-dev bzip2 cmake libz-dev libbz2-dev fuse3 clang git
libattr1-dev libplist-utils -y
cd /opt
git clone https://github.com/sgan81/apfs-fuse.git
cd apfs-fuse
git submodule init
git submodule update
mkdir build
cd build
cmake ..
```

```
make
ln /opt/afps-fuse/build/apfs-dump /usr/bin/apfs-dump
ln /opt/afps-fuse/build/apfs-dump-quick /usr/bin/apfs-dump-quick
ln /opt/afps-fuse/build/apfs-fuse /usr/bin/apfs-fuse
ln /opt/afps-fuse/build/apfsutil /usr/bin/apfsutil
```

2. **Mount the Image**: After setting up FUSE, you can mount the image using this command:

```
mkdir /mnt/apfs_mount  #create mount point
cd /mnt/ewf_mount #change to the directory where the E01 file is located.
apfs-fuse -o ro,allow_other ewf1 /mnt/apfs_mount # mount the image read
only
```

*If you want a script to automate this for **Debian-based distros** (like Ubuntu), check out the one available at this link.*

https://github.com/TazWake/Public/blob/master/Bash/apfs_setup.sh

---

Evidence Profiling:

1. Device Information:

   Location: `/System/Library/CoreServices/SystemVersion.plist`

   Use cat on a live system to view the .plist file contents.

2. Device Serial Number

   Location: `/root/private/var/folders/zz/zyxvpxvq6csfxvn_n00000sm00006d/C/`

   Files:

   - consolidated.db
   - cache_encryptedA.db
   - lockCache_encryptedA.db

3. Device Time Zone – Option 1
   Command: - ls -al /etc/localtime

4. Device Time Zone – Option 2

   `Location: /Library/Preferences/.GlobalPreferences.plist`

Command:(On live system or on MAC)

```
plutil -p /Library/Preferences/.GlobalPreferences.plist
```

<u>Note:-</u> *If location services are enabled, the automatic time zone update will regularly update this plist. However, when devices switch to static time zones, this plist may not be updated and it will point to the last automatic update location.*

To check If location service is enabled or not:

Location: `/Library/Preferences/com.apple.timezone.auto.plist`

If location services are enabled, the entry “active” will be set to 1 or true.

User Accounts:

Location: `/private/var/db/dslocal/nodes/Default/users/`
Location: `/private/var/db/dslocal/nodes/Default/groups/`

Network Setting:

Location: `/Library/Preferences/SystemConfiguration/NetworkInterfaces.plist`

MAC address will be in encoded format

echo "(encoded MAC)" | base64 –d | xxd

Network Configuration – Interfaces

Location: `/Library/Preferences/SystemConfiguration/preferences.plist`

DHCP Lease information:

Location: `/private/var/db/dhcpclient/leases/`

Use Cat Command

Persistence

1. Global Zsh Files:

/etc/zprofile	Alters the shell environment for all users, setting variables like $PATH
/etc/zshrc	Loads configuration settings for all users
/etc/zsh/zlogin	Runs after zshrc during login

2. User-Specific Zsh Files:

Located in User’s Hime directory (~)

~/.zshenv (optional)
~/.zprofile
~/.zshrc
~/.zlogin
~/.zlogout (optional)

User History:

~/.zsh_history
~/.zsh_sessions (directory**)**

Bash Equivalents

~/.bash_history
~/.bash_profile
~/.bash_login
~/.profile
~/.bashrc
~/.bash_logout

Installed shells:

It's not uncommon for users to install other shells. To verify which shells are installed, check the **/etc folder,** and look at the user's home directory for history files.

Launch Daemon (Launchd)

On live system:

```
launchctl list
```

On Disk Images

/Library/LaunchAgents	Per-user agents for all logged-in users
/Library/LaunchDaemons	System-wide daemons, installed by admins
/System/Library/LaunchDaemons	Apple-provided system-wide daemons
/System/Library/LaunchAgents	Apple-provided agents for user logins

User Jobs:

/Users/(username)/Library/LaunchAgents	Jobs specific to individual users are stored in

Cron Task:

System-wide cron jobs can be found in

/etc/cron.d/
/etc/cron.daily/
/etc/cron.hourly/
/etc/cron.monthly/
/etc/cron.weekly/

Cron Directory:

/etc/cron.weekly/

Cron Command:

```
crontab  -l
```

File Artifacts for User Preferences

configuration data in each user's	~/Library/Preferences	files are particularly useful during an investigation.
Browser Downloads:	com.apple.LaunchServices.QuarantineEventsV*	logs information about executable files downloaded from the internet
Recently Accessed Files:	**macOS Mojave and earlier:** com.apple.RecentItems.plist. **macOS Big Sur and later**: com.apple.shared.plist	

Finder Preferences:	com.apple.finder.plist	Finder app is configured, including information on mounted volumes.
Keychain Preferences	com.apple.keychainaccess.plist	provide clues about encrypted data access.

Common Log Locations

/var/log	Primary system logs.
/var/db/diagnostics	System diagnostic logs.
/Library/logs	System and application logs.
~/Library/Logs	User-specific logs.
/Library/Application Support/(App name)	Application logs.
/Applications	Logs for applications installed on the system.

Plain Text Logs

/var/log/system.log	General system diagnostics.
/var/log/DiskUtility.log	Disk mounting and management events.
/var/log/fsck_apfs.log	Filesystem-related events.
/var/log/wifi.log	Wi-Fi connections and known hotspots.
/var/log/appfirewall.log	Network events related to the firewall.

Binary Logs In MACoS

Apple System Logs (ASL)	**/var/log/asl/*.asl**
Apple Unified Logs (AUL)	**/var/db/diagnostics/Persist**
	/var/db/diagnostics/timesync
	/var/db/uuidtext/

File Type: .tracev3

AUL is the default logging format since macOS Sierra (10.12).

How to View AUL:

- View in live response: Use the log command or the **Console app.**

- File parsing: These logs are challenging to read manually. It's best to use specialized tools designed to extract and analyze AUL logs

---

For Live Log analysis:

1. User Last command: For most recent logins:
2. Reading ASL logs with syslog: (*syslog -f (filename).asl*)

---

mac_apt: macOS Artifact Parsing Tool

When you can't analyze logs on a macOS machine, especially during forensic analysis on Windows or Linux, **mac_apt** is a powerful, cross-platform solution.

https://github.com/ydkhatri/mac_apt

https://www.cyberengage.org/post/macos-incident-response-tactics-log-analysis-and-forensic-tools

---