COMPUTER FORENSICS LAB

London digital forensics & e-discovery · Established 2007

APFS and the Apple File System

Snapshots, Clones and Containers: Where Deleted Data Hides on Apple Devices

APFS, the Apple File System, is the way every modern Mac, iPhone and iPad organises its storage, and it behaves quite differently from the file systems forensic training was built on. It uses copy-on-write, so changing a file does not overwrite the old data but writes new data elsewhere, potentially leaving the old version recoverable. It takes snapshots, complete point-in-time pictures of the whole file system, that can preserve deleted and earlier data. It uses clones, space-saving copies that share data until changed. And it organises storage into containers holding several volumes that share free space. For the lawyer, the practical consequences are large: deleted and earlier versions of data survive on Apple devices more often, and in different places, than intuition suggests, snapshots in particular can be a treasure of prior states, while the same mechanisms can make simple questions about when a file existed more subtle. This guide sets out for UK lawyers how APFS works, where deleted and earlier data hides within it, how that data is recovered, and how APFS artefacts are interpreted honestly.

⏱ Estimated reading time: 24 min read

§ ABOUT THE AUTHOR

PREPARED BY COMPUTER FORENSICS LAB E-DISCOVERY TEAM

Computer Forensics Lab is a London-based digital forensics and electronic disclosure practice established in 2007. APFS analysis runs through all its Apple work: recovering deleted and earlier data from copy-on-write remnants and, above all, from APFS snapshots, understanding containers, volumes and clones, and interpreting file-system timestamps and structure honestly. The analysis supports the recovery of prior states and deleted material on Macs, iPhones and iPads, and is reported under CPR Part 35 and CrimPR Part 19 with the meaning and limits of APFS artefacts explained.

ESTABLISHED 2007 · LONDON | ISO 17025-ALIGNED PROCEDURES | APFS SNAPSHOT & CLONE ANALYSIS | DELETED-DATA RECOVERY | CPR PART 35 EXPERT REPORTS | FULL CHAIN-OF-CUSTODY DOCUMENTATION

§ CONTENTS

In this guide

§ 0 1 · ORIENTATION

Executive summary

The headline point: APFS, the file system on every modern Mac, iPhone and iPad, uses copy-on-write, snapshots and clones, so deleted and earlier versions of data survive more often, and in different places, than intuition suggests, and snapshots in particular can preserve whole prior states of the device.

How it works (§3): APFS organises storage into containers holding volumes that share space, and uses copy-on-write, changing a file writes new data elsewhere rather than overwriting the old, and clones that share data until modified. **Snapshots (§4):** point-in-time pictures of the entire file system that can preserve deleted files and earlier states, often the richest source of prior data on an Apple device. **Recovery (§5):** deleted and earlier data recovered from copy-on-write remnants, snapshots and container free space, because deletion in APFS frequently leaves recoverable traces. **Interpretation (§6):** APFS records rich timestamps and structure, but their meaning must be understood, since copy-on-write and cloning complicate simple questions about when a file existed, and findings are stated honestly. **Deployment (§7):** recovering prior versions and deleted material, dating changes, and building reliable timelines from a file system that keeps much of its past.

- **A file system that keeps the past:** copy-on-write and snapshots preserve deleted and earlier data.
- **Snapshots are treasure:** point-in-time pictures of the whole file system can hold entire prior states.
- **Deleted often survives:** APFS deletion frequently leaves recoverable remnants and earlier versions.
- **Containers share space:** volumes within a container share free space, shaping where remnants live.
- **Interpret with care:** copy-on-write and clones make timestamps and existence subtler than they seem.

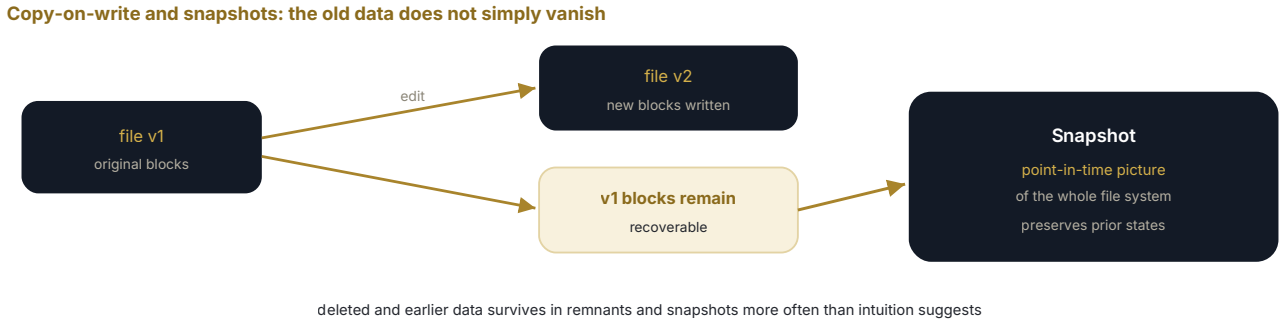

Figure 1 - editing a file writes new blocks and often leaves the old ones recoverable, and snapshots capture entire prior states, so APFS keeps much of its past.

§ 0 2 · FIRST PRINCIPLES

The problem in plain English: a file system that keeps the past

A file system is the scheme by which a device organises its storage, deciding where files live, how they are found, and what happens when they change or are deleted. Older file systems tended to overwrite: edit a file and the new content often replaced the old in place; delete a file and its space was marked free for reuse. Forensic recovery worked in the gaps this left. APFS, which Apple now uses on every Mac, iPhone and iPad, works differently, and the differences matter for evidence. Its defining feature is copy-on-write: when a file changes, APFS does not overwrite the existing data but writes the new version to a fresh location, leaving the old data in place until its space is eventually reused. It layers on snapshots, which freeze a complete picture of the whole file system at a moment in time, and clones, which let copies share the same underlying data until one of them is changed. And it groups storage into containers, within which several volumes share a common pool of free space.

The upshot for the lawyer is counter-intuitive and often favourable. Because APFS tends to write new data rather than overwrite old, and because snapshots capture entire prior states, deleted files and earlier versions of documents survive on Apple devices more frequently, and in more places, than one would expect, snapshots especially can be a genuine treasure, holding a complete earlier picture of a device including material the user thought was gone. At the same time, the same mechanisms make some questions subtler: because a change writes new blocks and cloning shares data, working out exactly when a particular version of a file existed, or what a timestamp truly represents, requires understanding how APFS behaves rather than reading the surface at face value. This guide explains how APFS works, where the recoverable past hides within it, and how its artefacts are recovered and interpreted honestly.

§ 0 3 · HOW APFS WORKS

How APFS works: containers, copy-on-write, clones

- **Containers and volumes:** APFS groups storage into a container that holds one or more volumes (the system, the data, and others) sharing a common pool of free space, so remnants and free space are container-wide, not neatly per-volume: the structure that shapes where recoverable data sits.
- **Copy-on-write:** the defining behaviour, that modifying a file writes the new data to a new location and updates pointers, rather than overwriting the old data in place, so earlier content persists until its space is reused (guide 4's deleted-data principle, amplified): the reason old versions survive.
- **Clones:** space-saving copies that share the same underlying data blocks until one copy is changed, at which point copy-on-write splits them, so a clone is not an independent duplicate and must be understood as such when interpreting copies and dates (§6).
- **Rich metadata:** APFS records extensive metadata, multiple timestamps, extended attributes and structure, providing detailed information about files and their history, valuable for timelines when interpreted correctly (§6): the metadata to be read carefully.
- **Encryption integration:** APFS integrates with the hardware and FileVault encryption of the device (guides 131, 133), so file-system recovery happens on the decrypted, unlocked device, the file-system analysis and the decryption being distinct steps (guide 131): the encrypted context.

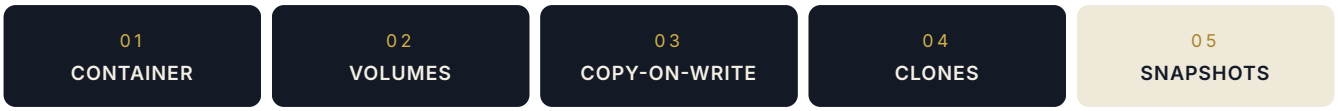

Figure 2 - the APFS structure: a container of shared-space volumes, with copy-on-write, clones and snapshots together determining where deleted and earlier data survives.

§ 0 4 · SNAPSHOTS

Snapshots: the treasure of prior states

- **What a snapshot is:** a point-in-time picture of the entire file system, preserving the state of every file as it was at the moment the snapshot was taken, so a snapshot can hold data since changed or deleted: the single richest source of prior states on an Apple device.
- **Where snapshots come from:** the operating system, Time Machine (guide 131) and other processes create snapshots automatically and on schedule, so a device often holds several, spanning recent history, without the user's awareness: the past preserved as a matter of routine.
- **What they preserve:** files deleted or altered since a snapshot may still exist intact within it, so a document the user deleted, or an earlier version they overwrote, can be recovered whole from a snapshot that predates the change: the deleted made present again.
- **Dating with snapshots:** the existence, and content, of a file at the time of a snapshot is powerful timeline evidence, showing what was on the device at a known past moment (guide 96): the snapshot as a dated checkpoint.
- **Snapshots and preservation:** because snapshots are finite and can be pruned, they are captured as part of a proper acquisition and not allowed to be lost, and their presence is a strong reason to acquire promptly (guide 131, §7): the treasure preserved before it is thinned.

§ 0 5 · RECOVERING DELETED AND EARLIER DATA

Recovering deleted and earlier data

- **From snapshots first:** the most reliable recovery, extracting deleted files and earlier versions intact from the snapshots that predate their change or deletion, whole files rather than fragments (§4): the first and best place to look.
- **From copy-on-write remnants:** earlier versions and deleted content recovered from the old blocks that copy-on-write left in place until reuse, in the container's free space (guide 4): the remnants the write-new behaviour preserves.
- **From container free space:** because volumes share the container's free pool, carving and analysis across the container's unallocated space can recover data from any volume, widening the recovery surface (§3).
- **Understanding clones:** recovered copies interpreted with the clone relationship in mind, so a shared-data clone is not mistaken for an independent duplicate and dates are read correctly (§6): the recovery interpreted, not just extracted.
- **On the decrypted device:** all of this performed on the unlocked, decrypted device (guide 131), since the file system is only readable there, and with integrity preserved and the method documented (guides 2, 3): the recovery grounded in a sound acquisition.

§ 0 6 · INTERPRETATION AND HONEST LIMITS

Timestamps, interpretation and honest limits

- **Rich but subtle timestamps:** APFS records several timestamps per file (created, modified, changed, accessed), which are valuable but must be interpreted in light of copy-on-write and cloning, since a change writes new structure and can affect what a timestamp represents (guide 108): the metadata read with understanding.
- **Existence versus current presence:** a file recovered from a snapshot existed at the snapshot's time; a remnant in free space shows a file once existed; neither is the same as current presence, so the finding states which it is (guide 4's presence-not-intent discipline).
- **Clones complicate copies:** a clone shares data with its source until changed, so two apparent copies may be one underlying object, and their relationship is established before conclusions about duplication or dates are drawn (§3).
- **Copy-on-write and deletion meaning:** the survival of old data is a property of the file system, not necessarily evidence of user intent, so recovered remnants are contextualised, not automatically read as deliberate acts (guide 4).
- **State findings honestly:** APFS conclusions expressed at the level the artefacts support, what a snapshot shows at its time, what a remnant proves once existed, and the limits of timestamp interpretation, rather than overstated certainty (guide 100): the honest reading of a rich but subtle file system.

§ 0 7 · DEPLOYMENT

Deployment: prior versions, deletion and timelines

- **Recovering the deleted document:** a file a party deleted recovered intact from a snapshot or remnant, in fraud, employment and commercial matters, where the deleted material is the evidence (guides 4, 98): the vanished document restored.
- **Proving an earlier version:** an earlier state of a document recovered from a snapshot, exposing later alteration or backdating by comparison with the version relied on (guide 109's backdating discipline): the change proved by the prior version.
- **Dating from snapshots:** the existence and content of material at a snapshot's known time used as a dated checkpoint in a timeline, showing what was on the device when (guide 96): the snapshot anchoring the chronology.
- **Deletion in context:** a deletion established and dated, and its significance assessed honestly, deliberate destruction distinguished from routine change, and read against any duty to preserve (guides 110, 4): the deletion interpreted, not assumed.
- **Corroboration across the estate:** APFS findings corroborated with iCloud, backups and paired-device data (guide 131, §8), so a recovered prior version or deletion is confirmed across the Apple estate, not rested on the file system alone.

§ 0 8 · THE WIDER MAP

Source architecture: where else the evidence lives

Per this series' source-architecture discipline: the APFS file system on one device is one node, and the prior states and deleted data it preserves are often reflected across the Apple estate as well. The device's APFS holds the live data, the copy-on-write remnants and, crucially, the local snapshots. But Time Machine and other backups hold further snapshots and prior states on separate media; iCloud holds synced current and sometimes prior versions and a recently-deleted area; the paired iPhone or iPad holds the same synced files with their own APFS remnants and snapshots; and the account ties the cloud copies together (guide 131). The discipline is to treat the device's snapshots and remnants as one rich source among several, so a prior version or deleted file absent from the live file system may be recovered from a local snapshot, a Time Machine backup, an iCloud version history or a paired device. When the live file is gone, a snapshot holds it; when local snapshots are pruned, a Time Machine backup preserves an older state; and when the device is inaccessible, the iCloud and paired-device copies carry much of the same data.

EVIDENCE	LIVE APFS	LOCAL SNAPSHOTS	TIME MACHINE / BACKUPS	ICLOUD	PAIRED IPHONE / IPAD	DELETED / RECOVERABLE
Current files	Y	Y: at snapshot times	Y: prior states	Y: synced	Y: synced	n/a
Earlier versions	COW remnants	Y: whole, at time	Y: backup versions	Version history	Device remnants	Y (snapshot/ remnant)
Deleted files	Free-space remnants	Y: if pre-deletion	Y: in backups	Recently deleted	Device remnants	Y (multiple routes)
Timestamps	Y: rich metadata	Y: snapshot time	Backup times	Sync times	Device metadata	n/a
File existed at time T	If still present	Y: snapshot proves	Y: backup proves	Cloud version	Paired copy	Snapshot is decisive

Fallback strategy in one line: look to the snapshots and remnants first, then across the estate; a file gone from the live system is often held whole in a local snapshot, a Time Machine backup or an iCloud version.

§ 0 9 · IN THE WILD

Worked examples

EXAMPLE 1 · THE DELETED DOCUMENT A SNAPSHOT KEPT WHOLE

A key document a party swore had never existed had, it was alleged, been created and then deleted from their Mac. The live file system showed no trace, but the §4 snapshot analysis found it: an automatic APFS snapshot taken before the deletion held the document intact, whole, readable, with its metadata, not a carved fragment. The §7 dating followed: because the snapshot's time was known, the document was shown to have existed on the device at that moment, and a later snapshot showed it gone, bracketing the deletion. The recovery was corroborated against an iCloud version (§8). What the party believed they had erased had been preserved, in full, by a file-system feature they did not know was running. The lesson is §4's: APFS snapshots are the richest source of prior states on an Apple device, routinely preserving deleted files and earlier versions whole, so a document gone from the live file system is very often recoverable, and datable, from the snapshots the system quietly keeps.

EXAMPLE 2 · THE EARLIER VERSION THAT EXPOSED THE EDIT

A contract produced in a dispute was challenged as having been altered after signing. The §5 recovery from copy-on-write remnants and an earlier snapshot produced a prior version of the very file: comparison showed that a clause had been changed after the date the document purported to bear, a §7 backdating and alteration finding (guide 109). The §6 interpretation was careful: the timestamps were read with copy-on-write behaviour in mind, and the clone and version relationships established, so the comparison rested on a sound understanding of what each recovered object was. The earlier version, recovered from the file system's own retained past, proved the later edit. The lesson is §5's: because APFS writes new data rather than overwriting old, earlier versions of altered documents frequently survive in remnants and snapshots, and recovering the prior version, interpreted correctly, can expose an alteration the current file conceals.

EXAMPLE 3 · THE CLONE THAT WAS NOT TWO DOCUMENTS

An analysis initially appeared to show two separate copies of a sensitive file, suggesting duplication and distribution. The §6 interpretation discipline corrected the picture: the two were an APFS clone, sharing the same underlying data until one was changed, so they were, at the relevant time, effectively one object with two names, not two independent copies made and moved. Reading the clone relationship correctly changed the conclusion materially, from apparent duplication to a single cloned file, and the §6 honest finding said so. Had the clone been mistaken for two independent duplicates, the report would have overstated the facts. The lesson is §6's: APFS features like clones make simple appearances misleading, and interpreting the file system correctly, understanding what a clone, a remnant or a timestamp actually represents, is as important as recovering the data, because an accurate account depends on reading APFS for what it truly does.

§ 10 · WHERE IT GOES WRONG

Common mistakes and technical limitations

Common mistakes

- **Ignoring snapshots:** the biggest missed opportunity, overlooking the APFS snapshots that hold whole prior states and deleted files (Example 1, §4).
- **Assuming deleted means gone:** treating an APFS deletion as final when copy-on-write and snapshots so often preserve the data (§5).
- **Misreading clones as duplicates:** a clone mistaken for two independent copies, overstating duplication (Example 3, §6).
- **Reading timestamps naively:** APFS timestamps interpreted without regard to copy-on-write, drawing wrong conclusions about when a file existed (§6).
- **Losing snapshots by delay:** snapshots pruned before acquisition because the device was not preserved promptly (§4, guide 131).
- **Recovering without interpreting:** data extracted but its clone, remnant or snapshot context not established, so its meaning is misstated (§6).
- **Forgetting the container:** free-space analysis confined to one volume when the container's shared pool holds relevant remnants (§3).
- **Not corroborating:** an APFS finding rested on the file system alone when the estate could confirm it (§8).

Technical limitations

- **Reuse eventually overwrites:** copy-on-write remnants persist only until their space is reused, so recovery is not guaranteed and time matters (§5).
- **Snapshots are finite:** they are pruned over time, so prompt preservation is needed to keep them (§4).
- **Timestamps need expertise:** APFS metadata is rich but subtle and easily misread (§6).
- **Recovery needs the decrypted device:** file-system analysis happens on the unlocked machine, so acquisition constraints apply (guide 131).
- **APFS evolves:** Apple changes APFS across OS versions, so behaviour is checked per device and version (§3).

§ 11 · INTERROGATORIES & DRAFTING AIDS

Questions to ask · Suggested wording

Ask your client

- Is the relevant data on an Apple device, and might a deleted file or earlier version be recoverable from APFS snapshots or remnants?
- Has the device been preserved promptly, so snapshots have not been pruned and remnants not overwritten?
- What earlier version or deleted document matters, so recovery and dating can be targeted?

Ask your opponent

- Please preserve the Apple device and its APFS snapshots without pruning or continued use that may overwrite remnants, and disclose any prior versions or deleted material recoverable from snapshots.
- For any document relied on, please confirm whether earlier versions exist in snapshots, Time Machine or iCloud version history.
- Please confirm whether relevant files have been deleted or altered, and when.

Ask your eDiscovery / forensic provider

- What do the APFS snapshots hold, and can the deleted or earlier data be recovered whole from them?
- How are the timestamps and any clones to be interpreted, and what do they honestly show?
- Does the estate (backups, iCloud, paired devices) corroborate the APFS findings?

SUGGESTED WORDING · INSTRUCTION FOR AN APFS ANALYSIS

> "We instruct you to examine the Apple File System (APFS) on the device at Schedule 1 to recover and interpret deleted and earlier data. Please, on the acquired, decrypted device and preserving integrity: (1) identify and preserve all APFS snapshots before any can be pruned, and recover from them, intact, any deleted files or earlier versions relevant to Schedule 2; (2) recover earlier and deleted data from copy-on-write remnants and from the container's shared free space; (3) interpret the recovered material correctly, establishing clone relationships and reading APFS timestamps in light of copy-on-write, distinguishing existence at a snapshot's time and past existence shown by a remnant from current presence; (4) date relevant material using snapshots as known checkpoints, and assess any deletion or alteration in context and against any duty to preserve; and (5) corroborate the findings against Time Machine, iCloud version history and paired-device data, and state conclusions at the level the artefacts honestly support, explaining the limits of timestamp interpretation."

§ 12 · QUICK CONTROL

Checklist and red flags · When to involve a digital forensic expert

The APFS checklist

- ☐ Device preserved promptly; snapshots not pruned, remnants not overwritten
- ☐ All APFS snapshots identified and preserved
- ☐ Deleted files and earlier versions recovered whole from snapshots
- ☐ Copy-on-write remnants and container free space analysed
- ☐ Clone relationships established before conclusions on copies
- ☐ Timestamps interpreted with copy-on-write in mind
- ☐ Existence-at-time distinguished from current presence
- ☐ Deletion and alteration assessed in context and dated
- ☐ Findings corroborated across the Apple estate
- ☐ Conclusions stated at honest confidence with limits explained

Red flags

- Snapshots overlooked when deleted or earlier data is sought: Example 1.
- An APFS deletion assumed final without checking snapshots and remnants.
- A clone reported as two independent duplicates: Example 3.
- Timestamps read naively, ignoring copy-on-write: §6.
- Delay allowing snapshots to be pruned.
- Data recovered but its snapshot or clone context not established.
- File-system findings not corroborated across the estate.

When to involve a digital forensic expert

Whenever deleted or earlier data on an Apple device matters, because APFS holds far more of the past than intuition suggests and reading it correctly is a specialist task: the expert preserves and mines the snapshots that so often hold deleted files and earlier versions whole (Example 1, §4), recovers from copy-on-write remnants and the shared container free space (§5), and, just as importantly, interprets what is found, establishing clone relationships and reading timestamps in light of copy-on-write so that appearances are not misstated (Examples 2 and 3, §6). The findings are dated using snapshots as checkpoints, assessed in context, and corroborated across the Time Machine, iCloud and paired-device estate (§7, §8), and reported at honest confidence under CPR Part 35 or CrimPR Part 19. Preservation is time-sensitive, since snapshots are pruned and remnants overwritten. Through **cflab.uk** and **e-discovery.uk**, APFS work turns a file system that quietly keeps its past into recovered, correctly-interpreted evidence: the deleted document restored whole, the earlier version that exposes an edit, and the clone read for what it truly is.

§ 13 · COMMON QUESTIONS

Frequently asked questions

If a file was deleted from an Apple device, is it recoverable?

Often, and more reliably than on older systems (§4, §5, Example 1). APFS uses copy-on-write, so changes write new data and leave old data recoverable, and it takes snapshots, point-in-time pictures of the whole file system, that frequently hold deleted files and earlier versions intact. A document gone from the live file system is very often recovered whole from a snapshot that predates its deletion, or from remnants in free space, provided the device is preserved before the data is overwritten or the snapshots pruned.

What is an APFS snapshot, and why does it matter?

A snapshot is a frozen picture of the entire file system at a moment in time (§4). The operating system and Time Machine create them automatically, so a device usually holds several without the user knowing. They matter because they can preserve files as they were at a past moment, including material since deleted or altered, making them the richest source of prior states on an Apple device and a powerful way to prove what existed on the device, and when.

Can we prove a document was altered after it was signed?

Frequently, using APFS (§5, §7, Example 2). Because APFS retains earlier data, a prior version of the document can often be recovered from a snapshot or copy-on-write remnants, and compared with the version relied on. If the comparison shows a change made after the purported date, that exposes the alteration or backdating (guide 109). The interpretation is done carefully, reading timestamps and clone relationships correctly, so the comparison rests on a sound understanding of each recovered version.

Why does it matter whether something is a clone?

Because a clone is not an independent copy (§3, §6, Example 3). APFS clones share the same underlying data until one is changed, so two apparent copies may really be one object with two names, not two separate duplicates that were made and moved. Mistaking a clone for independent duplication overstates the facts. Establishing the clone relationship before drawing conclusions about copying or distribution is essential to an accurate account, which is why interpretation matters as much as recovery.

How urgent is preserving an Apple device for APFS recovery?

Quite urgent (§4, §5). Copy-on-write remnants survive only until their space is reused, and snapshots are pruned over time, so continued use of the device steadily erodes what can be recovered. Preserving the device promptly, and acquiring it while snapshots and remnants remain, maximises recovery. As with the Mac acquisition itself (guide 131), the device should be preserved without careless use, and the sooner it is examined, the more of its past survives.

Do APFS timestamps tell us exactly when things happened?

They are valuable but must be read with care (§6). APFS records several timestamps per file, which help build timelines, but copy-on-write and cloning mean a timestamp may not mean what it appears to at first glance, since a change writes new structure. So the metadata is interpreted by someone who understands APFS behaviour, and findings are stated at honest confidence, using snapshots as reliable dated checkpoints where possible, rather than reading raw timestamps at face value.

§ 14 · REFERENCE

Glossary

APFS
The Apple File System, used on modern Macs, iPhones and iPads.

Container
An APFS storage pool holding volumes that share free space.

Volume
A file system within a container (system, data and others).

Copy-on-write
Writing changed data to a new location, leaving old data.

Snapshot
A point-in-time picture of the whole file system.

Clone
A copy sharing data with its source until changed.

Remnant
Old data left recoverable by copy-on-write until reused.

Free space
The container's shared unallocated pool where remnants sit.

Prior version
An earlier state of a file, recoverable from snapshots.

Metadata
APFS's rich per-file records, including timestamps.

Sources and authoritative references

The APFS disciplines in this guide draw on materials published by our laboratory and e-discovery practice, referenced here as sources:

- cflab.uk, digital forensics services (APFS snapshot and remnant recovery, deleted-data recovery, CPR Part 35 and CrimPR Part 19 reporting): cflab.uk/digital-forensics-services · guides library: cflab.uk/guides
- e-discovery.uk, EDRM Phase 04 · Collection and Phase 06 · Analysis (file-system recovery and interpretation as practised): e-discovery.uk/edrm/collection · e-discovery.uk/edrm/analysis
- e-discovery.uk, practice method and Knowledge Centre: e-discovery.uk · e-discovery.uk/knowledge
- Apple File System reference and Apple Platform Security guide (APFS structure, snapshots, cloning and encryption): support.apple.com, Platform Security · developer.apple.com, file system
- Forensic Science Regulator, Code of Practice v2 (recovery and interpretation): gov.uk, FSR Code · ISO/IEC 27037: iso.org, 27037
- PD 57AD and CPR Part 31 · CPR Part 35: justice.gov.uk, PD 57AD · justice.gov.uk, Part 31 · justice.gov.uk, Part 35

§ HOW A SPECIALIST LABORATORY CAN ASSIST

Working with Computer Forensics Lab

APFS work at the laboratory begins from the recognition that Apple devices keep much of their past. On the acquired, decrypted device (guide 131), the APFS snapshots are identified and preserved first, before any can be pruned, because they are the richest source of prior states and routinely hold deleted files and earlier versions whole (Example 1). Deleted and earlier data is recovered from those snapshots, from copy-on-write remnants and from the container's shared free space, and, crucially, it is interpreted correctly: clone relationships are established so shared-data copies are not mistaken for independent duplicates, and timestamps are read in light of copy-on-write, so appearances are not misstated (Examples 2 and 3). Snapshots serve as known dated checkpoints for the timeline, deletions and alterations are assessed in context and against any duty to preserve, and the findings are corroborated across the Time Machine, iCloud and paired-device estate. Everything is reported at honest confidence, with the limits of timestamp interpretation explained, under CPR Part 35 or CrimPR Part 19. Because remnants are overwritten and snapshots pruned with use, prompt preservation matters. Conflicts checks, confidential scoping discussions and fixed-fee estimates are routine; and the recurring message is an encouraging one: on an Apple device, the deleted and the earlier are, more often than clients expect, still there to be recovered and correctly read.

INSTRUCT THE LAB

Speak to a forensic examiner, not a salesperson.

Describe your matter in confidence and an examiner will respond the same working day. Conflicts checks and scoping calls with US counsel are routine.

NEW ENQUIRIES
+44 (0)20 7164 6971

EMAIL
info@cflab.uk

E-DISCOVERY
e-discovery.uk

Computer Forensics Lab Ltd · Euro House, 133 Ballards Lane, Finchley, London N3 1LJ, United Kingdom
cflab.uk · Customer service +44 (0)20 7164 6915 · ICO ZB395023 · NPCC · ISO 17025 aligned practice · UK Gov Digital Marketplace