

A Deep Dive Into Apple OS Updates via MDM

(now with more DDM!)

Hello!



Bryce Carlson
Product Manager

99.9%

Of your devices are in end user hands



100%

Of your devices are a threat vector



System Updates via MDM



System Updates via MDM



Agenda

- Updates via MDM (what, how, why)
- MDM Protocol for Updates
 - Commands and Logs
- MDM OS Updates user experience
- Updates via DDM improvements
 - Why are they needed
- MDM Watchdog
- Q/A

System Updates via MDM

What

System Updates installed on devices via MDM

How

Device Channel MDM commands sent to the device

Why?

macOS 11+ and more in macOS 12+



Requirements

Device is Supervised

macOS 12 and newer

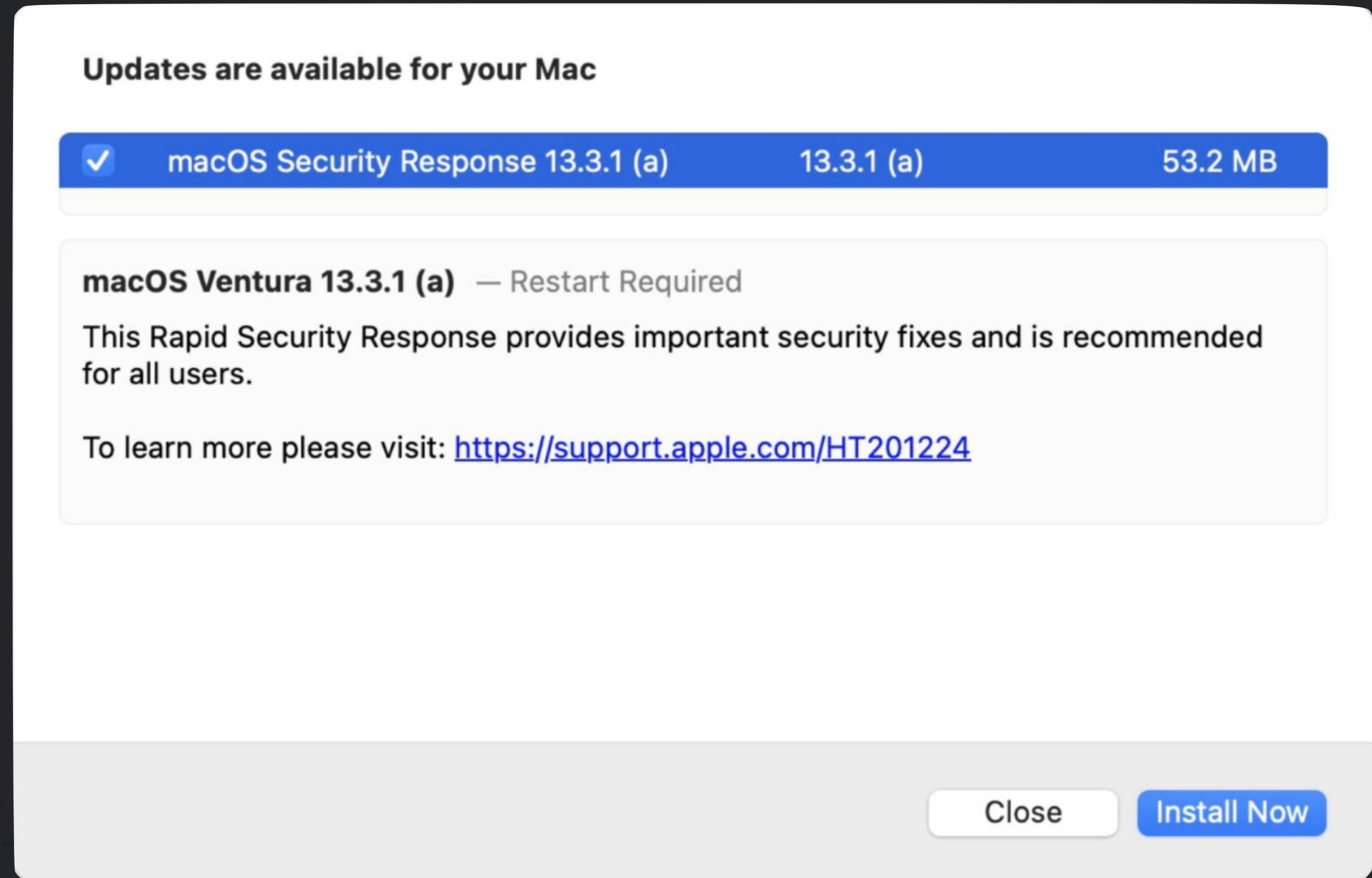
ADE or Reduced Security Mode

iOS 9 and newer

iPadOS 13 and newer

tvOS 12 and newer

System Updates via MDM



System Updates via MDM

System Updates Status

Some devices may need to be restarted to clear the current Update Status.

Update Status Available Updates History

Update available: XProtectPlistConfigData
Version: 2167
Last update: May 2, 2023 12:30:23PM EDT

✓ Update command sent & received

Installation started

Downloaded

Installing

Update available: XProtectPayloads
Version: 95
Last update: May 2, 2023 12:30:23PM EDT

✓ Update command sent & received

Installation started

Downloaded

Installing

Update available: macOS Security Response 13.3.1 (a)
Version: 13.3.1
Last update: May 2, 2023 12:38:33PM EDT

✓ Update command sent & received

Installation started

Downloaded

Installing

System Updates Status

Device: iPhone Supervised [Refresh](#)

Current OS: 16.4.1

Some devices may need to be restarted to clear the current Update Status.

Update Status Available Updates History

Update available: iOS Security Response 16.4.1 (a)
Version: 16.4.1
Last update: May 2, 2023 11:34:56AM CDT

✓ Update command sent & received

Installation started

Downloaded

Installing

System Updates via MDM

- Keep updated or set max version
- Re-send command
- Enforce when available, or allow user to defer
- Schedule updates to avoid user interruptions

Enable macOS Updates

Set maximum version

Keep devices updated to the latest OS (including major versions)

Re-send update command if last status is older than:

The default time for this property is set to 24 hours.

Options

Download and install depending on current device state.

Enable iOS Updates

Set maximum version

Keep devices updated to the latest OS (including major versions)

Re-send update command if last status is older than:

The default time for this property is set to 24 hours.

Enable iPadOS Updates

Set maximum version

Keep devices updated to the latest OS (including major versions)

Re-send update command if last status is older than:

The default time for this property is set to 24 hours.

Enable tvOS Updates

Set maximum version

Keep devices updated to the latest OS (including major versions)

Re-send update command if last status is older than:

The default time for this property is set to 24 hours.

Schedule Updates

Set the days and times when update commands are sent. Start time is based on each device's current time. The default schedule is nightly at 2am UTC.

Days Allowed

Sun Mon Tue Wed Thu Fri Sat

Start Time : **Time Window**

Stop Sending Commands prior to the time window close.

InstallAction Options

- *Default*
 - Download or install the update or upgrade, depending on the current state.
 - End user will get 60 second count down in Notification Center if a reboot is needed
- *InstallForceRestart*
 - Perform the default action, and then force a restart if the update requires it.
- *InstallLater* (this option supports end user deferrals)
 - Download the software update or upgrade and install it at a later time.
 - With **Deferrals allowed** set, the system will prompt the user once a day, up to the maximum amount of times, before showing the reboot pending (in Notification Center just like *Default* option) and having the device to continue with the minor update.
 - If "Allow user to defer minor updates" is not selected, the user will be able to infinitely defer updates.

MDM Update Changes (InstallActions)

macOS 13 Ventura

• macOS 13.0 Ventura RC2

- The `ScheduleOSUpdate` command now supports a new `Priority` key for minor updates. Administrators can set a value of High to prioritize downloading and preparing the update. The default is Low.
- A Mac will now acknowledge and respond to the `ScheduleOSUpdate`, `OSUpdateStatus`, and `AvailableOSUpdatecommands` when the device is asleep or in PowerNap mode instead of responding with `NotNow`.
- Resolves an issue where a software update initiated by MDM using `InstallLater` does not install overnight.

• macOS 13.1 Ventura RC

- If a user dismisses the countdown timer from an `InstallForceRestart` command, the Mac will proceed with the install and restart.

• macOS 13.3 Ventura RC3

- Resolved an issue where the user deferral count for `InstallLater` software updates reset after a restart.

• macOS 13.4 Ventura RC3

- Improves reliability when using MDM `InstallLater` action.

• macOS 13.5 Ventura RC3

- Setting the `restrict-software-update-require-admin-to-install` key to true in a `SoftwareUpdate` payload no longer prevents MDM from installing updates.

macOS 14 Sonoma

• macOS 14 Sonoma Beta5

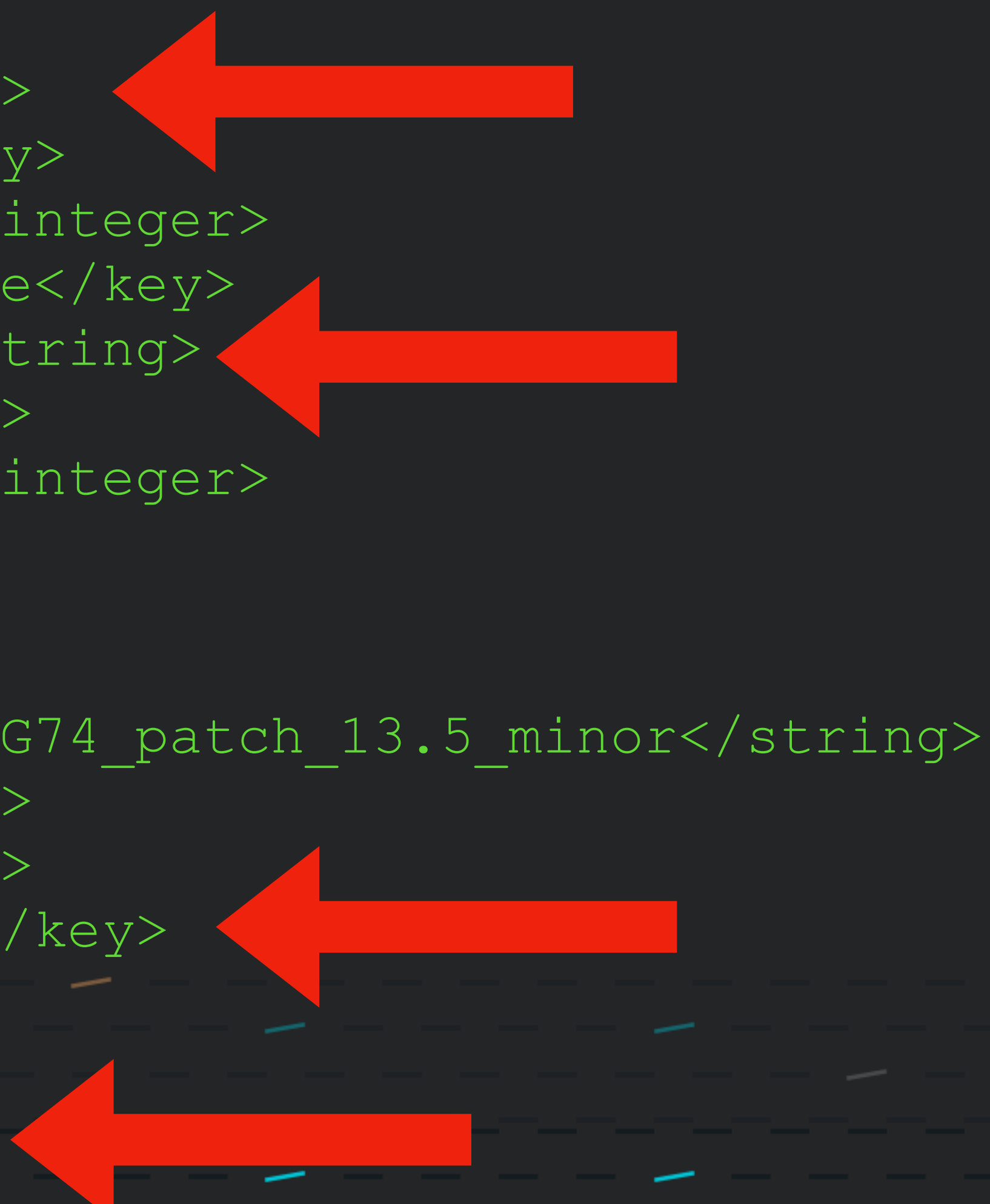
- Starting with macOS 14, `AvailableOSUpdates` will only return OTA updates as they are able to update or upgrade the OS and support all `InstallAction` options. Full replacement installers are no longer returned.

AvailableOSUpdate (response)

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>AvailableOSUpdates</key>
  <array>
    <dict>
      <key>AllowsInstallLater</key>
      <false/>
      <key>Build</key>
      <string>22G74</string>
      <key>DownloadSize</key>
      <integer>1556959678</integer>
      <key>HumanReadableName</key>
      <string>macOS 13.5</string>
      <key>InstallSize</key>
      <integer>1915138048</integer>
      <key>IsCritical</key>
      <false/>
      <key>ProductKey</key>
      <string>MSU_UPDATE_22G74_patch_13.5_minor</string>
      <key>ProductName</key>
      <string>macOS</string>
      <key>RestartRequired</key>
      <true/>
      <key>Version</key>
      <string>13.5</string>
    </dict>
  </array>
  <key>CommandUUID</key>
  <string>0001_AvailableOSUpdates</string>
  <key>Status</key>
  <string>Acknowledged</string>
  <key>UDID</key>
  <string>00008020-000915083C80012E</string>
</dict>
</plist>
```

AvailableOSUpdate (response)

```
<key>AvailableOSUpdates</key>
  <array>
    <dict>
      <key>AllowsInstallLater</key>
      <false/>
      <key>Build</key>
      <string>22G74</string>
      <key>DownloadSize</key>
      <integer>1556959678</integer>
      <key>HumanReadableName</key>
      <string>macOS 13.5</string>
      <key>InstallSize</key>
      <integer>1915138048</integer>
      <key>IsCritical</key>
      <false/>
      <key>ProductKey</key>
      <string>MSU_UPDATE_22G74_patch_13.5_minor</string>
      <key>ProductName</key>
      <string>macOS</string>
      <key>RestartRequired</key>
      <true/>
      <key>Version</key>
      <string>13.5</string>
    </dict>
```


A diagram consisting of five thick red arrows pointing left towards specific XML elements in the code block above. The arrows point to the following elements: 1. The string value '22G74' under the 'Build' key. 2. The string value 'macOS 13.5' under the 'HumanReadableName' key. 3. The string value 'MSU_UPDATE_22G74_patch_13.5_minor' under the 'ProductKey' key. 4. The boolean value 'true' under the 'RestartRequired' key. 5. The string value '13.5' under the 'Version' key.

ScheduleOSUpdate (send)

```
<?xml version="1.0" encoding="UTF-8"?>
  <!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
  <plist version="1.0">
    <dict>
      <key>Command</key>
      <dict>
        <key>RequestType</key>
        <string>ScheduleOSUpdate</string>
        <key>Updates</key>
        <array>
          <dict>
            <key>InstallAction</key>
            <string>Default</string>
            <key>ProductKey</key>
            <string>MSU_UPDATE_22G74_patch_13.5_minor</string>
            <key>ProductVersion</key>
            <string>13.5</string>
          </dict>
        </array>
      </dict>
      <key>CommandUUID</key>
      <string>0001_ScheduleOSUpdate</string>
    </dict>
  </plist>
```

ScheduleOSUpdate (send)

```
<key>InstallAction</key>  
<string>Default</string>  
<key>ProductKey</key>  
<string>MSU_UPDATE_22G74_patch_13.5_minor</string>  
<key>ProductVersion</key>  
<string>13.5</string>
```

Two red arrows are present. One arrow points horizontally from the right towards the <string>Default</string> element. The second arrow points diagonally from the bottom right towards the <string>13.5</string> element.

What does that actually look like?

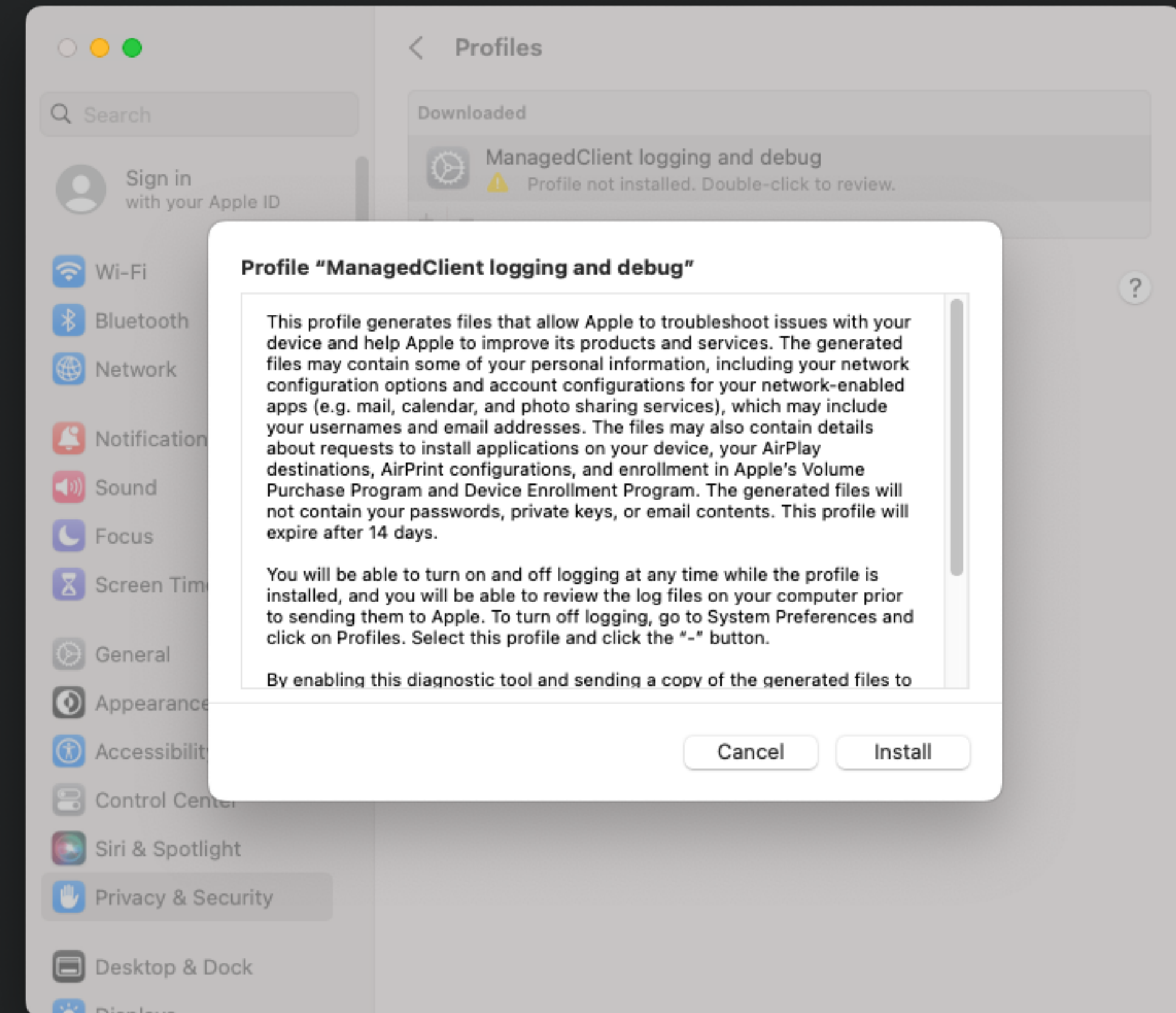
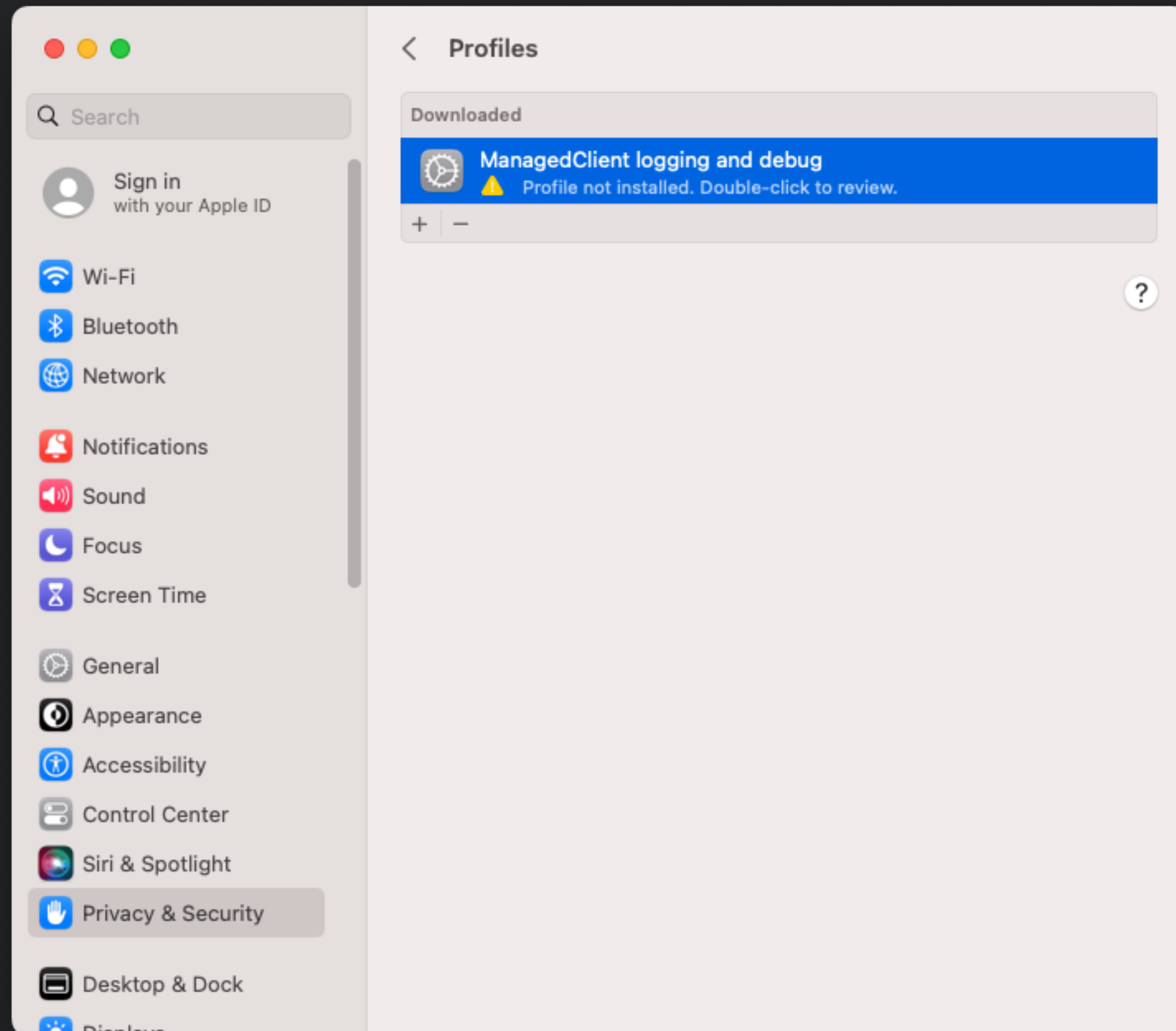


Debug Logging Profile for MDM

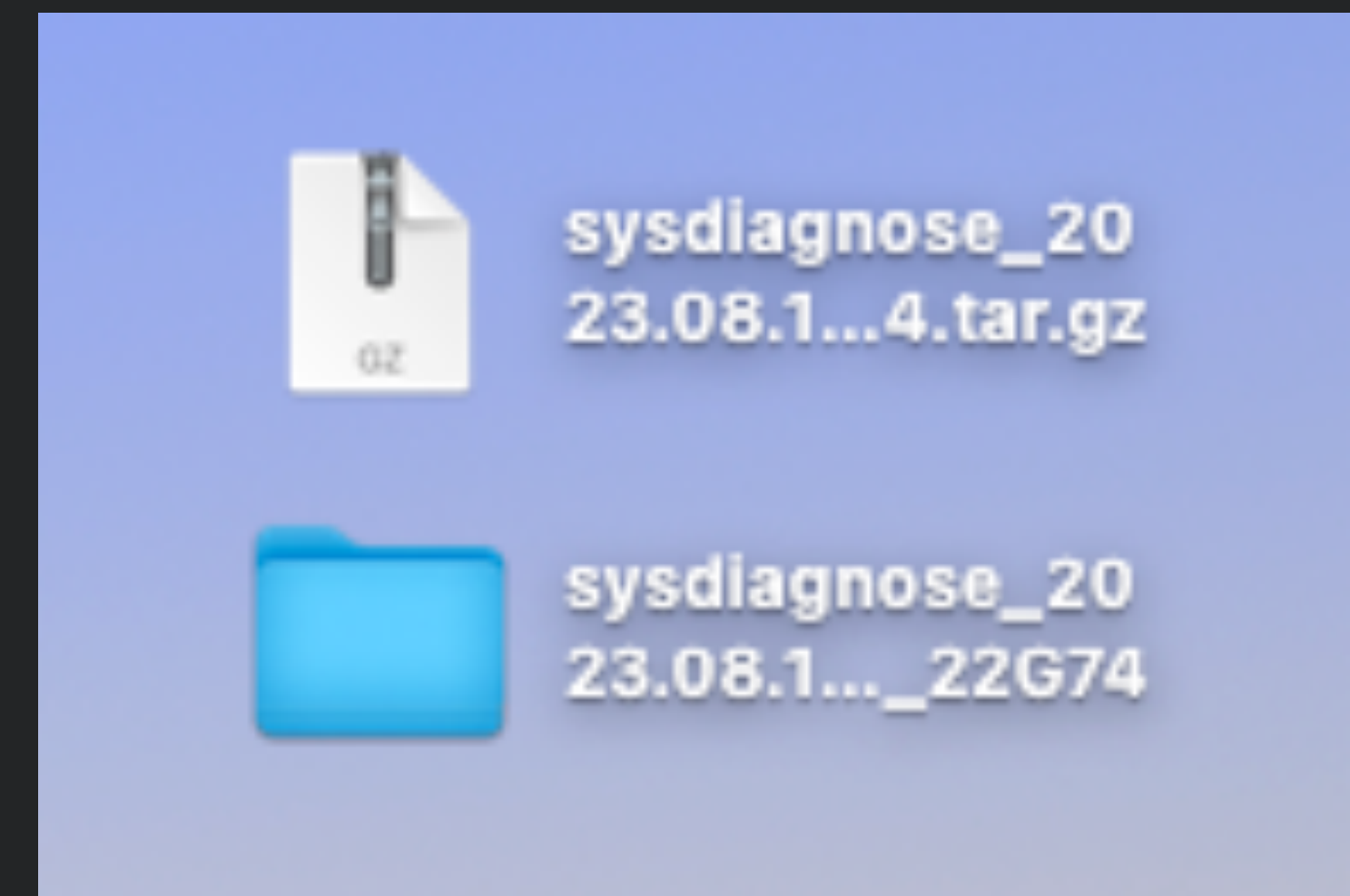
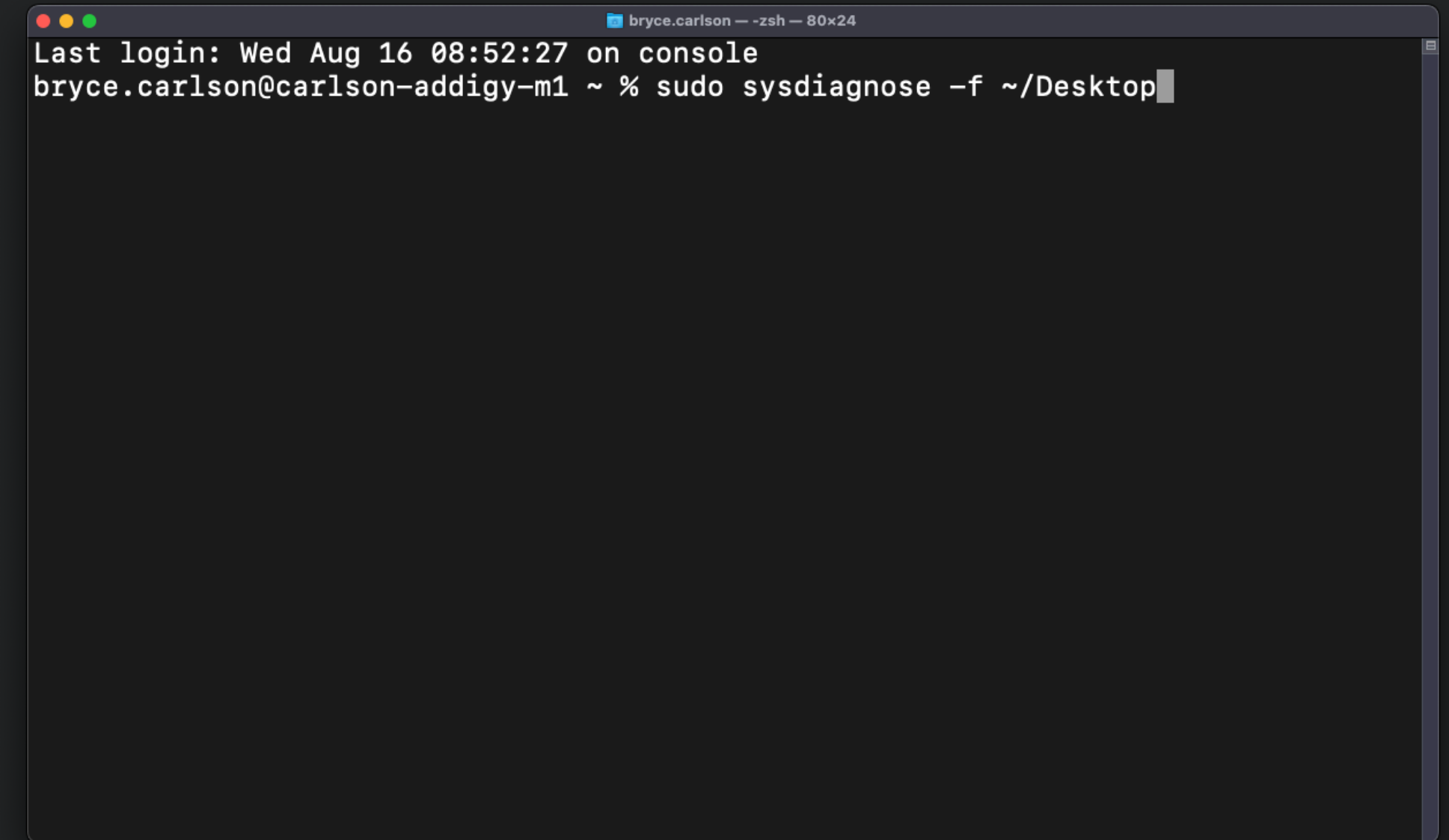
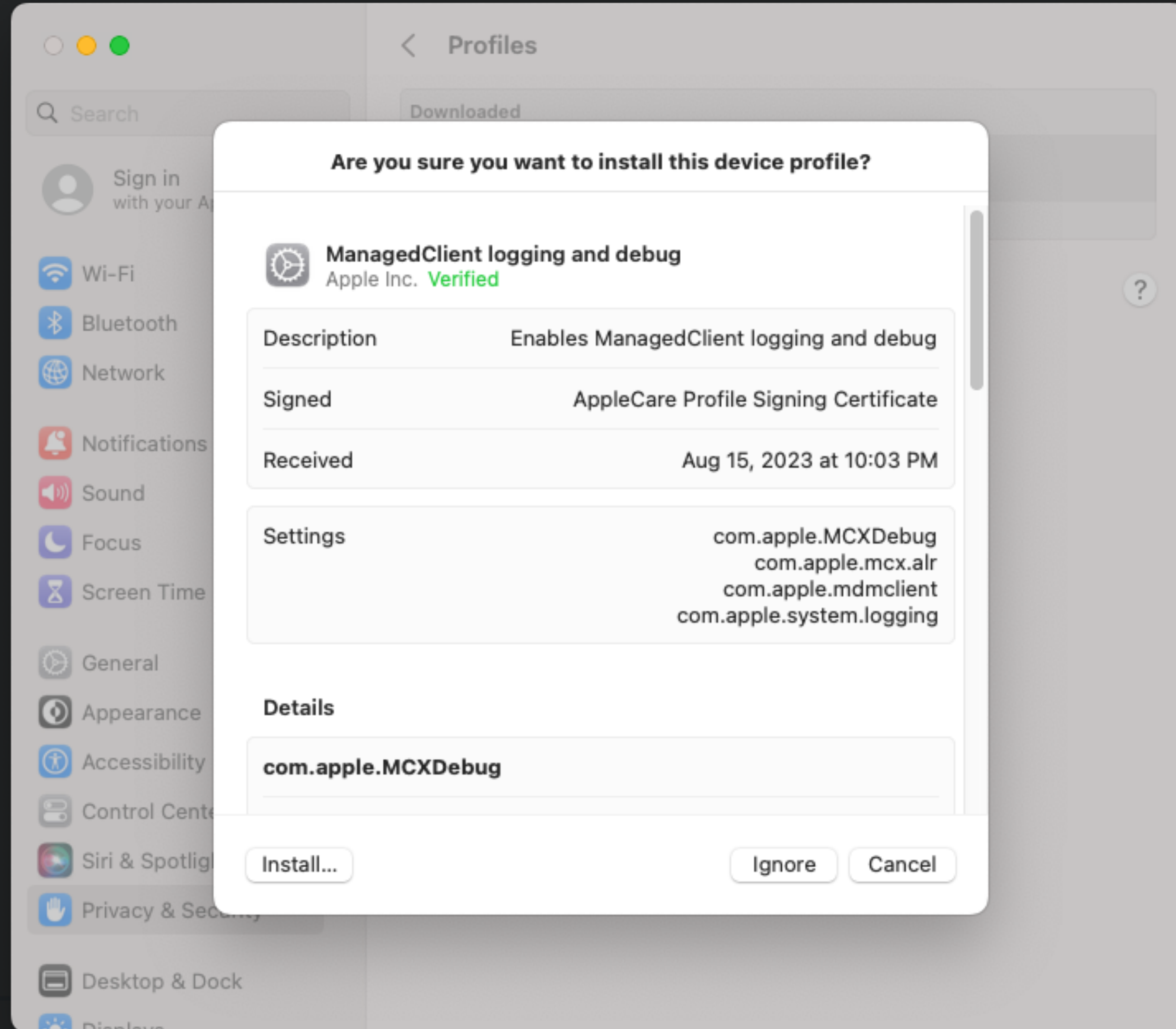
https://download.developer.apple.com/OS_X/OS_X_Logs/Managed_Client_Logging_Instructions.pdf

The screenshot shows the Apple Developer website interface. At the top, there is a navigation bar with links for 'Developer', 'News', 'Discover', 'Design', 'Develop', 'Distribute', 'Support', and 'Account'. Below this is a secondary navigation bar with 'Bug Reporting' on the left and 'Overview', 'Profiles and Logs', and 'Feedback Assistant' on the right. The main content area features a large heading 'Profiles and Logs' and a paragraph explaining that these profiles and logs are for developers to use to provide information about bugs to Apple. Below the text are filter tabs for 'All', 'iOS', 'macOS', 'tvOS', 'watchOS', and 'Other'. A search bar contains the text 'mana'. At the bottom of the main content area, there is a section for 'Managed Client for macOS' with two links: 'Instructions' and 'Profile'. A large red arrow points to the 'Profile' link. At the very bottom, a breadcrumb trail shows the path: 'Developer > Bug Reporting > Profiles and Logs'.

Debug Logging Profile for MDM



Debug Logging Profile for MDM



ScheduleOSUpdate (logs)

```
default      2023-08-15 22:15:15.348017 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] === OS Update Item ===
default      2023-08-15 22:15:15.348021 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Product Key:
MSU_UPDATE_22G74_patch_13.5_minor
default      2023-08-15 22:15:15.348026 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Title:                macOS
Ventura 13.5
default      2023-08-15 22:15:15.348031 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Version:              13.5 <Build:
22G74> <PMV: 13.5>
default      2023-08-15 22:15:15.348045 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Deferred:            no (Date: )
default      2023-08-15 22:15:15.348047 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Tags:                (null)
default      2023-08-15 22:15:15.348052 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] MacOSUpdate:        YES
default      2023-08-15 22:15:15.348056 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] MSU:                 YES (Major:
no Full: no DL: no label: macOS Ventura 13.5-22G74)
default      2023-08-15 22:15:15.348114 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Splat:              no <(null)>
(Revoked: no)
default      2023-08-15 22:15:15.348117 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] IsMacOSUpdate():    YES
default      2023-08-15 22:15:15.348119 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] IsMajorOSUpdateViaIA(): no
debug       2023-08-15 22:15:15.348124 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Download size:      1556959678
debug       2023-08-15 22:15:15.348128 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] ConfigData:        no
debug       2023-08-15 22:15:15.348131 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] FirmwareUpdate:    no
debug       2023-08-15 22:15:15.348134 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] ServerMetaDataURL: no
debug       2023-08-15 22:15:15.348138 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Allows InstallLater: YES
debug       2023-08-15 22:15:15.348144 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Apps to close:
debug       2023-08-15 22:15:15.348149 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Critical:          no
debug       2023-08-15 22:15:15.348153 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Mandatory:        no
debug       2023-08-15 22:15:15.348167 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Localization:     en-US
debug       2023-08-15 22:15:15.348175 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Long desc:        This update
provides important bug fixes and security updates and is recommended for all users. | | For detailed information about the security content of
this update, please visit: https://support.apple.com/kb/HT201222 |
debug       2023-08-15 22:15:15.348179 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Long attr desc:    YES
debug       2023-08-15 22:15:15.367501 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] License text:      YES
debug       2023-08-15 22:15:15.367506 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Firmware warning:  no
debug       2023-08-15 22:15:15.367510 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Post install:     <Restart>
debug       2023-08-15 22:15:15.367513 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] AutoUpdate:       no
debug       2023-08-15 22:15:15.367516 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Recommended:      YES
debug       2023-08-15 22:15:15.367521 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] Ramped:           no
debug       2023-08-15 22:15:15.367523 -0500      mdmclient    [0:MDMDaemon:OSUpdate:<0xbc25>] =====
```

ScheduleOSUpdate (logs)

```
default      2023-08-15 22:15:14.378034 -0500      softwareupdated [FSM(SUMacControllerScanManager)] >S> ScanReadyToBegin >E>
ScanForSUUpdates >N> ScanningForSoftwareUpdates >A> ScanForSoftwareUpdates info:[>>>
    Version(scanUUID:0BD2448F-97DA-4F43-A793-D68C33B5682F|targetRestoreVersion:(null)|targetBuildVersion:(null)|targetProductVersion:(null)|
Splat:NO|MajorBridgeOSUpdate:NO((null) (null) (null))|MinorBridgeOSUpdate:NO((null) (null) (null))|MajorRosettaUpdate:NO((null))|
MinorRosettaUpdate:NO((null))|MajorRecoveryOSUpdate:NO|MinorRecoveryOSUpdate:NO|scanFailed:NO)
    clientRequest: [>>>
    Client(clientID:softwareupdated|command:SUMacControllerCommandScan|cookie:9835D516-405B-40BF-86B9-FBD6959F61D4.<72>)
    Completions(completion:NO|scanCompletion:YES|documentationCompletion:NO|splatScanCompletion:NO|sfrScanCompletion:NO)
    proxyObject: <_NSXPCDistantObject: 0x1173b0800>
    progressManager: (null)
    accessControlContext: AccessControlContext(clientID:softwareupdated|priority:ClientInitiated|event:(null)|eventType:None|
clientDisconnectOccurred:NO|additionalContext:none)
    descriptor: (null)
    overrides: [>>>
    userInitiated:YES
    updateType:All
    localAuthenticationUserID:501
    localAuthenticationContext:present
    bridgeOSAllowTestMode:YES
    supervised:YES
    additionalUpdateMetricEventFields:present
<<<]
<<<]
    progress: (null)
    majorPrimarySU: (null)
    majorSecondarySU: (null)
    minorPrimarySU: (null)
    minorSecondarySU: (null)
    primarySFRSU: (null)
    secondarySFRSU: (null)
    splatUpdate: (null)
    majorUpdateBrain: (null)
    minorUpdateBrain: (null)
    majorError: (null)
    minorError: (null)
<<<]
default      2023-08-15 22:15:14.712502 -0500      softwareupdated [RecoveryOSManager] SPI: Starting new recoveryOS update
default      2023-08-15 22:15:14.851063 -0500      softwareupdated [RecoveryQueryUpdate] Querying for available updates with
targetMainOSVersion:13.5 targetMainOSBuild:22G74 options:{
    foreground = 1;
    MainOSUpdateAttributes = {
        ActualMinimumSystemPartition = 9102;
        AssetType = "com.apple.MobileAsset.MacSoftwareUpdate";
```


ScheduleOSUpdate (logs)

```
default      2023-08-15 22:15:14.378034 -0500      softwareupdated [FSM(SUMacControllerScanManager)] >S> ScanReadyToBegin >E>
ScanForSUUpdates >N> ScanningForSoftwareUpdates >A> ScanForSoftwareUpdates info:[>>>

default      2023-08-15 22:15:14.851063 -0500      softwareupdated [RecoveryQueryUpdate] Querying for available updates with
targetMainOSVersion:13.5 targetMainOSBuild:22G74 options:{
  PrerequisiteBuild = 22F66;
  PrerequisiteOSVersion = "13.4";

default      2023-08-15 22:15:14.971513 -0500      softwareupdated [RecoveryQueryUpdate] QueryNeRDUpdate completed with updateAttributes:
(null) required:NO info:{
  BrainVersion = 22D49;
  ProductBuildVersion = 22F66;
  ProductVersion = "13.4";
  RestoreLongVersion = "22.6.66.0.0,0";
} error:(null)

default      2023-08-15 22:15:14.975090 -0500      softwareupdated [FSM(scan[0BD2448F-97DA-4F43-A793-D68C33B5682F])] >S> Idle >E>
CheckForAvailableUpdate >N> DownloadingSUCatalog >A> DownloadSUCatalog

default      2023-08-15 22:15:14.975159 -0500      softwareupdated [POLICY] constructMASoftwareUpdateCatalogDownloadOptionsWithUUID for
MacUpdateBrainPolicy: SUCorePolicyMacUpdateBrain(compatibilityVersion:20|majorTargetBuildVersion:22G74|minorTargetBuildVersion:22G74|super:
enableOSPpersonalization:NO)

default      2023-08-15 22:15:14.975263 -0500      softwareupdated [POLICY] MacUpdateBrain catalog downloadOptions: MADownloadOptions
allowsCellular: 0 resourceTimeout: -1 canUseCacheServer: 0 discretionary: 0 sessionId: 0BD2448F-97DA-4F43-A793-D68C33B5682F additionalServerParams:
{   BuildVersion = 22F66;   CompatibilityVersion = 20;   DeviceName = Mac;   HWModelStr = VMA2MACOSAP;   ProductType = "VirtualMac2,1";
TargetBuildVersionArray =   (   22G74,   22G74   ); } allowsExpensiveAccess:1 requiresPowerPluggedIn: 0 prefersInfraWiFi: 1
liveServerOnly: -1 DownloadAuthorizationHeader: not present analyticsData: not present + MAMsuDownloadOptions reqProdVersion: (null) delayPeriod:
0 managedDevice: 0 allowSameVersion: 0 prereqBuild: (null) prereqVersion: (null) prereqReleaseType: (default) liveAssetAudienceUUID: (null)
purpose: (null)
default      2023-08-15 22:15:14.975267 -0500      softwareupdated [scan]startCatalogDownload... (su)

info         2023-08-15 22:15:36.471386 -0500      com.apple.MobileSoftwareUpdate.UpdateBrainService      6d41b000 : Cryptex patch
progress: 17%
default      2023-08-15 22:15:36.471493 -0500      softwareupdated [SPI] perform_prepare_command | PrepareUpdate | PROGRESS (Continue) |
state:<private>
info         2023-08-15 22:15:37.544348 -0500      com.apple.MobileSoftwareUpdate.UpdateBrainService      6d41b000 : Cryptex patch
progress: 18%
default      2023-08-15 22:15:37.544461 -0500      softwareupdated [SPI] perform_prepare_command | PrepareUpdate | PROGRESS (Continue) |
state:<private>
```

ScheduleOSUpdate (logs)

```
default      2023-08-15 22:15:14.378034 -0500      softwareupdated [FSM(SUMacControllerScanManager)] >S> ScanReadyToBegin >E>
ScanForSUUpdates >N> ScanningForSoftwareUpdates >A> ScanForSoftwareUpdates info:[>>>

default      2023-08-15 22:15:14.851063 -0500      softwareupdated [RecoveryQueryUpdate] Querying for available updates with
targetMainOSVersion:13.5 targetMainOSBuild:22G74 options:{
  PrerequisiteBuild = 22F66;
  PrerequisiteOSVersion = "13.4";

default      2023-08-15 22:15:14.971513 -0500      softwareupdated [RecoveryQueryUpdate] QueryNeRDUpdate completed with updateAttributes:




default      2023-08-15 22:15:14.975090 -0500      softwareupdated [FSM(scan[0BD2448F-97DA-4F43-A793-D68C33B5682F])] >S> Idle >E>
CheckForAvailableUpdate >N> DownloadingSUCatalog >A> DownloadSUCatalog


default      2023-08-15 22:15:14.975159 -0500      softwareupdated [POLICY] constructMASoftwareUpdateCatalogDownloadOptionsWithUUID for
MacUpdateBrainPolicy: SUCorePolicyMacUpdateBrain(compatibilityVersion:20|majorTargetBuildVersion:22G74|minorTargetBuildVersion:22G74|super:
enableOSPersonalization:NO)

default      2023-08-15 22:15:14.975263 -0500      softwareupdated [POLICY] MacUpdateBrain catalog downloadOptions: MADownloadOptions
allowsCellular: 0 resourceTimeout: -1 canUseCacheServer: 0 discretionary: 0 sessionId: 0BD2448F-97DA-4F43-A793-D68C33B5682F additionalServerParams:
{   BuildVersion = 22F66;   CompatibilityVersion = 20;   DeviceName = Mac;   HWModelStr = VMA2MACOSAP;   ProductType = "VirtualMac2,1";
TargetBuildVersionArray =   (   22G74,   22G74   ); } allowsExpensiveAccess:1 requiresPowerPluggedIn: 0 prefersInfraWiFi: 1
liveServerOnly: -1 DownloadAuthorizationHeader: not present analyticsData: not present + MAMsuDownloadOptions reqProdVersion: (null) delayPeriod:
0 managedDevice: 0 allowSameVersion: 0 prereqBuild: (null) prereqVersion: (null) prereqReleaseType: (default) liveAssetAudienceUUID: (null)
purpose: (null)
default      2023-08-15 22:15:14.975267 -0500      softwareupdated [scan]startCatalogDownload... (su)

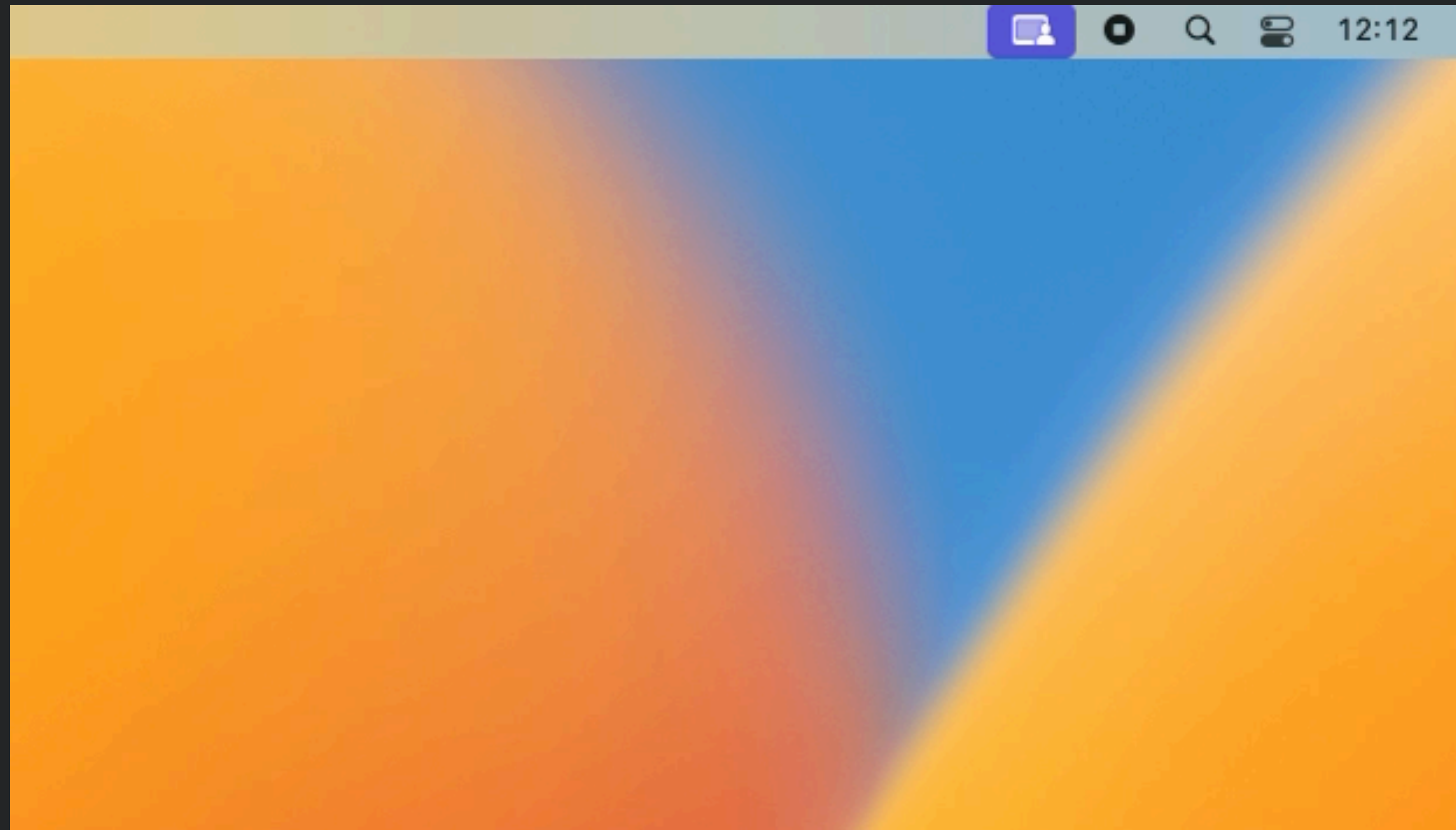
info         2023-08-15 22:15:36.471386 -0500      com.apple.MobileSoftwareUpdate.UpdateBrainService 6d41b000 : Cryptex patch
progress: 17%
default      2023-08-15 22:15:36.471493 -0500      softwareupdated [SPI] perform_prepare_command | PrepareUpdate | PROGRESS (Continue) |
state:<private>
info         2023-08-15 22:15:37.544348 -0500      com.apple.MobileSoftwareUpdate.UpdateBrainService 6d41b000 : Cryptex patch
progress: 18%
default      2023-08-15 22:15:37.544461 -0500      softwareupdated [SPI] perform_prepare_command | PrepareUpdate | PROGRESS (Continue) |
state:<private>
```

System Updates via MDM

  **Required managed update** 

This update requires at least 50% battery or connection to a power source. Your computer will restart to perform the update. You may defer 3 more times. 

- Install
- Try Tonight
- Remind Me Tomorrow

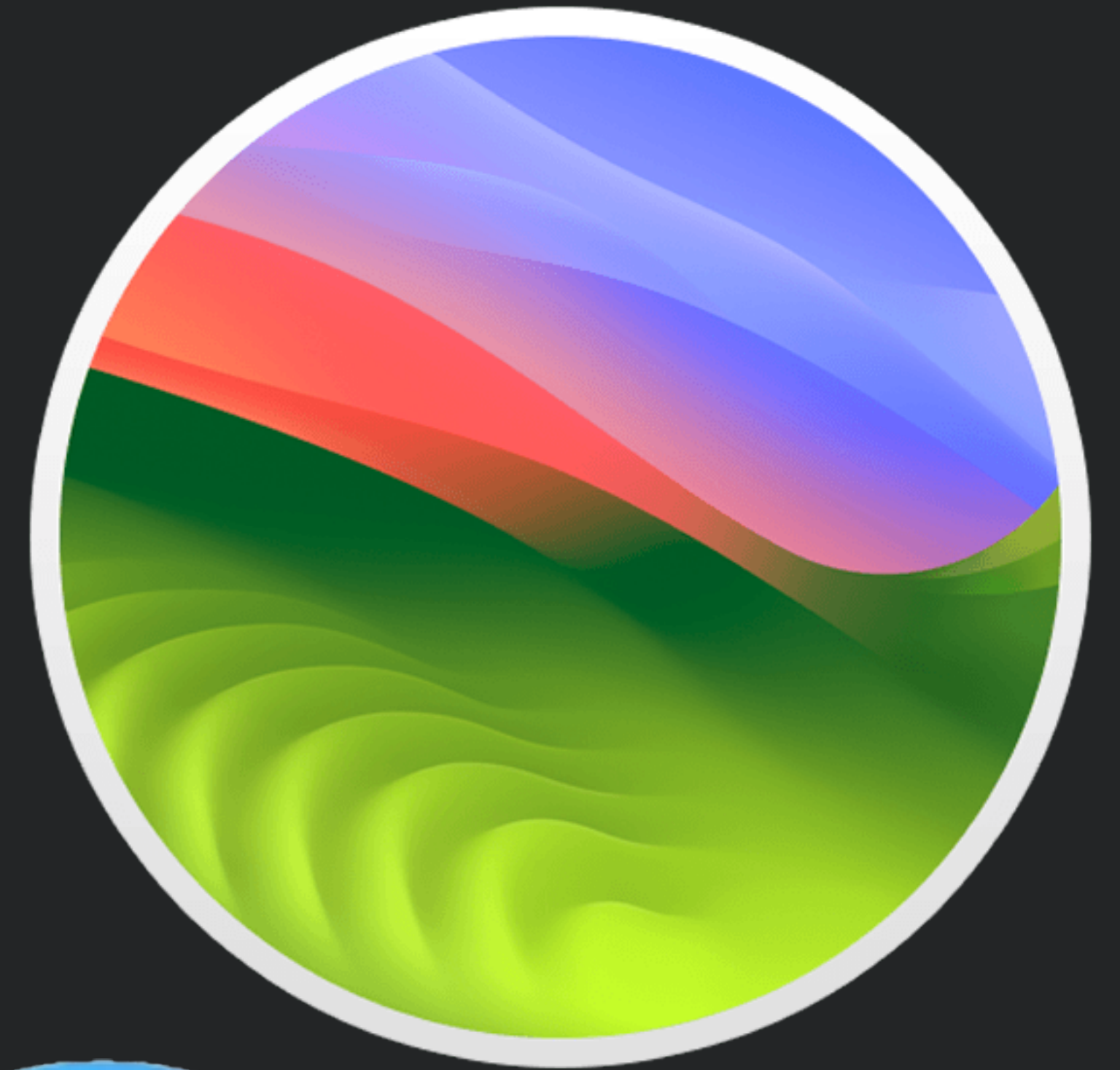


DDM Details

- Introduced at WWDC 2021, added to at WWDC 2022
- Nothing necessary from an administration standpoint
- Why is this important?
 - Allows devices to apply management logic without prompting from the server
 - Asynchronous status reporting
 - No need to poll devices
- The way forward for device management

DDM Details

- Platform Support
 - iOS
 - iPadOS
 - macOS
 - tvOS
 - watchOS



DDM Details

- 3 key components
 - Declarations
 - Support device functionality
 - Status
 - Track changes in device state
 - Extensibility
 - Allow devices and servers to communicate the changes in their capabilities



That is great.... but... updates

Where's the Beef?



Software Updates via DDM

- New functions/features with DDM over MDM
- New status items which can report on:
 - Software update status on a device
 - Details of installation state
 - Failures and failure reasons
- Enhanced user deferral workflow
- DDM software update configurations can coexist with MDM software update commands
- One thing stood out...
 - `softwareupdated` still relevant



developer.apple.com/documentation/devicemanagement/softwareupdateenforcementspecific

Developer News Discover Design Develop Distribute Support Account

Device Management Documentation / Device Management / SoftwareUpdateEnforcementSpecific **Beta** Language: Swift API Changes: Show

SoftwareUpdateEnforcementSpecific **Beta**

iOS 17.0+ **Beta** iPadOS 17.0+ **Beta** macOS 14.0+ **Beta**

Properties

DetailsURL	string
TargetBuildVersion	string
TargetLocalDateTime	string (Required)
TargetOSVersion	string (Required)

Beta Software

This documentation contains preliminary information about an API or technology in development. This information is subject to change, and software implemented according to this documentation should be tested with final operating system software.

[Learn more about using Apple's beta software >](#)

Filter /

- object ScreenSharingConnection **Beta**
- object ScreenSharingConnectionDisp... **Beta**
- object ScreenSharingConnectionGroup **Beta**
- object ScreenSharingHostSettings **Beta**
- object SecurityCertificate **Beta**
- object SecurityIdentity **Beta**
- object SecurityPasskeyAttestation **Beta**
- object ServicesConfigurationFiles **Beta**
- object SoftwareUpdateEnforcemen... **Beta****
- object StatusDeviceModelNumber **Beta**
- object StatusDiskManagementFileVa... **Beta**
- object StatusSecurityCertificateList **Beta**
- object StatusSecurityCertificateListC... **Beta**
- object StatusServicesBackgroundTask **Beta**
- object StatusServicesBackgroundTas... **Beta****
- object StatusServicesBackgroundTas... **Beta**
- object StatusSoftwareUpdateFailureR... **Beta**
- object StatusSoftwareUpdateFailureR... **Beta**
- object StatusSoftwareUpdateInstallR... **Beta**
- object StatusSoftwareUpdateInstallR... **Beta**
- object StatusSoftwareUpdateInstallSt... **Beta**
- object StatusSoftwareUpdatePendin... **Beta**
- object StatusSoftwareUpdatePendin... **Beta**

Example Configuration

```
com.apple.configuration.softwareupdate.enforcement.specific  
  
{ "TargetOSVersion": "14.0",  
  "TargetBuildVersion": "23A500",  
  "TargetLocalDateTime": "2023-06-05T10:00:00" }
```

`TargetOSVersion`

OS version the device should update to

`TargetBuildVersion`

Will take precedence over `TargetOSVersion` if both are set, if no update is found for the specified version the device will search for the default/recommended update offered by software update service

`TargetLocalDateTime`

Specifies local date and time when the update will be forced

MDM Watchdog



What is the big deal?

Updates = Security

- `softwareupdated`
- `mdmclient`



Updates Not Installed

Some updates could not be installed.

WHO LET THE DOGS OUT?



How it works!



#1

Checks to see if the device is enrolled into an MDM server

How it works!

#2



Checks to see if the device has a valid identity certificate in place

How it works!

#3



Checks if `mdmclient` process is not stuck using the last 90 minutes of logs from the `mdmclient` process
Part of this is checking the logs for HTTP 200 entries

How it works!

#4



Checks `softwareupdated` using the logs from `mdmclient` and `ManagedClient` subsystem

Looking at the last command received from MDM having to do with the OS Update processes

What then?

MDM Watchdog unloads / reloads services



mdmclient

```
`/bin/launchctl kickstart -k system/com.apple.mdmclient.daemon`
```


softwareupdated

```
`/usr/bin/dscacheutil -flushcache`  
`/usr/bin/killall -HUP mDNSResponder`  
`/usr/bin/pkill BrainService`  
`/bin/launchctl kickstart k system/com.apple.softwareupdated`
```

What then?

MDM Watchdog unloads / reloads services



 **frogor** 8 months ago

[@1macm](#) Hah, no, I work for Apple 😊

There is separately other issues solved by later versions of macOS (such as 12.6.1) in regards to even seeing updates -at all- in the first place.

If you've got a device like that at hand - try doing: `sudo launchctl kickstart -k system/com.apple.softwareupdated` while you've got System Preferences closed - then try re-opening and see if the update shows up.

If you've got a fleet of pre-12.6.1 devices still at this point and you're having a lot of problems seeing 12.6.1 in the first place on them, even when trying to use the launchctl command above to restart the software update lookup daemons, then you may want to consider using the 12.6.1 (or 12.6.2) full installer to get those devices updated ASAP.

👍 1 🗨️

softwareupdated

```
`/usr/bin/dscacheutil -flushcache`  
`/usr/bin/killall -HUP mDNSResponder``  
`/usr/bin/pkill BrainService``  
`/bin/launchctl kickstart k system/com.apple.softwareupdated`
```

What then?

MDM Watchdog unloads / reloads services



mdmclient

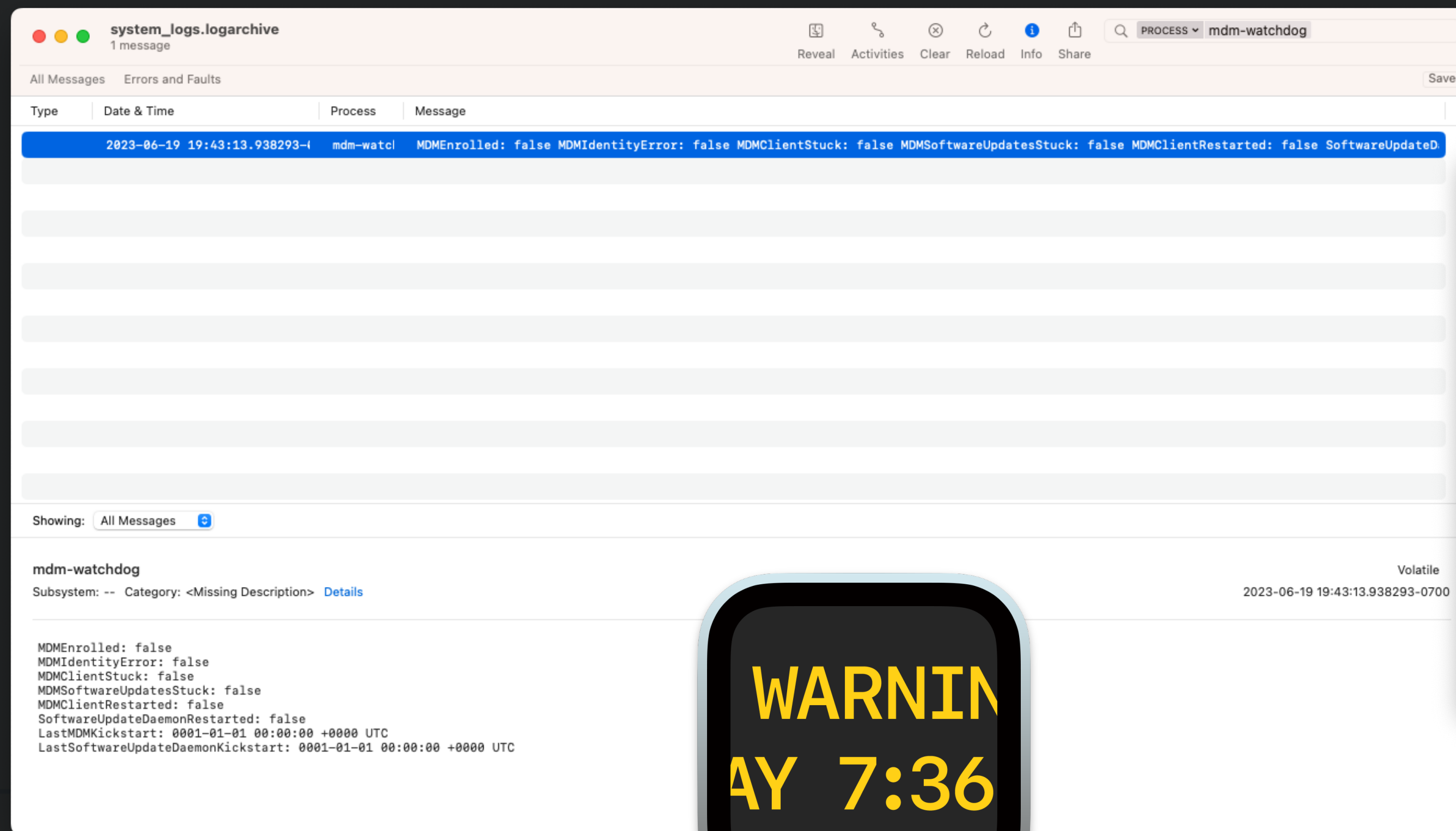
```
`/bin/launchctl kickstart -k system/com.apple.mdmclient.daemon`
```

What can I expect?

- **/Library/LaunchDaemons**
 - com.addigy.mdm-watchdog.plist
- **Run every 95 minutes**
- **Command Line Tool**
 - mdm-watchdog
- **Does not use network resources or phone home**
 - Does not auto update
 - Updates can be found at <https://addigy.com/mdm-watchdog>

What can I expect?

- log show –predicate ‘process = “mdm-watchdog”’
- Terminal: mdm-watchdog



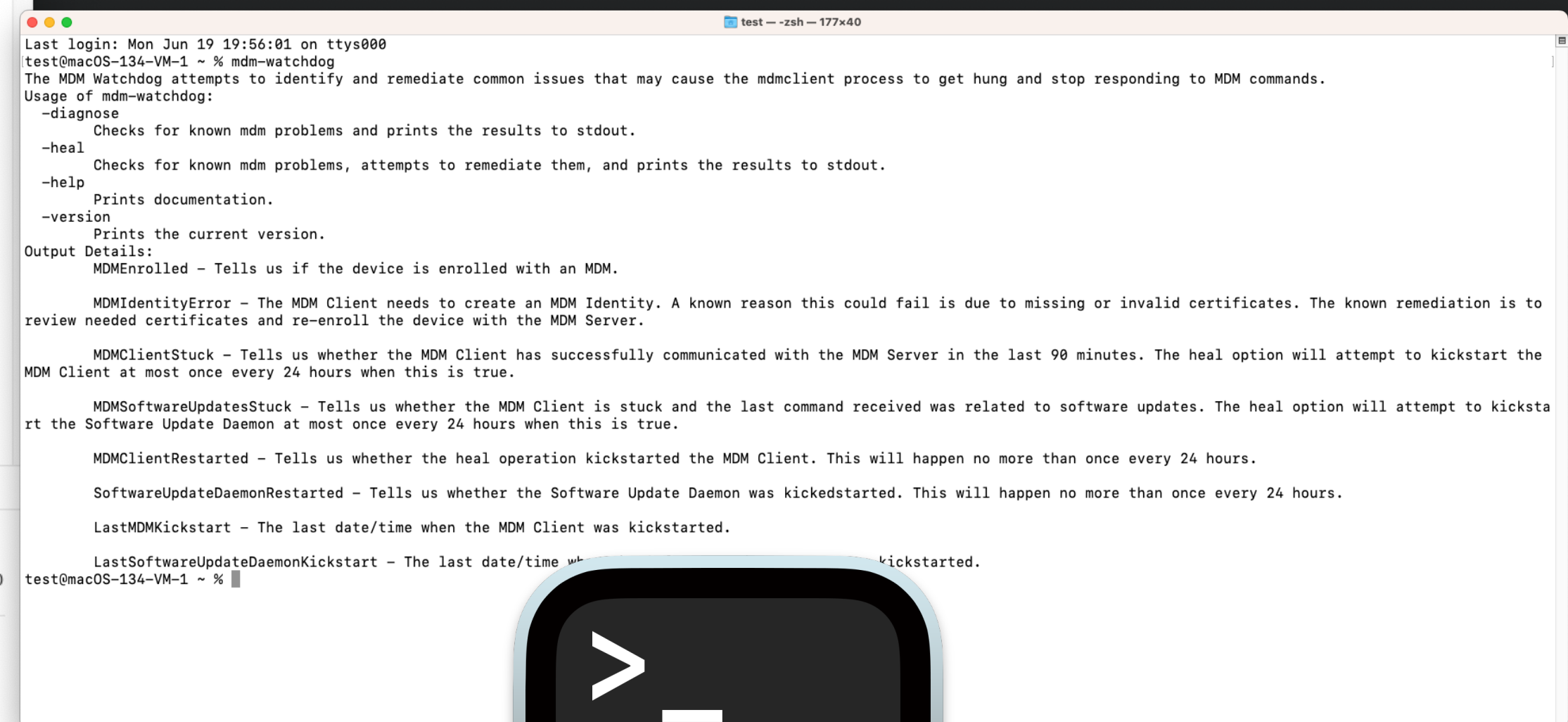
The screenshot shows the 'system_logs.logarchive' application with a search filter for 'PROCESS mdm-watchdog'. A single log message is displayed in a table:

Type	Date & Time	Process	Message
	2023-06-19 19:43:13.938293-4	mdm-watcl	MDMEnrolled: false MDMIdentityError: false MDMClientStuck: false MDMSoftwareUpdatesStuck: false MDMClientRestarted: false SoftwareUpdateD

Below the table, the details for the 'mdm-watchdog' process are shown:

```
mdm-watchdog
Subsystem: -- Category: <Missing Description>
MDMEnrolled: false
MDMIdentityError: false
MDMClientStuck: false
MDMSoftwareUpdatesStuck: false
MDMClientRestarted: false
SoftwareUpdateDaemonRestarted: false
LastMDMKickstart: 0001-01-01 00:00:00 +0000 UTC
LastSoftwareUpdateDaemonKickstart: 0001-01-01 00:00:00 +0000 UTC
```

WARNING
AY 7:36



The terminal window shows the output of the 'mdm-watchdog' command:

```
Last login: Mon Jun 19 19:56:01 on ttys000
test@macOS-134-VM-1 ~ % mdm-watchdog
The MDM Watchdog attempts to identify and remediate common issues that may cause the mdmclient process to get hung and stop responding to MDM commands.
Usage of mdm-watchdog:
  -diagnose
    Checks for known mdm problems and prints the results to stdout.
  -heal
    Checks for known mdm problems, attempts to remediate them, and prints the results to stdout.
  -help
    Prints documentation.
  -version
    Prints the current version.
Output Details:
  MDMEnrolled - Tells us if the device is enrolled with an MDM.
  MDMIdentityError - The MDM Client needs to create an MDM Identity. A known reason this could fail is due to missing or invalid certificates. The known remediation is to review needed certificates and re-enroll the device with the MDM Server.
  MDMClientStuck - Tells us whether the MDM Client has successfully communicated with the MDM Server in the last 90 minutes. The heal option will attempt to kickstart the MDM Client at most once every 24 hours when this is true.
  MDMSoftwareUpdatesStuck - Tells us whether the MDM Client is stuck and the last command received was related to software updates. The heal option will attempt to kickstart the Software Update Daemon at most once every 24 hours when this is true.
  MDMClientRestarted - Tells us whether the heal operation kickstarted the MDM Client. This will happen no more than once every 24 hours.
  SoftwareUpdateDaemonRestarted - Tells us whether the Software Update Daemon was kickedstarted. This will happen no more than once every 24 hours.
  LastMDMKickstart - The last date/time when the MDM Client was kickstarted.
  LastSoftwareUpdateDaemonKickstart - The last date/time when the Software Update Daemon was kickedstarted.
```

>
|

What can I expect?

- `log show --predicate 'process = "mdm-watchdog"'`

```
default 2023-06-19 19:43:13.938293 -0700 mdm-watchdog MDMEnrolled:  
false  
MDMIdentityError: false  
MDMClientStuck: false  
MDMSoftwareUpdatesStuck: false  
MDMClientRestarted: false  
SoftwareUpdateDaemonRestarted: false  
LastMDMKickstart: 0001-01-01 00:00:00 +0000 UTC  
LastSoftwareUpdateDaemonKickstart: 0001-01-01 00:00:00 +0000 UTC
```

WARNING
7:36

What can I expect?

•Terminal: `mdm-watchdog`



The MDM Watchdog attempts to identify and remediate common issues that may cause the `mdmclient` process to get hung and stop responding to MDM commands.

Usage of `mdm-watchdog`:

`-diagnose`

Checks for known `mdm` problems and prints the results to `stdout`.

`-heal`

Checks for known `mdm` problems, attempts to remediate them, and prints the results to `stdout`.

`-help`

Prints documentation.

`-version`

Prints the current version.

Output Details:

`MDMEnrolled` - Tells us if the device is enrolled with an MDM.

`MDMIdentityError` - The MDM Client needs to create an MDM Identity. A known reason this could fail is due to missing or invalid certificates. The known remediation is to review needed certificates and re-enroll the device with the MDM Server.

`MDMClientStuck` - Tells us whether the MDM Client has successfully communicated with the MDM Server in the last 90 minutes. The `heal` option will attempt to kickstart the MDM Client at most once every 24 hours when this is true.

`MDMSoftwareUpdatesStuck` - Tells us whether the MDM Client is stuck and the last command received was related to software updates. The `heal` option will attempt to kickstart the Software Update Daemon at most once every 24 hours when this is true.

`MDMClientRestarted` - Tells us whether the `heal` operation kickstarted the MDM Client. This will happen no more than once every 24 hours.

`SoftwareUpdateDaemonRestarted` - Tells us whether the Software Update Daemon was kickedstarted. This will happen no more than once every 24 hours.

`LastMDMKickstart` - The last date/time when the MDM Client was kickstarted.

`LastSoftwareUpdateDaemonKickstart` - The last date/time when the Software Update Daemon was kickstarted.

What can I expect?

• **Terminal:** `mdm-watchdog -diagnose`

```
bryce@macOS-134-VM-1 ~ % mdm-watchdog -diagnose
MDMEnrolled: false
MDMIdentityError: false
MDMClientStuck: false
MDMSoftwareUpdatesStuck: false
```

MDM Watchdog in Addigy vs Standalone with any MDM

	Addigy	Other MDMs
Run Frequency	Every 5 min. audit then 30 min. agent	Every 95 minutes
Reporting	GoLive, Alert, and Dashboard	log show or scripted CMD
Identity Certificate	Built in device fact	log show or scripted CMD
MDM Client Stuck	Built in device fact	log show or scripted CMD
Software Update Stuck	Built in device fact	log show or scripted CMD
Software Updates via MDM Retry Logic	24 hour default; 1 to 24 hour increments	Depends
Deployment	Part of Addigy binary	.pkg
Updates	Part of Addigy binary	New .pkg

MDM Watchdog



• <https://addigy.com/mdm-watchdog/>

• <https://agents.addigy.com/tools/mdm-watchdog/latest/mdm-watchdog.pkg>



Thanks!

addigy.com

[@addigy](https://twitter.com/addigy)



MacAdmins [#addigy](https://twitter.com/addigy)