

Version notice. Artifacts change between versions of iOS, iPadOS, macOS and watchOS. A term, process, subsystem or message that works on one build can change, move or disappear on another. This sheet reports what was measured on the builds named at the bottom of the last page. It is not a guarantee for any other build. Verify every term on your own reference device before you rely on it in a case.

How to use this sheet

This sheet is the companion to the search term and process cheatsheet. That sheet says what each term is, which process writes it and where it misleads. This sheet says which terms to run together to answer one question. Run every term of a set inside the same marker window. Write down each result, including zero hits, and name the level at which you counted. A set is not a proof. Each term shows that one process wrote one string. When terms from different processes agree, they support a reconstruction. When they disagree, you have found something to explain.

Unlock and authentication

Question	Run these in one marker window	What the set can support	What it cannot show
Unlock, iPhone: did it happen?	<code>kernel, AppleSEPKeyStore, apfs is now UN-locked</code> <code>softwareupdateservicesd Device unlocked</code>	<i>That the device was unlocked. Two kernel lines and one Device unlocked line per unlock on iOS.</i>	<i>The method. Device unlocked says nothing about it. success=YES is not evidence of passcode entry.</i>
Unlock, iPhone: which method?	<code>biometrickitd, kernel inBioUnlock matchResult MATCH</code> <code>Face ID hardware isFaceDetected=1 getFaceDetectInfo </code> <code>PearlCamFrameReceived</code> <code>Touch ID hardware kAppleBiometricFingerOnEvent </code> <code>kAppleBiometricFingerOffEvent</code> <code>Passcode, SpringBoard SBUIPasscodeTextField Suppress Scanning State</code>	<i>A biometric method when the biometric terms fall in the same window as the endpoint. matchResult contains the user ID and enrollment UUID. The passcode field on screen is a lead for a passcode entry (not measured on iOS 18 and older).</i>	<i>A Face ID frame on its own: FD Distance 0 means no detection, and the line alone is not a match. Transition: is absent on a biometric unlock.</i>
Unlock, iPhone: attempt rejected	<code>biometrickitd matchResultHandler: MATCH -1 matchResult:timestamp: NO-MATCH </code> <code>has received no-match</code> <code>same window the method terms of the previous row</code>	<i>That an attempt was rejected.</i>	<i>Which method was rejected. These lines fire on a wrong passcode and on a failed Face ID. The cluster determines which. matchResultHandler: iOS 26.6.1. The other two: macOS 26.6.2.</i>
Unlock, macOS	<code>kernel is now UN-locked</code> <code>loginwindow, LWScreenLockAuthentication Unlock succeeded, with password</code>	<i>That the Mac was unlocked and that a password was used. The endpoint gives one line per unlock on macOS.</i>	<i>Other methods: only the password message is in this set. Search the message, not the function name _authSuccessUsingPassword.</i>

Touch

Question	Run these in one marker window	What the set can support	What it cannot show
Touch while the screen is off	<code>backboardd, BackBoard:TouchEvents contact 0 presence: touching</code> <code>backboardd, SpringBoard, Backlight received tapToWake send tapToWake </code> <code>tapToWakeDidRecognize</code> <code>backboardd, Multitouch Dispatching</code>	<i>Physical contact with the screen, before any unlock took place.</i>	<i>Position, gesture type or repetition. A double tap appears as two touches. Never search on tapToWake alone.</i>
Swipe from the edge	<code>backboardd, SpringBoard system gesture stealing the touches </code> <code>SBGrabberTongue</code>	<i>A swipe from the edge when nine to twenty-three lines fall in the window.</i>	<i>A swipe from a single line. One line is an edge touch.</i>

Question	Run these in one marker window	What the set can support	What it cannot show
Typing in an app	<code>UIKit:KeyboardTouch</code> , <code>KeyboardSignposts</code> touch down touch drag touch up <code>appTouchDown</code>	<i>That the keyboard was used. The process name names the app.</i>	<i>The typed text.</i>

Buttons, movement and cable

Question	Run these in one marker window	What the set can support	What it cannot show
A physical button press	<code>backboardd</code> <code>com.apple.BackBoard:Button</code> <code>VolumeIncrement</code> <code>VolumeDecrement</code> <code>Lock</code> <code>finished: firstDown:</code> <code>SpringBoard, buttons:Lock</code> press count: <code>SpringBoard, buttons:Volume</code> <code>RECEIVED rawVolumeIncreasePress</code> <code>Lock</code> button single press recognized <code>SpringBoard, buttons:Combo</code> SOS button gesture: press type=	<i>Which button, how long it was held and how many times it was pressed. In a combination: 102 volume up, 103 volume down, 104 side button.</i>	<i>An emergency call. <code>SBSOSClawGestureObserver</code> appears on every press, and <code>SS disabled: YES</code>, reason: <code>SOSTriggered</code> also appears when the power-off menu is opened. Zero hits for <code>HIDEvent Power</code>, <code>HIDEvent Volume</code> or <code>homeButton</code> are not an absence.</i>
Device picked up or rotated	<code>SpringBoard, audiomxd, biometrickitd</code> <code>Gesture notification: 1(Detected)</code> <code>Start:FaceUp</code> <code>backboardd</code> Received orientation. <code>kernel(AOP) [TTW]</code> Orientation changed join key Timestamp	<i>That the device was moved. The end state is reliable. The same Timestamp value means one event.</i>	<i>Who moved it, or the start state. Three processes write the pick-up line, so count at the kernel line.</i>
Cable connected	<code>kernel, IOAccessoryManager</code> <code>AppleUSBCableDetect 1</code> <code>AppleUSBCableDetect 0</code> <code>USB Power (VBUS) Present</code> <code>kernel, UserEventAgent</code> <code>AppleUSBCableType</code> <code>USBHost</code> <code>connectType: 2</code> <code>lockdownd</code> <code>usb_host_connected_block_invoke</code> <code>bump_connection_count</code> <code>powerd</code> <code>handleChargingStateUpdate</code> <code>State:Charging</code> <code>State:Disconnected</code>	<i>That a cable was connected, whether to a computer (<code>connectType 2</code>) or a power source (<code>connectType 1</code>), and that a session with a host existed.</i>	<i>One count. Four processes give four counts for the same cable. Name the level you counted at.</i>

Apps and Apple Watch

Question	Run these in one marker window	What the set can support	What it cannot show
App opened from the home screen	<code>SpringBoard</code> <code>Allowing tap for icon view</code> <code>Launching application</code> <code>SBIconLocationDock</code> <code>SBIconLocationRoot</code> masked lines <code>Icon tapped</code> <code>Icon touch began</code> <code>macOS LAUNCH: AppSandbox</code> request successful	<i>That an app was opened and which one. The bundle ID is in <code>Allowing tap for icon view</code> and in <code>Launching application</code>. Count on <code>Launching application</code>.</i>	<i>The app, from a masked line: it gives the moment, not the app. Prewarming is not a launch.</i>
Apple Watch input	<code>Carousel, CarouselPlugins</code> <code>Description: Crown down</code> <code>Description: Button held</code> <code>Description: Button long-held</code>	<i>Crown presses and side-button presses. Short and long presses are separate lines.</i>	<i>Anything without a <code>co-sysdiagnose</code>: acquisition is via <code>co-sysdiagnose</code>.</i>

Using the sets together

The sets answer different questions. An unlock, the touch that came before it and the button press before that each have their own set. Combine the results of several sets in one timeline, and keep the window of each set visible.

No single term proves an event. Taken together, hits from different processes, counted at a named level and checked against your own reference device, can support a sound reconstruction. This sheet and the search term and process cheatsheet are meant to be used together.

Disclaimer

The search terms and notes in this sheet come from the author's own reference devices. No case data was used. Apple changes what it logs between versions of iOS, iPadOS, macOS and watchOS, including in minor updates and betas. A term can change, move to another process or subsystem, or stop being written at all. A message that is absent on one version can appear on the next.

This sheet is a starting point for your own reference research. It does not replace validation on the build you are examining, and it does not say what a line means in a specific case. Record the device, the build and the acquisition method for every search you run. The author accepts no liability for conclusions drawn from this sheet. It is a personal research publication and does not represent the position of any employer or working group.

Contribute an artifact

Is an artifact missing, or does a term no longer work on a newer version? Send it to tim@thesisfriday.com. Include the device model, the OS version and build, the exact action you performed with the time and the clock you read it from, the raw log lines, and how you acquired the log. Do not send case data. Contributions are checked against a reference action before they are added. Accepted contributions are credited by name unless you ask otherwise.

Copyright and use

© 2026 Tim Korver. This sheet is licensed under CC BY-NC-ND 4.0 (creativecommons.org/licenses/by-nc-nd/4.0). You may share it unchanged and for non-commercial purposes, with attribution.

Attribution is required for every public use, including presentations, courses, articles, tool documentation and repositories. Name the author (Tim Korver), the source (thesisfriday.com) and the license.

Do not place this sheet, or content copied from it, behind a paywall, a login or a paid course, and do not sell it. Commercial use requires written permission via tim@thesisfriday.com.

Do not modify the sheet and redistribute the changed version. Send proposed changes to the address above.