Behind the MDM Curtain

Uh oh, what did I just run?

Joseph Chilcote | MacAdmins 2023

kandji

Product Pricing Resources Support Company Log in Request Access

What is the Apple MDM protocol?

The Apple MDM protocol is a Device Management API created by Apple to allow MDM vendors to communicate with Apple devices, change settings, enable controls, move data, and more. IT admins use an MDM vendor's software to execute commands that Apple has made available in their protocol. For Mac management, MDM vendors can create a macOS agent, which can augment Apple's MDM framework with their own software and extend IT's capabilities beyond the MDM protocol.

- https://www.kandji.io/apple-mdm-definitions/what-is-the-apple-mdm-protocol

Web Service

Device Management

Manage your organization's devices remotely.

iOS 13.0+ | iPadOS 13.0+ | macOS 10.15+ | tvOS 13.0+ | watchOS 6.0+ | Device Assignment Services 5.0+ | VPP License Management 1.0.0+

Overview

Deploying a mobile device management (MDM) solution allows administrators to securely and remotely configure enrolled devices. Administrators use Apple School Manager or Apple Business Manager to enroll organization-owned devices, and users can enroll their own devices. Once a device is enrolled, administrators can update software and device settings, monitor compliance with organizational policies, remotely erase or lock devices, and install apps and books developed in-house or purchased through Apple School Manager or Apple Business Manager.

- https://developer.apple.com/documentation/devicemanagement

Documentation / Device Manage... / Commands and Q... / Set Recovery Lock Co... / SetRecoveryLockCo...

Device Management Command

SetRecoveryLockCommand

The command to set the password for Recovery Lock.

macOS 11.5+

Properties

Command `SetRecoveryLockCommand.Command`	**(Required)** The command dictionary.
CommandUUID `string`	**(Required)** The unique identifier of the command.

- https://developer.apple.com/documentation/devicemanagement

Product Solutions Open Source Pricing Search

apple / device-management Public

Code Pull requests Security Insights

release 2 branches 0 tags Go to file Code

cyrusdaboo Release_iOS-16-4_macOS-13-3 5a8fb0d on Apr 5 7 commits

.github	Release_iOS-15_macOS-12	last year
declarative	Release_iOS-16-4_macOS-13-3	3 months ago
docs	Release_iOS-16-2_macOS-13-1	6 months ago
mdm	Release_iOS-16-4_macOS-13-3	3 months ago
other	Release_iOS-16-4_macOS-13-3	3 months ago
.gitignore	Release_iOS-15_macOS-12	last year
LICENSE.txt	Release_iOS-15_macOS-12	last year
README.md	Release_iOS-16-4_macOS-13-3	3 months ago

- https://github.com/apple/device-management

MDM Command Schema

github / apple / device-management

```
title: Set Recovery Lock Command
description: Sets or clears the recovery lock password (AppleSilicon devices only)
payload:
  requesttype: SetRecoveryLock
  supportedOS:
    macOS:
      introduced: '11.5'
```

MDM Commands

hope you like plists!

- Install apps & pkgs
- Erase & lock devices
- Enable features & settings
- Query device & security info
- Software updates
- Install profiles

MDM Vendor Implementations

abstractions

- Subtle differences
- Product synergies
- User experience

JumpCloud

OS Updates

OS	13.3.1 (22E261)
Total Number of Versions Available	1
Last Scanned for Updates	07-14-2023 at 12:38pm

Select an OS Update to Schedule

Available Updates

macOS Ventura 13.4.1 (Minor) 22F82

Install Action

Install Later Today

Schedule...

Kandji

Updates

Specify how macOS versions are automatically enforced. Learn more...

Version enforcement

Manually enforce a minimum version

Specify if updates should not be managed, automatically be enforced after they are released, or select a minimum macOS version for enforcement.

⚠ The latest macOS version available at the time of install will be used regardless of these settings.

Minimum version

13.4.1

Specify the minimum macOS version to enforce.

Enforcement deadline

July 14, 2023

(GMT-08:00) Pacific Time - Los Angeles

12:00 PM (Noon)

Specify when the macOS version should be enforced. The update will be cached and users will be able to voluntarily update five (5) days before enforcement deadline, or sooner if a shorter time frame is selected.

Workspace ONE

Update

Update Name	macOS Sonoma 14 Beta 3
Version	14.0
Device Installation Method *	Download/Install the software update

SEND CANCEL

SimpleMDM

Update OS Version

This will send an OS update command to the selected devices.

OS Update version

✓ 12.6.7
13.4.1

OS Update Mode

- (●) **Smart Update (Install Later)**
  Install the update when macOS deems it to be an opportune time.
- ( ) **Notify Only**
  Download the software update and notify the user through the App Store.
- ( ) **Download Only**
  Download the software update without installing it.
- ( ) **Install As Soon As Possible**
  When the OS update has finished downloading, the user will be prompted and shown a timer before a restart occurs.
- ( ) **Force Update**
  Download and install the update immediately.

Allowed deferrals

3

Cancel | Update Devices

OS Updates

OS

Total Number of Versions Available

Last Scanned for Updates

Select an OS Update to Schedule

Available Updates

macOS Ventura 13.4.1 (Minor) 22F82

Install Action

Install Later Today

Sched

Updates

Specify how macOS versions are automatically enforced. Learn more...

Version enforcement

Manually enforce a minimum version

Specify if updates should not be managed, automatically be enforced after they are released, or select a minimum macOS version for enforcement.

The latest macOS version available at the time of install will be used regardless of these settings.

Minimum version

13.4.1

e will be cached and users will be
line, or sooner if a shorter time

Update

Update Name macOS Sonoma 14 Beta 3

Version 14.0

Device Installation Method * Download/Install the software update

SEND CANCEL

Update OS Version

This will send an OS update command to the selected devices.

OS Update version ✓ 12.6.7
13.4.1

OS Update Mode

- Smart Update (Install Later)
  Install the update when macOS deems it to be an opportune time.
- Notify Only
  Download the software update and notify the user through the App Store.
- Download Only
  Download the software update without installing it.
- Install As Soon As Possible
  When the OS update has finished downloading, the user will be prompted and shown a timer before a restart occurs.
- Force Update
  Download and install the update immediately.

Allowed deferrals 3

Cancel Update Devices

Anatomy of a command

Commands & Responses

anatomy of a command

```
<!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>DeviceInformation</string>
    </dict>
    <key>CommandUUID</key>
    <string>4c6de625-16a6-45df-8073-73cb3fca7866</string>
</dict>
</plist>
```

Commands & Responses

anatomy of a command

```
<!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>DeviceInformation</string>
    </dict>
    <key>CommandUUID</key>
    <string>4c6de625-16a6-45df-8073-73cb3fca7866</string>
</dict>
</plist>
```

Commands & Responses

anatomy of a command

```
<!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>DeviceInformation</string>
    </dict>
    <key>CommandUUID</key>
    <string>4c6de625-16a6-45df-8073-73cb3fca7866</string>
</dict>
</plist>
```

Commands & Responses

anatomy of a response


```
  <key>CommandUUID</key>
  <string>291a2336-8cf2-4ae7-9f35-3341baf8b0ce</string>
  <key>QueryResponses</key>
  <dict>
    <key>IsAppleSilicon</key>
      <true/>
     <key>Model</key>
      <string>Mac14,2</string>
     <key>OSVersion</key>
      <string>14.0</string>
</dict>
```

Commands & Responses

anatomy of a response


```
<key>CommandUUID</key>
<string>291a2336-8cf2-4ae7-9f35-3341baf8b0ce</string>
<key>QueryResponses</key>
<dict>
  <key>IsAppleSilicon</key>
    <true/>
   <key>Model</key>
    <string>Mac14,2</string>
   <key>OSVersion</key>
    <string>14.0</string>
</dict>
```

Commands & Responses

https://developer.apple.com/documentation/devicemanagement/get_device_information

```
    <key>Command</key>
    <dict>
        <key>Queries</key>
        <array>
            <string>UDID</string>
            <string>Languages</string>
            <string>Locales</string>
            <string>DeviceID</string>
            <string>OrganizationInfo</string>
            <string>LastCloudBackupDate</string>
            <string>AwaitingConfiguration</string>
            <string>MDMOptions</string>
            <string>iTunesStoreAccountIsActive</string>
```

```
            <string>AutomaticSecurityUpdatesEnabled</string>
            <string>OSUpdateSettings</string>
            <string>LocalHostName</string>
            <string>HostName</string>
            <string>IsMultiUser</string>
            <string>IsMDMLostModeEnabled</string>
            <string>MaximumResidentUsers</string>
            <string>PushToken</string>
            <string>DiagnosticSubmissionEnabled</string>
            <string>AppAnalyticsEnabled</string>
            <string>IsNetworkTethered</string>
            <string>ServiceSubscriptions</string>
        </array>
        <key>RequestType</key>
        <string>DeviceInformation</string>
    </dict>
```

Commands & Responses

anatomy of an error

```
<key>ErrorChain</key>
<array>
    <dict>
        <key>ErrorCode</key>
        <integer>94</integer>
        <key>ErrorDomain</key>
        <string>MDMClientError</string>
        <key>LocalizedDescription</key>
        <string>Client crashed processing: InstallProfile</string>
    </dict>
</array>
<key>Status</key>
<string>Error</string>
```

Commands & Responses

anatomy of an error

```
<key>ErrorChain</key>
<array>
    <dict>
        <key>ErrorCode</key>
        <integer>94</integer>
        <key>ErrorDomain</key>
        <string>MDMClientError</string>
        <key>LocalizedDescription</key>
        <string>Client crashed processing: InstallProfile</string>
    </dict>
</array>
<key>Status</key>
<string>Error</string>
```

Crafting a command

GitHub / nanomdm / tools / cmdr.py

```
% ./cmdr.py DeviceInformation
<?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>DeviceInformation</string>
    </dict>
    <key>CommandUUID</key>
    <string>809d2c1e-9d74-4164-97cd-deb1b9285ecd</string>
</dict>
</plist>
```

Crafting a command

GitHub / nanomdm / tools / cmdr.py

```
% ./cmdr.py DeviceLock 123456
<dict>
    <key>Command</key>
    <dict>
        <key>PIN</key>
        <string>123456</string>
        <key>RequestType</key>
        <string>DeviceLock</string>
    </dict>
    <key>CommandUUID</key>
    <string>6e395d79-ef4f-4973-acc4-7ddd79b00ece</string>
</dict>
</plist>
```

Crafting a command

GitHub / nanomdm / tools / cmdr.py

```
<key>Command</key>
 <dict>
    <key>RequestType</key>
    <string>ScheduleOSUpdate</string>
    <key>Updates</key>
    <array>
       <dict>
          <key>InstallAction</key>
          <string>InstallASAP</string>
         <key>ProductVersion</key>
          <string>13.5</string>
       </dict>
    </array>
 </dict>
```

Software Update

everyone’s favorite topic!

Software Update

everyone’s favorite topic!

- ScheduleOSUpdateScan
- AvailableOSUpdates
- ScheduleOSUpdate
- OSUpdateStatus

Software Update

everyone’s favorite topic!

- ScheduleOSUpdateScan
- AvailableOSUpdates
- ScheduleOSUpdate
- OSUpdateStatus

ScheduleOSUpdate

everyone’s favorite subtopic!

- DownloadOnly
- NotifyOnly
- InstallLater (w/deferrals)
- InstallASAP
- InstallForceRestart

ScheduleOSUpdate

github / nanomdm / tools / cmd.py

```
% ./cmdr.py ScheduleOSUpdate -h
usage: cmdr.py ScheduleOSUpdate [-h] [--version VERSION] [--key KEY] [--deferrals DEFERRALS] [--priority PRIORITY] action

positional arguments:
  action                InstallAction (ex. InstallASAP, InstallLater, etc.)

optional arguments:
  -h, --help            show this help message and exit
  --version VERSION     ProductVersion (ex. 12.1, 12.2.1, etc.)
  --key KEY             ProductKey (ex. MSU_UPDATE_21E5227a_patch_12.3, etc.)
  --deferrals DEFERRALS
                        MaxUserDeferrals (ex. 3, 30, etc.)
  --priority PRIORITY   Priority (ex. Low, High.)
```

ScheduleOSUpdate

github / nanomdm / tools / cmd.py


```
  <key>AvailableOSUpdates</key>
   <array>
      <dict>
         <key>Build</key>
          <string>22F82</string>
         <key>ProductKey</key>
          <string>MSU_UPDATE_22F82_patch_13.4.1_minor</string>
          <key>Version</key>
          <string>13.4.1</string>
      </dict>
   </array>
</dict>
```

ScheduleOSUpdate

https://gdmf.apple.com/v2/pmv

gdmf.apple.com


```
{"PublicAssetSets":{"iOS":[{"ProductVersion":"12.5.7","PostingDate":"2023-06-28","ExpirationDate":"2023-10-07","SupportedDevices":
["iPad4,1","iPad4,2","iPad4,3","iPad4,4","iPad4,5","iPad4,6","iPad4,7","iPad4,8","iPad4,9","iPhone6,1","iPhone6,2","iPhone7,1","iPhone7,
{"ProductVersion":"15.7.7","PostingDate":"2023-06-28","ExpirationDate":"2023-10-07","SupportedDevices":
["iPad5,1","iPad5,2","iPad5,3","iPad5,4","iPhone8,1","iPhone8,2","iPhone8,4","iPhone9,1","iPhone9,2","iPhone9,3","iPhone9,4","iPod9,1"]}
06-28","ExpirationDate":"2023-10-07","SupportedDevices":
["iPad11,1","iPad11,2","iPad11,3","iPad11,4","iPad11,6","iPad11,7","iPad12,1","iPad12,2","iPad13,1","iPad13,10","iPad13,11","iPad13,16",
ad13,4","iPad13,5","iPad13,6","iPad13,7","iPad13,8","iPad13,9","iPad14,1","iPad14,2","iPad14,3","iPad14,4","iPad14,5","iPad14,6","iPad6,
6,8","iPad7,1","iPad7,11","iPad7,12","iPad7,2","iPad7,3","iPad7,4","iPad7,5","iPad7,6","iPad8,1","iPad8,10","iPad8,11","iPad8,12","iPad8
,7","iPad8,8","iPad8,9","iPhone10,1","iPhone10,2","iPhone10,3","iPhone10,4","iPhone10,5","iPhone10,6","iPhone11,2","iPhone11,6","iPhone1
e12,8","iPhone13,1","iPhone13,2","iPhone13,3","iPhone13,4","iPhone14,2","iPhone14,3","iPhone14,4","iPhone14,5","iPhone14,6","iPhone14,7"
{"ProductVersion":"5.3.9","PostingDate":"2023-06-21","ExpirationDate":"2023-10-07","SupportedDevices":
["Watch2,3","Watch2,4","Watch2,6","Watch2,7","Watch3,1","Watch3,2","Watch3,3","Watch3,4","Watch4,1","Watch4,2","Watch4,3","Watch4,4"]},{
21","ExpirationDate":"2023-10-07","SupportedDevices":["Watch2,3","Watch2,4","Watch2,6","Watch2,7"]},{"ProductVersion":"8.8.1","PostingDa
07","SupportedDevices":
["Watch3,1","Watch3,2","Watch3,3","Watch3,4","Watch4,1","Watch4,2","Watch4,3","Watch4,4","Watch5,1","Watch5,10","Watch5,11","Watch5,12",
6,1","Watch6,2","Watch6,3","Watch6,4","Watch6,6","Watch6,7","Watch6,8","Watch6,9"]},{"ProductVersion":"9.5.2","PostingDate":"2023-06-21"
["Watch4,1","Watch4,2","Watch4,3","Watch4,4","Watch5,1","Watch5,10","Watch5,11","Watch5,12","Watch5,2","Watch5,3","Watch5,4","Watch5,9",
tch6,13","Watch6,14","Watch6,15","Watch6,16","Watch6,17","Watch6,18","Watch6,2","Watch6,3","Watch6,4","Watch6,6","Watch6,7","Watch6,8","
{"ProductVersion":"16.5","PostingDate":"2023-05-18","ExpirationDate":"2023-10-07","SupportedDevices":
["AppleTV11,1","AppleTV14,1","AppleTV5,3","AppleTV6,2","AudioAccessory1,1","AudioAccessory1,2","AudioAccessory5,1","AudioAccessory6,1"]}
12-13","ExpirationDate":"2023-10-07","SupportedDevices":
["iPad11,1","iPad11,2","iPad11,3","iPad11,4","iPad11,6","iPad11,7","iPad12,1","iPad12,2","iPad13,1","iPad13,10","iPad13,11","iPad13,16",
13,6","iPad13,7","iPad13,8","iPad13,9","iPad14,1","iPad14,2","iPad6,11","iPad6,12","iPad6,3","iPad6,4","iPad6,7","iPad6,8","iPad7,1","iP
iPad7,5","iPad7,6","iPad8,1","iPad8,10","iPad8,11","iPad8,12","iPad8,2","iPad8,3","iPad8,4","iPad8,5","iPad8,6","iPad8,7","iPad8,8","iPa
[{"ProductVersion":"1.0","PostingDate":"2023-06-16","ExpirationDate":"2023-09-14","SupportedDevices":["RealityDevice14,1"]}],"macOS":[{"
28","ExpirationDate":"2023-10-07","SupportedDevices":
["J132AP","J137AP","J140AAP","J140KAP","J152FAP","J160AP","J174AP","J185AP","J185FAP","J213AP","J214AP","J214KAP","J215AP","J223AP","J23
,"J457AP","J680AP","J780AP","Mac-06F11F11946D27C5","Mac-06F11FD93F0323C5","Mac-0CFF9C7C2B63DF8D","Mac-112818653D3AABFC","Mac-112B0A653D3
1F7F29AD0135F9BC","Mac-226CB3C6A851A671","Mac-27AD2F918AE68F61","Mac-2BD1B31983FE1663","Mac-35C1E88140C3E6CF","Mac-35C5E08120C7EEAF","Ma
```

ScheduleOSUpdate response

https://developer.apple.com/documentation/devicemanagement/scheduleosupdateresponse

- Acknowledged: The device processed the command successfully.
- Error: An error occurred. See the ErrorChain for more details.
- CommandFormatError: A protocol error occurred, which can result from a malformed command.
- Idle: The device is idle; there’s no status.
- NotNow: The device received the command, but couldn’t execute it.

ScheduleOSUpdate response

https://developer.apple.com/documentation/devicemanagement/scheduleosupdateresponse

- Acknowledged: The device processed the command successfully.
- Error: An error occurred. See the ErrorChain for more details.
- CommandFormatError: A protocol error occurred, which can result from a malformed command.
- Idle: The device is idle; there’s no status.
- NotNow: The device received the command, but couldn’t execute it.

Reviewing responses

does your MDM expose logs?

- Debug mode
- Webhooks
- syslog

Unified logs

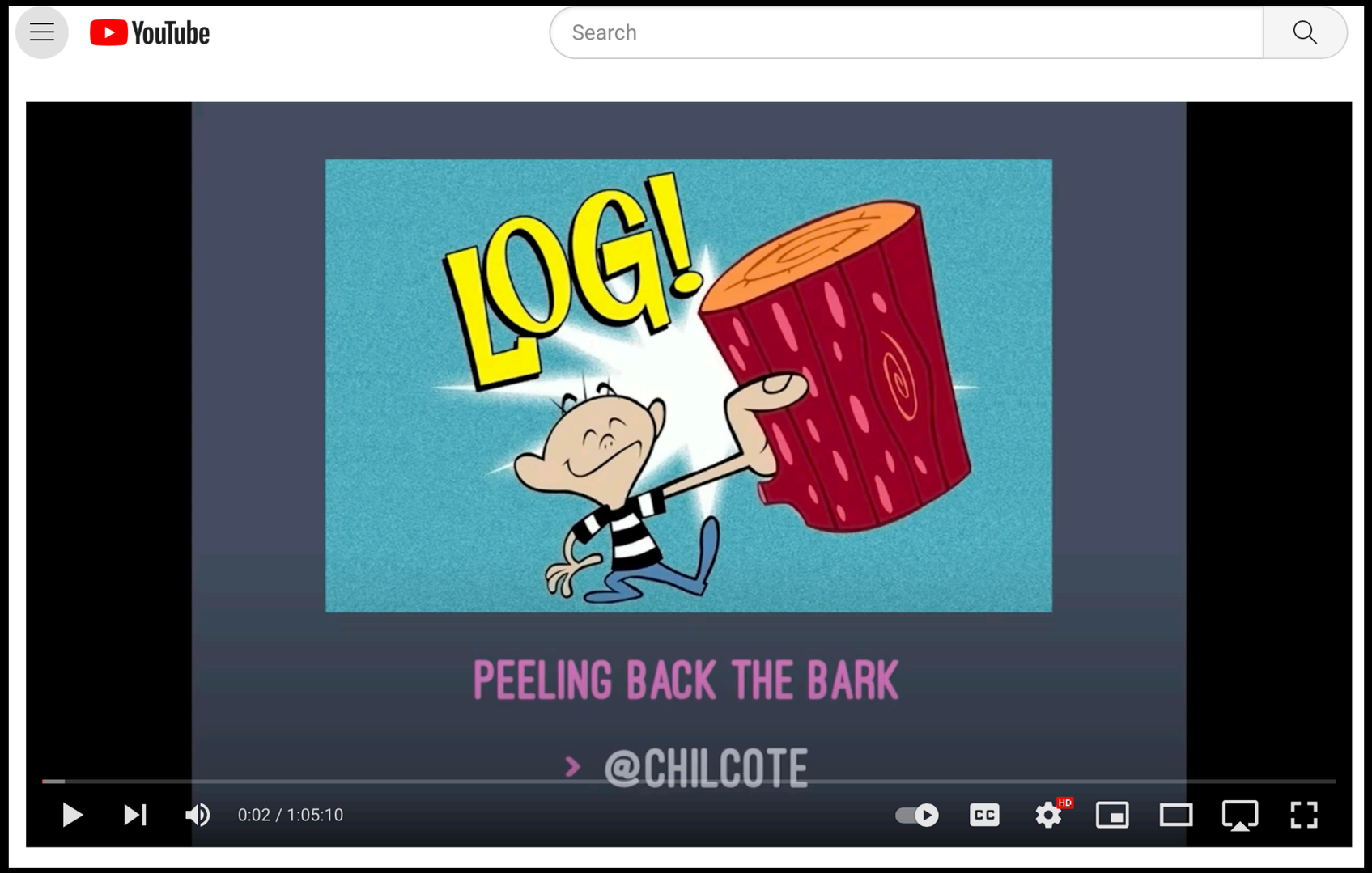
Unified logs

mdmclient tells the tale

- Clients log many of the commands that are queued
- Other subsystems may contain additional information
- User interactions can help troubleshoot issue

Unified logs

oh no I have to re-learn this stuff?



Unified logs

https://blog.kandji.io/mac-logging-and-the-log-command-a-guide-for-apple-admins

- Provide an efficient logging mechanism for both user and kernel mode
- Collect as much data as possible while being performant
- Design privacy into the system
- Interact with the `log` cli tool

Unified logs

don’t cross the streams

- log stream
- log show
- log collect
- log show --archive ./path/to/archive.logarchive

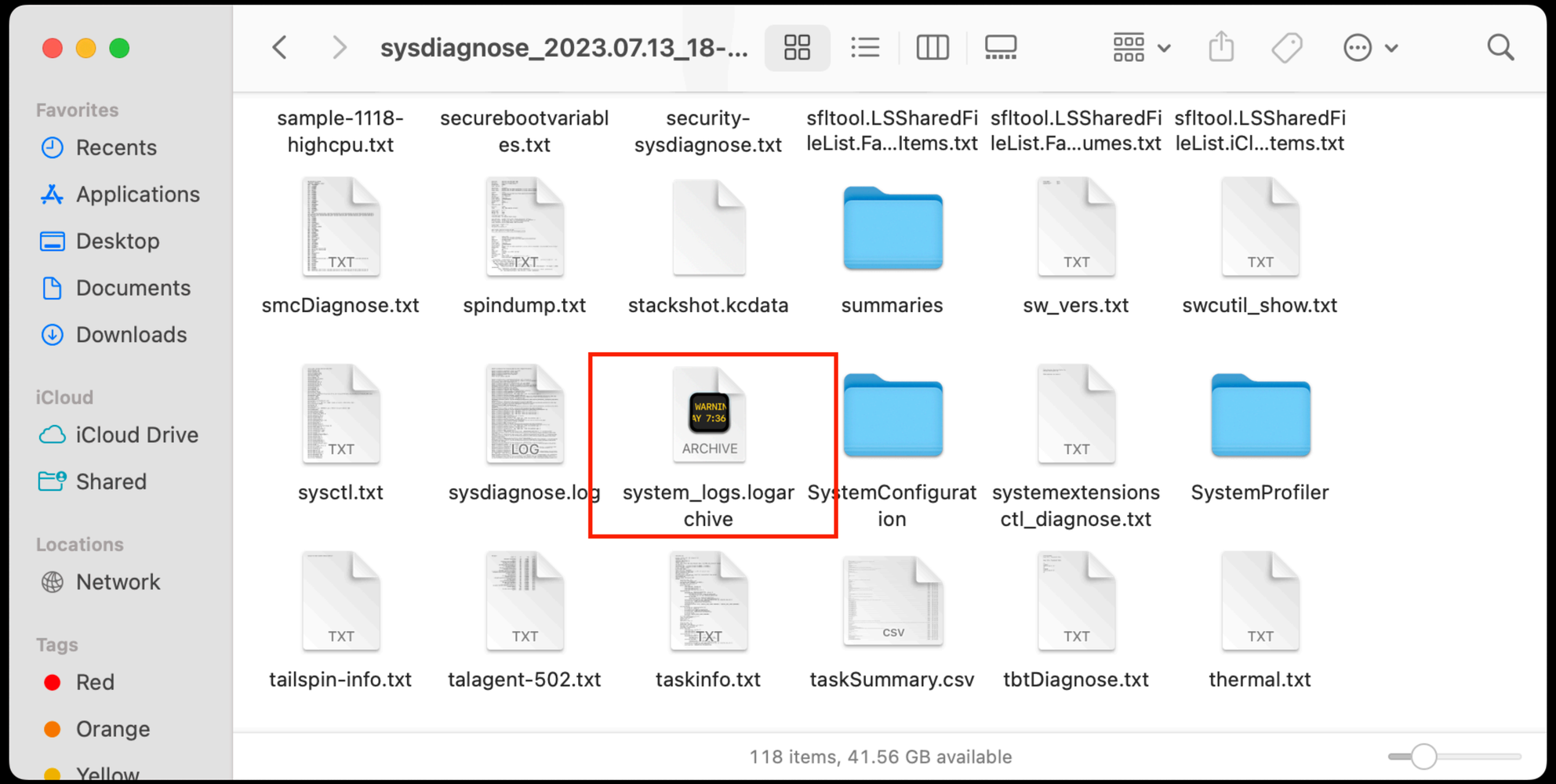
Unified logs

calling collect

- log stream
- log show
- log collect
- log show --archive ./path/to/archive.logarchive

Unified logs

sysdiagnose



Unified logs

`log` syntax

- log stream --info
- log stream --debug
- log show --style {compact, syslog, json}
- log show --predicate 'logType == "error"'

Unified logs

predicates

- process
- subsystem
- category (requires subsystem)
- eventMessage

Unified logs

predicates

- process
- subsystem
- category (requires subsystem)
- eventMessage
- logType (aka messageType)
- processImagePath
- senderImagePath

Unified logs

predicates

2023-07-10 mdmclient[com.apple.ManagedClient:MDMDaemon] InstallProfile com.foo.loginwindow

Unified logs

process

2023-07-10 mdmclient[com.apple.ManagedClient:MDMDaemon] InstallProfile com.foo.loginwindow

Unified logs

subsystem

2023-07-10 mdmclient[com.apple.ManagedClient:MDMDaemon] InstallProfile com.foo.loginwindow

Unified logs

category

2023-07-10 mdmclient[com.apple.ManagedClient:MDMDaemon] InstallProfile com.foo.loginwindow

Unified logs

eventMessage

2023-07-10 mdmclient[com.apple.ManagedClient:MDMDaemon] InstallProfile com.foo.loginwindow

Unified logs

time-based filters

- --last 7d
- --start '2023-07-14 12:00:00'
- --end '2023-07-15 00:00:00'

Unified logs example

tcc

```
log show --info --debug --predicate 'subsystem == "com.apple.TCC"
&& eventMessage BEGINSWITH[cd] "AttributionChain"'
```

Unified logs example

anyconnect

```
log show --info --debug --predicate 'subsystem IN { "com.cisco.anyconnect.vpn",
"com.cisco.secureclient.vpn" } && category IN { "acvpnagent", "acvpnui", "acvpncli",
"csc_vpnagent", "csc_ui", "csc_vpncli" } && ((eventMessage CONTAINS[cd] "has been requested
by the user" || eventMessage CONTAINS[cd] "contacting" || eventMessage CONTAINS[cd]
"initiating vpn connection" || eventMessage CONTAINS[cd] "system suspend" || eventMessage
CONTAINS[cd] "system resume" || eventMessage CONTAINS[cd] "reconnect" || eventMessage
CONTAINS[cd] "connected to" || eventMessage CONTAINS[cd] "vpn connection has been" ||
eventMessage CONTAINS[cd] "vpn connection is being" || eventMessage CONTAINS[cd]
"connection terminated" || eventMessage CONTAINS[cd] "user has requested to disconnect" ||
eventMessage CONTAINS[cd] "vpn state" || eventMessage CONTAINS[cd] "reconfigure" ||
eventMessage CONTAINS[cd] "correction of the routing table has failed" || eventMessage
CONTAINS[cd] "reason code" || eventMessage CONTAINS[cd] "HTTP response") ||
(eventMessage CONTAINS[cd] "function:" && messageType = "error"))' --style compact --last
30m
```

Unified logs example

anyconnect

```
log show --info --debug --predicate 'subsystem IN { "com.cisco.anyconnect.vpn",
"com.cisco.secureclient.vpn" } && category IN { "acvpnagent", "acvpnui", "acvpncli",
"csc_vpnagent", "csc_ui", "csc_vpncli" } && ((eventMessage CONTAINS[cd] "has been requested
by the user" || eventMessage CONTAINS[cd] "contacting" || eventMessage CONTAINS[cd]
"initiating vpn connection" || eventMessage CONTAINS[cd] "system suspend" || eventMessage
CONTAINS[cd] "system resume" || eventMessage CONTAINS[cd] "reconnect" || eventMessage
CONTAINS[cd] "connected to" || eventMessage CONTAINS[cd] "vpn connection has been" ||
eventMessage CONTAINS[cd] "vpn connection is being" || eventMessage CONTAINS[cd]
"connection terminated" || eventMessage CONTAINS[cd] "user has requested to disconnect" ||
eventMessage CONTAINS[cd] "vpn state" || eventMessage CONTAINS[cd] "reconfigure" ||
eventMessage CONTAINS[cd] "correction of the routing table has failed" || eventMessage
CONTAINS[cd] "reason code" || eventMessage CONTAINS[cd] "HTTP response") ||
(eventMessage CONTAINS[cd] "function:" && messageType = "error"))' --style compact --last
30m
```

Unified logs example

anyconnect

```
log show --info --debug --predicate 'subsystem IN { "com.cisco.anyconnect.vpn",
"com.cisco.secureclient.vpn" } && category IN { "acvpnagent", "acvpnui", "acvpncli",
"csc_vpnagent", "csc_ui", "csc_vpncli" } && ((eventMessage CONTAINS[cd] "has been requested
by the user" || eventMessage CONTAINS[cd] "contacting" || eventMessage CONTAINS[cd]
"initiating vpn connection" || eventMessage CONTAINS[cd] "system suspend" || eventMessage
CONTAINS[cd] "system resume" || eventMessage CONTAINS[cd] "reconnect" || eventMessage
CONTAINS[cd] "connected to" || eventMessage CONTAINS[cd] "vpn connection has been" ||
eventMessage CONTAINS[cd] "vpn connection is being" || eventMessage CONTAINS[cd]
"connection terminated" || eventMessage CONTAINS[cd] "user has requested to disconnect" ||
eventMessage CONTAINS[cd] "vpn state" || eventMessage CONTAINS[cd] "reconfigure" ||
eventMessage CONTAINS[cd] "correction of the routing table has failed" || eventMessage
CONTAINS[cd] "reason code" || eventMessage CONTAINS[cd] "HTTP response") ||
(eventMessage CONTAINS[cd] "function:" && messageType = "error"))' --style compact --last
30m
```

Unified logs example

anyconnect

```
log show --info --debug --predicate 'subsystem IN { "com.cisco.anyconnect.vpn",
"com.cisco.secureclient.vpn" } && category IN { "acvpnagent", "acvpnui", "acvpncli",
"csc_vpnagent", "csc_ui", "csc_vpncli" } && ((eventMessage CONTAINS[cd] "has been requested
by the user" || eventMessage CONTAINS[cd] "contacting" || eventMessage CONTAINS[cd]
"initiating vpn connection" || eventMessage CONTAINS[cd] "system suspend" || eventMessage
CONTAINS[cd] "system resume" || eventMessage CONTAINS[cd] "reconnect" || eventMessage
CONTAINS[cd] "connected to" || eventMessage CONTAINS[cd] "vpn connection has been" ||
eventMessage CONTAINS[cd] "vpn connection is being" || eventMessage CONTAINS[cd]
"connection terminated" || eventMessage CONTAINS[cd] "user has requested to disconnect" ||
eventMessage CONTAINS[cd] "vpn state" || eventMessage CONTAINS[cd] "reconfigure" ||
eventMessage CONTAINS[cd] "correction of the routing table has failed" || eventMessage
CONTAINS[cd] "reason code" || eventMessage CONTAINS[cd] "HTTP response") ||
(eventMessage CONTAINS[cd] "function:" && messageType = "error"))' --style compact --last
30m
```

Case Study

“It was written I should be loyal to the nightmare of my choice.”
— Joseph Conrad, Heart of Darkness

Software update logs

where to start?

```
log show --info --debug --predicate 'process = "softwareupdated"' --last 7d
```

Software update logs

where to start?

```
log show --info --debug --predicate 'process = "softwareupdated"' --last 7d
```

38,316 lines returned

Software update logs

structured output

```
log show --info --debug --predicate 'process = "softwareupdated"' --style json
```

Software update logs

structured output

log show --info --debug --predicate 'process = "softwareupdated"' **--style json**


```
{
 "messageType" : "Default",
  "eventType" : "logEvent",
  "subsystem" : "com.apple.SoftwareUpdateMacController",
  "category" : "SU",
  "processImagePath" : "\/System\/Library\/CoreServices\/Software Update.app\/Contents\/Resources\/softwareupdated",
  "senderImagePath" : "\/System\/Library\/PrivateFrameworks\/SoftwareUpdateCoreSupport.framework\/Versions\/A\/SoftwareUpdateCoreSupport",
  "timestamp" : "2023-07-14 15:59:53.095635-0700",
  "eventMessage" : "Big fail",
}
```

Software update logs

structured output

log show --info --debug --predicate 'process = "softwareupdated"' **--style json**


```
{
 "messageType" : "Default",
  "eventType" : "logEvent",
  "subsystem" : "com.apple.SoftwareUpdateMacController",
  "category" : "SU",
  "processImagePath" : "/System/Library/CoreServices/Software Update.app/Contents/Resources/softwareupdated",
  "senderImagePath" : "/System/Library/PrivateFrameworks/SoftwareUpdateCoreSupport.framework/Versions/A/SoftwareUpdateCoreSupport",
  "timestamp" : "2023-07-14 15:59:53.095635-0700",
  "eventMessage" : "Big fail",
}
```

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug --predicate 'process = “softwareupdated"' --style json \
  | ./logreport --list-subsystems
```

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug --predicate 'process = “softwareupdated"' --style json \
  | ./logreport --list-subsystems
```

com.apple.mac.install
com.apple.RemoteServiceDiscovery
com.apple.AssetCacheServices
com.apple.CFNetwork
com.apple.Foundation
com.apple.su
com.apple.BootPolicy
com.apple.networkextension
com.apple.SoftwareUpdateMacController
com.apple.CoreAnalytics
com.apple.xpc
com.apple.launchservices
com.apple.xpc.activity
com.apple.SkyLight
com.apple.mobileassetd
com.apple.network
com.apple.defaults
com.apple.SoftwareUpdate
com.apple.MobileSoftwareUpdate

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug --predicate 'process = “softwareupdated"' --style json \
  | ./logreport --list-subsystems
```

com.apple.mac.install
com.apple.RemoteServiceDiscovery
com.apple.AssetCacheServices
com.apple.CFNetwork
com.apple.Foundation
com.apple.su
com.apple.BootPolicy
com.apple.networkextension
com.apple.SoftwareUpdateMacController
com.apple.CoreAnalytics
com.apple.xpc
com.apple.launchservices
com.apple.xpc.activity
com.apple.SkyLight
com.apple.mobileassetd
com.apple.network
com.apple.defaults
com.apple.SoftwareUpdate
com.apple.MobileSoftwareUpdate

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug \
--predicate 'subsystem = "com.apple.SoftwareUpdateMacController"' \
--style json \
  | ./logreport --list-categories
```

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug \
--predicate 'subsystem = "com.apple.SoftwareUpdateMacController"' \
--style json \
  | ./logreport --list-categories
```

RosettaManager

RecoveryOSManager

SU

SUMacControllerClient

SERVER-com.apple.sumaccontroller

SUMacController

CLIENT

ScanManager

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug \
--predicate 'subsystem = "com.apple.SoftwareUpdateMacController"' \
--style json \
  | ./logreport --list-categories
```

RosettaManager

RecoveryOSManager

SU

SUMacControllerClient

SERVER-com.apple.sumaccontroller

SUMacController

CLIENT

ScanManager

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug \
--predicate '(subsystem == "com.apple.SoftwareUpdateMacController" && \
category == "SU" && \
(eventMessage CONTAINS[cd] "Sending event" && \
eventMessage CONTAINS[cd] “ota")'
```

Software update logs

GitHub / chilcote / logreport

```
log show --info --debug \
--predicate '(subsystem == "com.apple.SoftwareUpdateMacController" && \
category == "SU" && \
(eventMessage CONTAINS[cd] "Sending event" && \
eventMessage CONTAINS[cd] “ota")'
```

2023-07-14 13:54:10.356 Df softwareupdated [com.apple.SoftwareUpdateMacController:SU] [SUMacControllerEventReporter] Sending event (**otaAvailable**): {

UUID = "93ACDFF0-BD23-45A4-9807-519751379B78";

clientAccessControlPriority = ClientInitiated;

clientCommand = SUMacControllerCommandScan;

commandLine = false;

configUserInitiated = true;

descriptor = "SUMacControllerDescriptor(UUID:16FADDB1-9B8A-4980-B6F0-B78996C60752 SU:macOS Ventura 13.4.1 22F82 (Customer)(SU) SFR:macOS 13.4.1 22F82 (Customer) BRAIN:22F82 BRIDGEOS:NO((null)) ROSETTA:YES((null)) RECOVERYOS:YES)";

installTonight = true;

installedRecoveryOSBuild = 22E261;

installedRecoveryOSVersion = "13.3.1";

mdm = false;

notification = false;

targetOSVersion = 22F82;

updateType = incremental;

}

2023-07-14 13:54:10.356 Df softwareupdated [com.apple.SoftwareUpdateMacController:SU] [SUMacControllerEventReporter] Sending event (**otaAvailable**): {

UUID = "93ACDFF0-BD23-45A4-9807-519751379B78";

clientAccessControlPriority = ClientInitiated;

clientCommand = SUMacControllerCommandScan;

commandLine = false;

configUserInitiated = true;

descriptor = "SUMacControllerDescriptor(UUID:16FADDB1-9B8A-4980-B6F0-B78996C60752 SU:macOS Ventura 13.4.1 22F82 (Customer)(SU) SFR:macOS 13.4.1 22F82 (Customer) BRAIN:22F82 BRIDGEOS:NO((null)) ROSETTA:YES((null)) RECOVERYOS:YES)";

installTonight = true;

installedRecoveryOSBuild = 22E261;

installedRecoveryOSVersion = "13.3.1";

mdm = false;

notification = false;

targetOSVersion = 22F82;

updateType = incremental;

}

2023-07-14 13:54:14.509 Df softwareupdated [com.apple.SoftwareUpdateMacController:SU] [SUMacControllerEventReporter] Sending event (**otaAvailable**): {

UUID = "93ACDFF0-BD23-45A4-9807-519751379B78";

clientAccessControlPriority = ClientInitiated;

clientCommand = SUMacControllerCommandScan;

commandLine = false;

configUserInitiated = true;

descriptor = "SUMacControllerDescriptor(UUID:10B630E4-70F4-4CA2-8ACC-1C85C82A965C SU:macOS Ventura 13.4.1 22F82 (Customer)(SU) SFR:macOS 13.4.1 22F82 (Customer) BRAIN:22F82 BRIDGEOS:NO((null)) ROSETTA:YES((null)) RECOVERYOS:YES)";

installTonight = false;

installedRecoveryOSBuild = 22E261;

installedRecoveryOSVersion = "13.3.1";

mdm = true;

notification = true;

targetOSVersion = 22F82;

updateType = incremental;

}

OTA Logs

“over the air”

- otaAvailable
- otaDownloaded
- otaReady
- otaInstalling
- otaAbandonded

Software update log examples

updateFinished

- subsystem == "com.apple.MobileSoftwareUpdate" AND category == "Info" AND eventMessage CONTAINS[cd] “updateFinished"

2023-07-11 23:42:08.765 I softwareupdated[85533:7c34d] [com.apple.MobileSoftwareUpdate:Info] 1eba23280 : Attempting to gather analytics data for phase : UpdateFinished

Software Update log examples

SoftwareUpdateNotificationManager

Received notification response w/ request identifier: com.apple.SoftwareUpdateNotificationManager.RestartCountdown, action identifier: RESTART, source: RestartCountdown, type: (null), userInfo: {

SecondsRemaining = 52;

Source = RestartCountdown;

}

Software Update log examples

SoftwareUpdateNotificationManager

Received notification response w/ request identifier: com.apple.SoftwareUpdateNotificationManager.RestartCountdown, action identifier: RESTART, source: RestartCountdown, type: (null), userInfo: {

SecondsRemaining = 52;

Source = RestartCountdown;

}

SUOSUNotificationCenterDelegate: Restart countdown: user clicked restart

Software Update log examples

SoftwareUpdateNotificationManager

Received notification response w/ request identifier: com.apple.SoftwareUpdateNotificationManager.RestartCountdown, action identifier: RESTART, source: RestartCountdown, type: (null), userInfo: {

SecondsRemaining = 52;

Source = RestartCountdown;

}

SUOSUNotificationCenterDelegate: Restart countdown: user dismissed the notification

Software update log examples

BootstrapToken

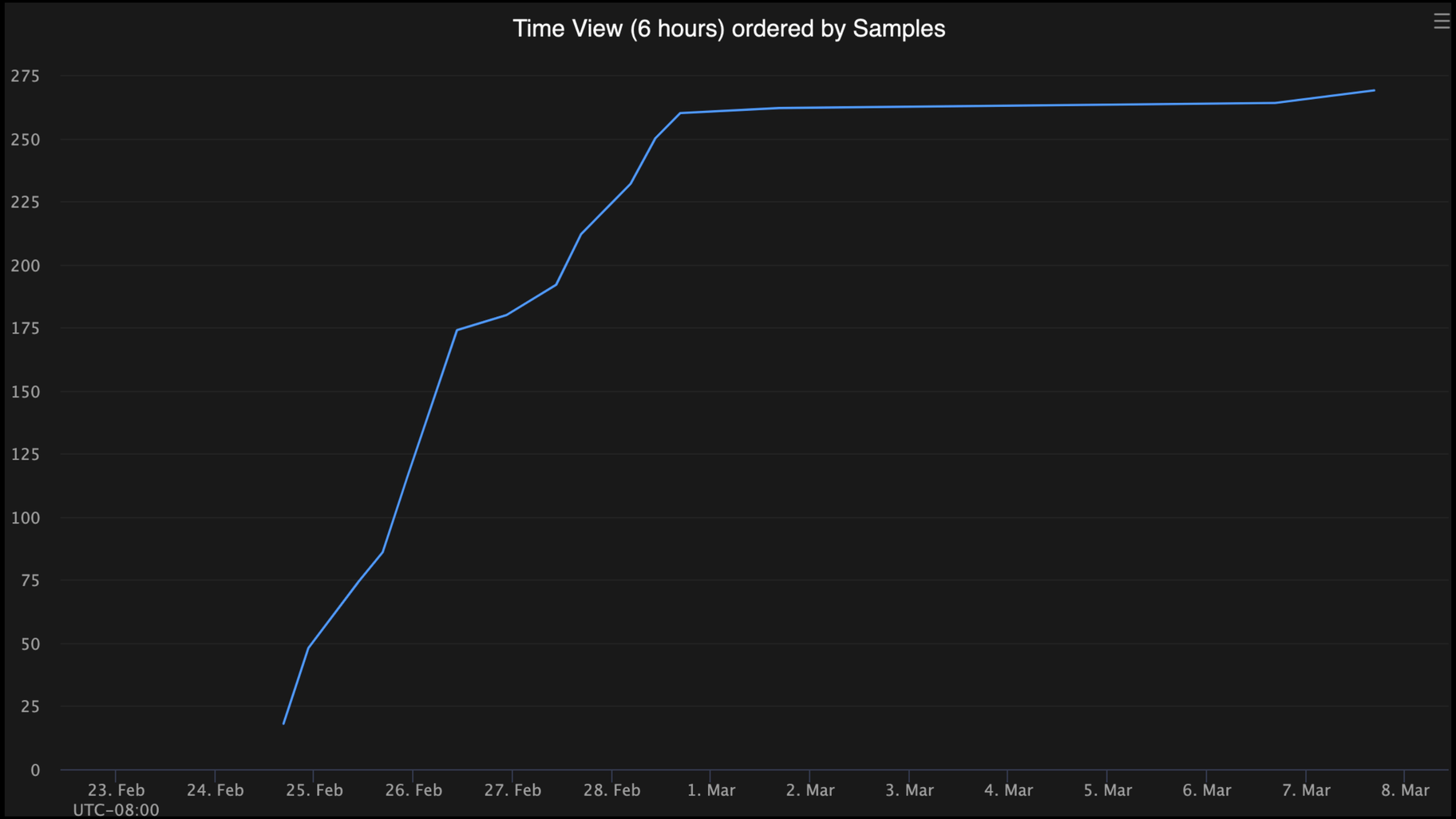
- subsystem == "com.apple.ManagedClient" && category = "HTTPUtil" && eventMessage CONTAINS[cd] "BootstrapToken"

2023-07-13 18:10:04.740 Df mdmclient[895:4a73] [com.apple.ManagedClient:HTTPUtil] [0:MDMDaemon:HTTPUtil:<0x4a73>] >>>>> Sending HTTP request (PUT) [MDM_SetBootstrapToken] >>>>>

2023-07-13 18:10:04.983 Df mdmclient[895:4a73] [com.apple.ManagedClient:HTTPUtil] [0:MDMDaemon:HTTPUtil:<0x4a73>] <<<<< Received HTTP response (200) [MDM_SetBootstrapToken] <<<<<

Software Update log examples

BootstrapToken



Bring it home

You think you're bored? Imagine rehearsing this talk!

Inspecting MDM Commands

server and client side

- MDM Commands and Responses

Inspecting MDM Commands

server and client side

- MDM Commands and Responses
- Software Update command syntax

Inspecting MDM Commands

server and client side

- MDM Commands and Responses
- Software Update command syntax
- Webhooks & Server side logging

Inspecting MDM Commands

server and client side

- MDM Commands and Responses
- Software Update command syntax
- Webhooks & Server side logging
- Unified logs

Kandji example conclusion

Updates

Specify how macOS versions are automatically enforced. Learn more...

Version enforcement

Manually enforce a minimum version

Specify if updates should not be managed, automatically be enforced after they are released, or select a minimum macOS version for enforcement.

⚠ The latest macOS version available at the time of install will be used regardless of these settings.

Minimum version

13.4.1

Specify the minimum macOS version to enforce.

Enforcement deadline

July 14, 2023

(GMT-08:00) Pacific Time - Los Angeles

12:00 PM (Noon)

Specify when the macOS version should be enforced. The update will be cached and users will be able to voluntarily update five (5) days before enforcement deadline, or sooner if a shorter time frame is selected.

Kandji example conclusion

Processing server request: ScheduleOSUpdateScan

Processing server request: AvailableOSUpdates

Kandji example conclusion

Processing server request: ScheduleOSUpdateScan

Processing server request: AvailableOSUpdates

```
MDM requested: {
    InstallAction = DownloadOnly;
    Priority = High;
    ProductKey = "MSU_UPDATE_22F82_patch_13.4.1_minor";
    ProductVersion = "13.4.1";
}
```

Kandji example conclusion

Processing server request: ScheduleOSUpdateScan

Processing server request: AvailableOSUpdates

```
MDM requested: {
    InstallAction = Default;
    Priority = High;
    ProductKey = "MSU_UPDATE_22F82_patch_13.4.1_minor";
    ProductVersion = "13.4.1";
}
```

Kandji example conclusion

InstallAction: Default options: {

```
    BootstrapToken = "<NON-EMPTY STRING>";

    DoInForeground = 1;

    MDMInitiated = 1;

    SkipFLO = 1;
}
```

Kandji example conclusion

```
UpdateStatus 'MSU_UPDATE_22F82_patch_13.4.1_minor' ==> {
    phase = downloading;
    productKeys =     (
        "MSU_UPDATE_22F82_patch_13.4.1_minor"
    );
    progress = "91.80247783660889";
}
```

Kandji example conclusion

```
UpdateStatus 'MSU_UPDATE_22F82_patch_13.4.1_minor' ==> {
    phase = downloaded;
    productKeys =     (
        "MSU_UPDATE_22F82_patch_13.4.1_minor"
    );
    progress = 100;
}
```

Kandji example conclusion

SUOSUCountdownNotification: seconds remaining: 9

SUOSUCountdownNotification: seconds remaining: 8

SUOSUCountdownNotification: seconds remaining: 7

SUOSUCountdownNotification: seconds remaining: 6

SUOSUCountdownNotification: seconds remaining: 5

SUOSUCountdownNotification: seconds remaining: 4

SUOSUCountdownNotification: seconds remaining: 3

SUOSUCountdownNotification: seconds remaining: 2

SUOSUCountdownNotification: seconds remaining: 1

SUOSUCountdownNotification: Remove restart countdown!

Kandji example conclusion

SUOSUCountdownNotification: seconds remaining: 9

SUOSUCountdownNotification: seconds remaining: 8

SUOSUCountdownNotification: seconds remaining: 7

SUOSUCountdownNotification: seconds remaining: 6

SUOSUCountdownNotification: seconds remaining: 5

SUOSUCountdownNotification: seconds remaining: 4

SUOSUCountdownNotification: seconds remaining: 3

SUOSUCountdownNotification: seconds remaining: 2

SUOSUCountdownNotification: seconds remaining: 1

SUOSUCountdownNotification: Remove restart countdown!

(updateFinished)

JumpCloud example conclusion

OS Updates

OS	13.3.1 (22E261)
Total Number of Versions Available	1
Last Scanned for Updates	07-14-2023 at 12:38pm

Select an OS Update to Schedule

Available Updates

macOS Ventura 13.4.1 (Minor) 22F82

Install Action

Install Later Today

Schedule...

JumpCloud example conclusion

```
MDM requested: {
    InstallAction = InstallLater;
    MaxUserDeferrals = 1;
    ProductKey = "MSU_UPDATE_22F82_patch_13.4.1_minor";
}
```

JumpCloud example conclusion

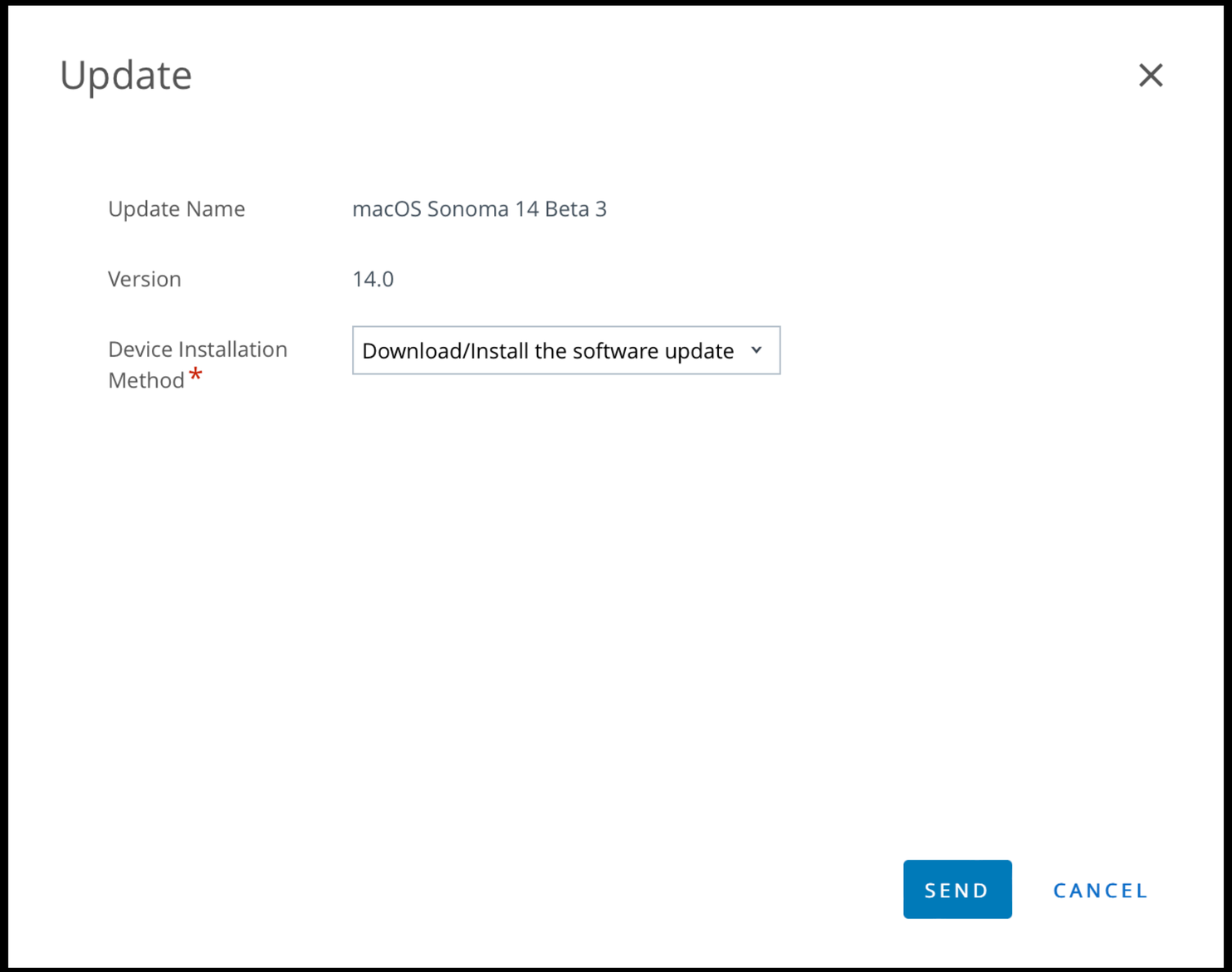
SUOSUNotificationCenterDelegate: User dismissed the notification com.apple.SoftwareUpdateNotificationManager.UpdatesAvailableOfferLater

JumpCloud example conclusion

SUOSUNotificationCenterDelegate: User dismissed the notification com.apple.SoftwareUpdateNotificationManager.UpdatesAvailableOfferLater

SUOSUNotificationCenterDelegate: Schedule MDM tonight action

Workspace ONE conclusion



Workspace ONE conclusion

```
MDM requested: {
    InstallAction = InstallLater;
    MaxUserDeferals = 3;
    ProductKey = "MSU_UPDATE_23A5286i_patch_14.0_minor";
    ProductVersion = "14.0";
}
```

Workspace ONE conclusion

```
MDM initiated, scanning for update with options: {
    BootstrapToken = Exists;
    DoInForeground = 0;
    MDMInitiated = 1;
    ProductKeys =     (
        "MSU_UPDATE_23A5286i_patch_14.0_minor"
    );
    ScheduleUpdateForLater = 1;
}
```

Workspace ONE conclusion

```
Sending event (otaAbandoned): {
   event = otaAbandoned;
    failureReason = "[SUMacControllerError:7749]";
    result = "SUMacControllerError - 7749 (SUMacControllerErrorCommitStashInvalidState)";
}
```

Workspace ONE conclusion

```
UpdateStatus 'MSU_UPDATE_23A5286i_patch_14.0_minor' ==> {
    DoItLaterScheduledDate = "2023-07-12 09:00:00 +0000";
    phase = "not running";
    progress = 0;
}
```

Workspace ONE conclusion

Prefetching Bootstrap Token for SoftwareUpdate

```
>>>>> Sending HTTP request (PUT) [MDM_GetBootstrapToken] >>>>>

<<<<< Received HTTP response (200) [MDM_GetBootstrapToken] <<<<<

MDM requested: {
    InstallAction = Default;
    ProductKey = "MSU_UPDATE_23A5286i_patch_14.0_minor";
}
```

Workspace ONE conclusion

Sending event (otaDownloaded)

Sending event (otaReady)

Workspace ONE conclusion

Sending event (otaDownloaded)

Sending event (otaReady)

SUOSUCountdownNotification: seconds remaining: 1

SUOSUCountdownNotification: Remove restart countdown!

Sending event (otaInstalling)

Workspace ONE conclusion

Sending event (otaDownloaded)

Sending event (otaReady)

SUOSUCountdownNotification: seconds remaining: 1

SUOSUCountdownNotification: Remove restart countdown!

Sending event (otaInstalling)

(updateFinished)

SimpleMDM conclusion

Update OS Version

This will send an OS update command to the selected devices.

OS Update version

✓ 12.6.7
13.4.1

OS Update Mode

- (●) Smart Update (Install Later)
  Install the update when macOS deems it to be an opportune time.
- ( ) Notify Only
  Download the software update and notify the user through the App Store.
- ( ) Download Only
  Download the software update without installing it.
- ( ) Install As Soon As Possible
  When the OS update has finished downloading, the user will be prompted and shown a timer before a restart occurs.
- ( ) Force Update
  Download and install the update immediately.

Allowed deferrals

3

Cancel | Update Devices

SimpleMDM conclusion

```
MDM requested: {
    InstallAction = NotifyOnly;
    ProductVersion = "13.4.1";
}
```

SimpleMDM conclusion

```
mdmUpdateStatus: {
    error = "No updates were selected.";
    phase = failed;
    productKeys =     (
    );
    productMarketingVersion = "13.4.1";
    progress = 0;
}
```

SimpleMDM conclusion

```
SUOSUNotificationUpdateService: Available updates did change: (
    "<SUOSUProduct: MSU_UPDATE_22F82_patch_13.4.1_minor>"
)
```

SimpleMDM conclusion

```
InstallASAP options: {
    BootstrapToken = "<NON-EMPTY STRING>";
    DoInForeground = 0;
    MDMInitiated = 1;
    SkipFLO = 1;
}
```

SimpleMDM conclusion

InstallASAP options: {

BootstrapToken = "<NON-EMPTY STRING>";

DoInForeground = 0;

MDMInitiated = 1;

SkipFLO = 1;

}

SUOSURestartCountdownOperation: Starting restart countdown

Skipping restart notification, immediately going down for the reboot

The End

Thanks:
- Greg Neagle
- Mike Boylan
- Tom Bridge
- Chris Dawe