Apple General Info

The Apple General Info section serves as a catch-all resource for a wide range of Apple-related information that doesn't fit within specific categories. From tips and tricks for maximizing productivity on your Apple devices to discussions on Apple services, accessories, and general troubleshooting guidance, this section covers it all. Stay up to date with the latest Apple announcements, explore lesser-known features and functionalities, and find solutions to common issues you may encounter. Discover insights and helpful advice that will enhance your overall Apple experience and make the most of your devices and services. The Apple General Info section is your go-to destination for all things Apple beyond the scope of specific topics.

Adapting to Apple's TLS Server Certificate Validity Limits

What

This article provides guidance on adapting to Apple's updated policy regarding the maximum allowed lifetimes of TLS server certificates. Effective from September 1, 2020, 00:00 GMT/UTC, TLS server certificates must have a validity period no greater than 398 days. This policy, part of Apple's efforts to enhance web security, affects TLS server certificates issued from Root CAs preinstalled with iOS, iPadOS, macOS, watchOS, and tvOS.

When/Why

The policy is critical for administrators using FileWave to manage Apple devices. It ensures that device profiles and their associated TLS server certificates comply with the new security standards. Non-compliance results in network and application failures, and can prevent websites from loading on affected Apple devices.

How

To comply with Apple's policy:

1. Certificate Issuance and Renewal: Certificates should be issued with a maximum validity of 397 days to avoid edge case issues.
2. Check Existing Certificates: Certificates issued before September 1, 2020, are not affected by this change. However, their renewal must comply with the 398-day limit.
3. Profile Deployment in FileWave: Ensure all TLS server certificates embedded in profiles for Apple devices meet these validity requirements.
4. Monitoring and Planning: Regularly monitor certificate expiration dates and plan renewals accordingly.

Related Links

- Apple's Certificate Policy Announcement - Details on the TLS server certificate validity limit.
- RFC 5280, Section 4.1.2.5 - Reference for certificate validity period definition.

Digging Deeper

This policy shift reflects a broader move towards enhancing digital security and trustworthiness in online environments. By reducing certificate lifetimes, Apple aims to mitigate risks such as certificate compromise and mis-issuance. For FileWave users, adapting to these new requirements is essential for maintaining secure, reliable, and compliant management of Apple devices across various environments.

Address Stalled MDM Commands

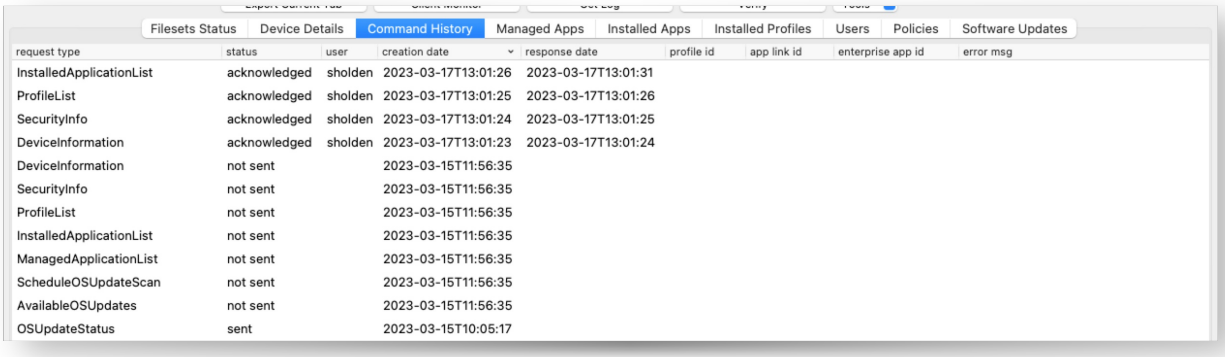
Description

It is possible that device MDM communication can become stalled for macOS, iOS, and Apple TV due to an issue that Apple is working on that impacts all MDM vendors as recently as anything pre-iOS/iPadOS 17.1 and pre-macOS Sonoma 14.1. This can impact all MDM communication, including Reported Issues with macOS Software Updates. If you are experiencing these issues we strongly encourage you to open a ticket with FileWave Customer Technical Support and open an Apple Enterprise support case. If you can share the Apple Enterprise ticket number with FileWave support then we can link the Apple ticket with the FileWave ticket.

When this occurs, the Command History will appear similar to the below image. Commands sent to the 'User' channel (in this example the user is sholden) are acknowledged, however commands sent to the System channel (those that have no user name shown) remain 'not sent'. The reason behind this is related to the MDM Software Update processes stalling on the device.

DeviceInformation Example

For example the `DeviceInformation` command has been acknowledged for the User, but not for the System. In the example, the commands for the System channel were acknowledged over 2 days prior than the acknowledged User channel commands.



Filesets Status | Device Details | Command History | Managed Apps | Installed Apps | Installed Profiles | Users | Policies | Software Updates

request type	status	user	creation date	response date	profile id	app link id	enterprise app id	error msg
InstalledApplicationList	acknowledged	sholden	2023-03-17T13:01:26	2023-03-17T13:01:31				
ProfileList	acknowledged	sholden	2023-03-17T13:01:25	2023-03-17T13:01:26				
SecurityInfo	acknowledged	sholden	2023-03-17T13:01:24	2023-03-17T13:01:25				
DeviceInformation	acknowledged	sholden	2023-03-17T13:01:23	2023-03-17T13:01:24				
DeviceInformation	not sent		2023-03-15T11:56:35					
SecurityInfo	not sent		2023-03-15T11:56:35					
ProfileList	not sent		2023-03-15T11:56:35					
InstalledApplicationList	not sent		2023-03-15T11:56:35					
ManagedApplicationList	not sent		2023-03-15T11:56:35					
ScheduleOSUpdateScan	not sent		2023-03-15T11:56:35					
AvailableOSUpdates	not sent		2023-03-15T11:56:35					
OSUpdateStatus	sent		2023-03-15T10:05:17					

InstalledApplicationList Example

The `InstalledApplicationList` is seen below in this stuck state. You will see that on a device that things will not progress and it will simply hang on this command. We have seen from several customers however that iOS and iPadOS 17.1 do appear to fix this behavior. This is reflected in this note from Apple: What's new for enterprise in iOS 17 - Apple Support and you should investigate if you can get to that version. macOS Sonoma 14.1 also appears to have MDM updates to it as the release notes mention "MDM fails to install enterprise apps after installing a VPP app" for macOS 14.1.

Filesets Status | Device Details | Command History | Managed Apps | Installed Apps | Managed Documents

request type	status	user	creation date	response date
DeviceInformation	not sent		2023-12-07T10:08:53	
SecurityInfo	not sent		2023-12-07T10:08:53	
Restrictions	not sent		2023-12-07T10:08:53	
ProfileList	not sent		2023-12-07T10:08:53	
Restrictions	acknowledged		2023-12-07T10:03:57	2023-12-07T10:03:57
ProfileList	acknowledged		2023-12-07T10:03:57	2023-12-07T10:03:57
DeviceInformation	acknowledged		2023-12-07T10:03:57	2023-12-07T10:03:57
SecurityInfo	acknowledged		2023-12-07T10:03:57	2023-12-07T10:03:57
InstalledApplicationList	sent		2023-12-07T10:03:57	
InstallProfile	not sent		2023-12-07T10:03:57	
ManagedApplicationList	not sent		2023-12-07T10:03:35	
ManagedMediaList	not sent		2023-12-07T10:03:35	
AvailableOSUpdates	not sent		2023-12-07T10:03:35	
OSUpdateStatus	not sent		2023-12-07T10:03:35	
ManagedApplicationConfiguration	not sent		2023-12-07T10:03:35	
ManagedApplicationAttributes	not sent		2023-12-07T10:03:35	
ValidateApplications	acknowledged		2023-12-07T10:00:37	2023-12-07T10:00:37
InstallApplication	acknowledged		2023-12-07T09:51:49	2023-12-07T09:51:51

Workaround for macOS

The following recipe provides a method to built out a setup for monitoring devices that are in stalled state and addressing this with a given Fileset. Note that this workaround can only work for macOS and not iOS, iPadOS or tvOS because you can not run scripts on those other platforms. Rebooting the device is many times the solution for those OS, but updating to the latest release of iOS, iPadOS, and tvOS should resolve this as long as they are capable of getting to 17.1.

> Devices experiencing this state occurs at unknown times. A device that is addressed is likely to experience the same issue after being addresses at an unknown duration of time after. The below process is designed to automatically identify devices when this occurs and as such devices experiencing the issue more than once should still be addressed, on each subsequent experience.

Ingredients

- FW Central
- Two Custom Fields
- Script provided for running on the server
- API authorisation token
- zsh on the server
- Fileset to restart the stalled service on the device

Custom Fields:

MDM Custom Fields.zip

Server script:

fix_mdm_system_channel.sh.zip

Fileset:

FWPS - Kickstart Software Update.fileset.zip

Directions

Creation of Custom Fields

1. Open the Admin console and use the drop down menu ‘Assistants' to select: Custom Fields > Edit Custom Fields
2. Use the Import button to import the two provided Custom Fields

The Custom Fields should already be configured as:

- Administrator
- Date/Time
- Associated to all devices

Server Side Script

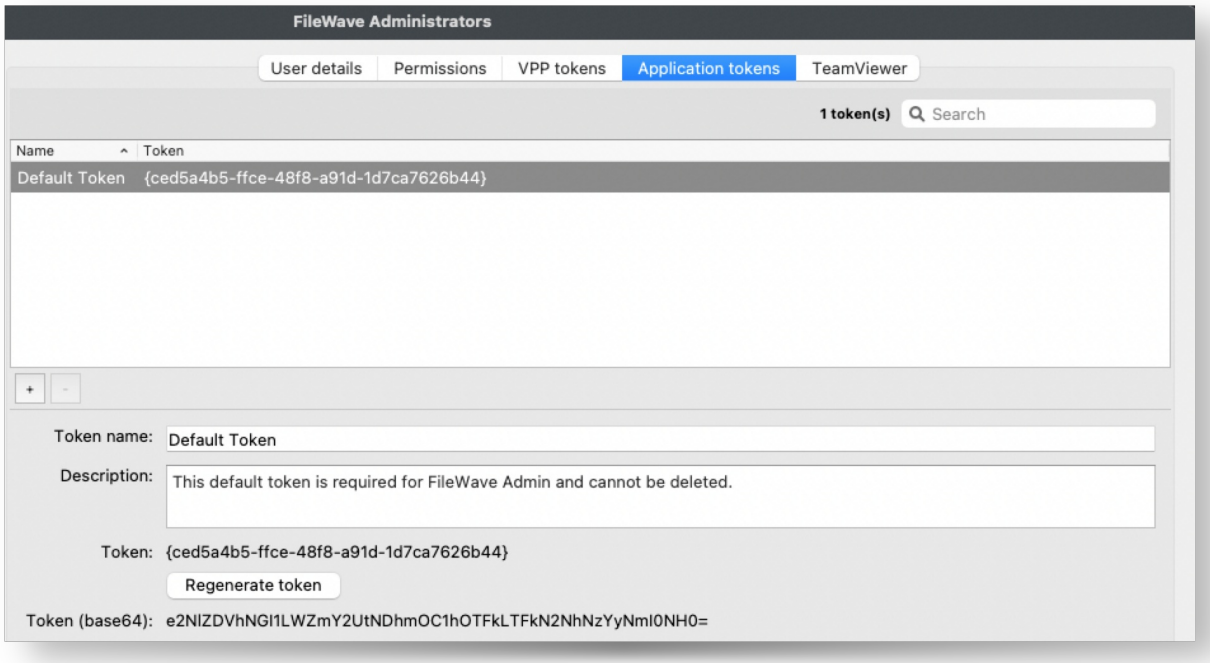
1. Copy the provided Script to the FileWave Server
2. Edit the top of the script, providing the FileWave Administrator Authorisation Token and Server URL values

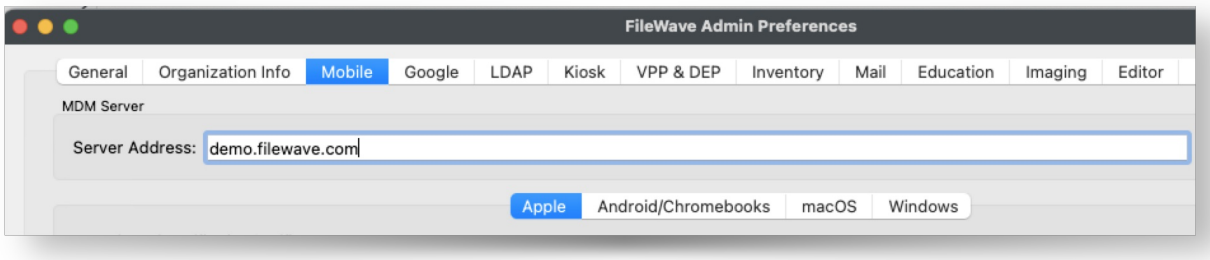
```
#!/bin/zsh

# Source file providing API token and server address
auth=""
server_dns=""

# DO NOT EDIT BELOW THIS LINE
```

Example values:





File edited with above values:

```
#!/bin/zsh

# Source file providing API token and server address
auth="e2NlZDVhNGI1LWZmY2UtNDhmOC1hOTFkLTFkN2NhNzYyNmI0NH0="
server_dns="demo.filewave.com"

# DO NOT EDIT BELOW THIS LINE
```

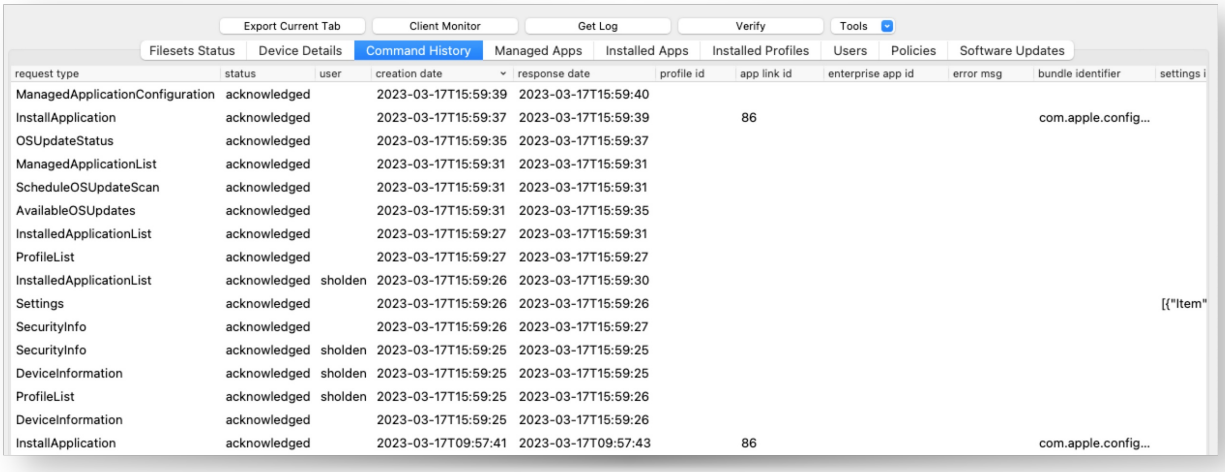
Since this script was written using ZSH, then if not already installed, the ZSH shell will require installing. For macOS servers, ZSH is default, however on CentOS servers it is likely not yet installed. To instal ZSH use the following command:

```
yum install zsh
```

Testing the Script

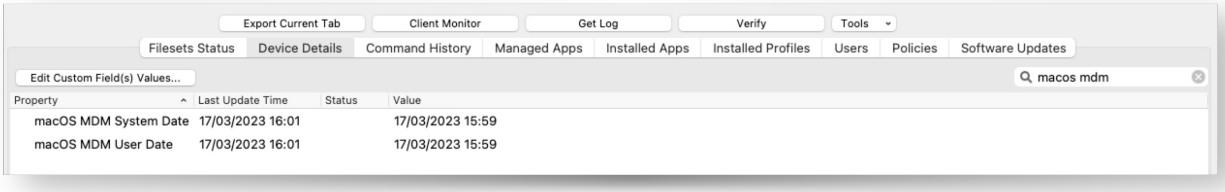
Run the script from the chosen location. On success, each device should now show two dates for the two given Custom Fields. e.g.

Two images, Command History showing the response dates of a machine successfully communicating in the first image:



request type	status	user	creation date	response date	profile id	app link id	enterprise app id	error msg	bundle identifier	settings i
ManagedApplicationConfiguration	acknowledged		2023-03-17T15:59:39	2023-03-17T15:59:40						
InstallApplication	acknowledged		2023-03-17T15:59:37	2023-03-17T15:59:39		86			com.apple.config...	
OSUpdateStatus	acknowledged		2023-03-17T15:59:35	2023-03-17T15:59:37						
ManagedApplicationList	acknowledged		2023-03-17T15:59:31	2023-03-17T15:59:31						
ScheduleOSUpdateScan	acknowledged		2023-03-17T15:59:31	2023-03-17T15:59:31						
AvailableOSUpdates	acknowledged		2023-03-17T15:59:31	2023-03-17T15:59:35						
InstalledApplicationList	acknowledged		2023-03-17T15:59:27	2023-03-17T15:59:31						
ProfileList	acknowledged		2023-03-17T15:59:27	2023-03-17T15:59:27						
InstalledApplicationList	acknowledged	sholden	2023-03-17T15:59:26	2023-03-17T15:59:30						
Settings	acknowledged		2023-03-17T15:59:26	2023-03-17T15:59:26						[{"Item"
SecurityInfo	acknowledged		2023-03-17T15:59:26	2023-03-17T15:59:27						
SecurityInfo	acknowledged	sholden	2023-03-17T15:59:25	2023-03-17T15:59:25						
DeviceInformation	acknowledged	sholden	2023-03-17T15:59:25	2023-03-17T15:59:25						
ProfileList	acknowledged	sholden	2023-03-17T15:59:25	2023-03-17T15:59:26						
DeviceInformation	acknowledged		2023-03-17T15:59:25	2023-03-17T15:59:26						
InstallApplication	acknowledged		2023-03-17T09:57:41	2023-03-17T09:57:43		86			com.apple.config...	

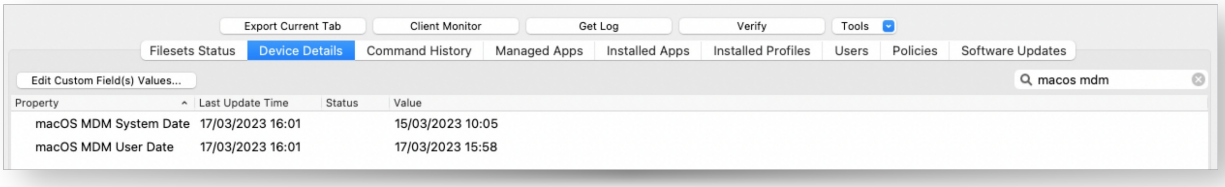
The second image shows the Custom Fields populated with those values, note both date values match:



Property	Last Update Time	Status	Value
macOS MDM System Date	17/03/2023 16:01		17/03/2023 15:59
macOS MDM User Date	17/03/2023 16:01		17/03/2023 15:59

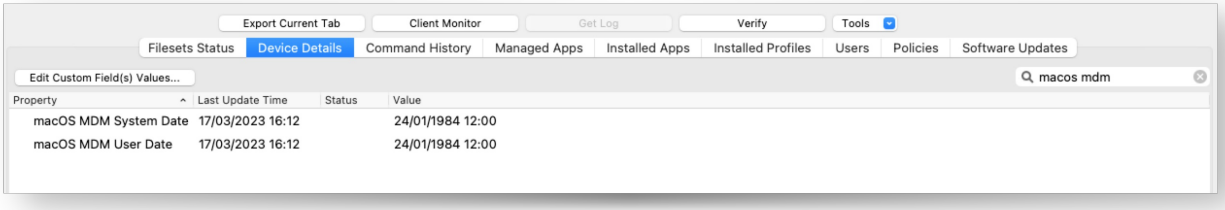
The following image is an example of a device where the device was communicating, but is no longer acknowledging MDM System Channel commands, note the System date value

is more than 2 days older:



Property	Last Update Time	Status	Value
macOS MDM System Date	17/03/2023 16:01		15/03/2023 10:05
macOS MDM User Date	17/03/2023 16:01		17/03/2023 15:58

It is possible that devices may not even have responded yet to a command. If that is the case, then they won't even be any acknowledged commands for either User or System. Where a date is not yet acknowledged, an old date is reported instead. This date may then be used to target these devices also. E.g. note the old date from 24th Jan 1984



Property	Last Update Time	Status	Value
macOS MDM System Date	17/03/2023 16:12		24/01/1984 12:00
macOS MDM User Date	17/03/2023 16:12		24/01/1984 12:00

Cronjob

Once confirmed all is well, a cronjob may be created to action the server script periodically, such that Custom Fields are updated.

- Where devices have been addressed and now working, the script re-run should cause those devices to leave the Smart Group
- Where devices are now having an issue, the script re-run should cause those devices to enter the Smart Group and subsequently receive the Fileset

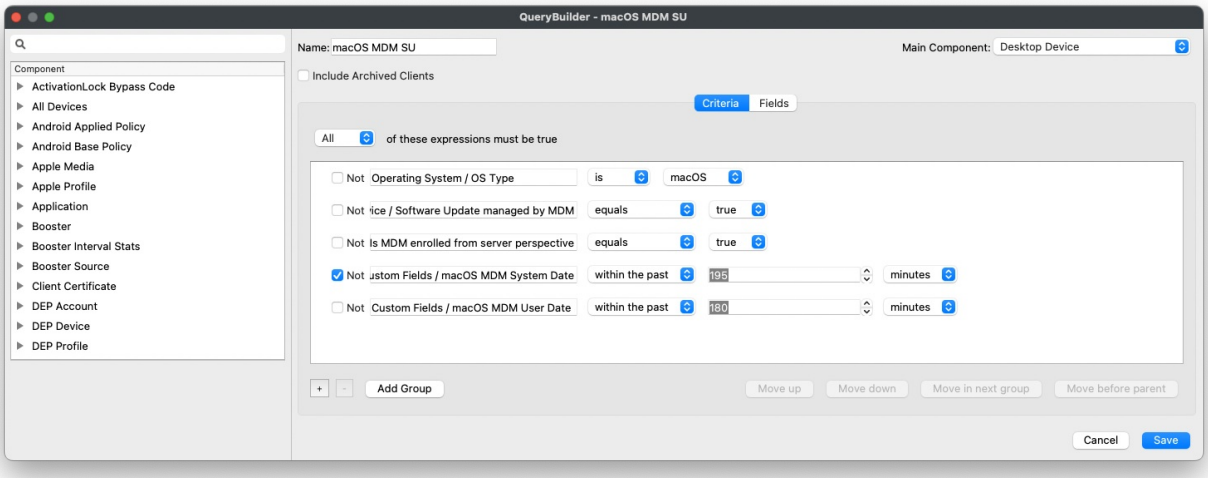
> ⚠ Do not run the cronjob too often. However, in the same right, if a device were addressed and subsequently MDM stalled again between the two cronjob actions, since the device would not have left and re-entered the Smart Group, the Fileset will not reinstall, unless manually re-triggered.

The following links explain how to add a cronjob:

- macOS
- CentOS

Fileset

1. The provided Fileset download should be unzipped and the containing Fileset dragged into the FileWave Admin
2. A Smart Group should be created to associate this Fileset. Example criteria:



On receiving the Fileset, devices should start updating the Command History for both System and User channel.

> Remember, devices will not leave the Smart Group until addressed and then a subsequent run of the server side script is actioned.

Logging

The script will log to the home directory of the user running the script. We would recommend using the root account. The script will keep the logs of 9 prior attempts.

Example entry for a single device:

```
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel ---------------------------
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel Processing Client ID:
152:2e81e79502a54d93823fad08de699eb6c17d47b7
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel Processing User commands
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel System date:
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel Response array:  2023-03-17 15:59:25
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel Processing System channel commands
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel System date:  2023-03-17 15:59:26
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel Response array:  2023-03-17 15:59:25
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel Sending API Patch command with the following...
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel System_date 2023-03-17T15:59:26Z
2023-03-17 16:12:41.***|main|CUSTOM|CLIENT|fix_mdm_system_channel User date 2023-03-17T15:59:25Z
```

Related Content

- https://support.apple.com/en-us/HT213892 - Release notes for iOS/iPadOS 17.1 show a short note mentioning the MDM issue for those platforms.

Apple App Store and Automatic Updates

Developers are frequently updating applications and submitting new versions to Apple App Store. FileWave is automatically sending requests to devices to upgrade applications when a new version is available.

How does FileWave detect if there is an update?

For iOS 11.3 and later and macOS 10.13.4 and later:

Apple's MDM protocol returns the information when FileWave requests the list of installed applications: each application will provide, via the HasUpdateAvailable flag, if an update is available, and in this case, FileWave send the device a command to upgrade the application (InstallApplication command).

For previous OS versions:

Every hour, FileWave is contacting Apple's iTunes database and updates metadata about applications used in your environment. When a device checks in for Verify, we compare the application versions (on the device, from iTunes) and if the version from iTunes is higher we know there is an update.

The automatic update process does not work as expected, what could be the reasons?

If the flag is not provided by the device (older iOS / macOS version for instance), it may happen that the version from iTunes and the version reported by the device are not accurate (they are filled by hand by the developer and they may be incorrect). Apple introduced the hasUpdateAvailable flag exactly to solve this problem.

From Apple's documentation:

> If `true`, the app has an update available. This key is present only for App Store apps. In macOS, this key is present only for Volume Purchase Program (VPP) apps. This status updates daily and isn't always up-to-date when installing an app.

Unfortunately, the flag provided by the device is not 100% reliable. FileWave will only update the application once a day. The InstallApplication command will not be sent if the device reports HasUpdateAvailable=True and the same command has been already sent that same day. But if the flag is not properly updated by the device the next day, FileWave will request an update.

We have seen reports in the past that some iOS versions would not update the flag properly even if the application is up to date. And because the flag is not updated within 24h, FileWave will request the application to be reinstalled every day, which may be an issue for the end-user or for network bandwidth. In this situation, the best is to contact FileWave support and Apple Care. We will help you configuring both FileWave and your device to gather all the logs Apple will require to investigate the issue. Most of the time, upgrading iOS or macOS to the most recent available version solves the problem.

When looking at FileWave's Client Info for impacted devices, the Managed Applications tab will show you the value of the flag - if it is always "True", there is likely something wrong on at the device level.

Apple Content Caching service

What

Every device will individually reach out to Apple to pull Software Updates and VPP Apps and their respective updates. To alleviate the amount of external traffic, Apple provide Content Caching.

When/Why

Any macOS device may be configured to provide Content Caching. Once configured, devices residing on the same network will be told to use the Content Caching device to pull all updates. This means each update is only pulled from Apple once, for that network, and devices will pull any cached updates from the device sharing this cache.

How

When a device is informed to instal an OS update or VPP App (or App update), the device reaches out to the App Store. Typically the device would then pull that update directly from Apple. However, for any registered Content Caching devices on that network, the response from Apple would be to pull the update from the Content Caching device instead. For any update, the initial request for any one update will cause the Content Caching device to download the update first. Once downloaded all requesting devices may then receive the update from the local caching server instead.

Related Content

- Intro to content caching – Apple Support (UK)
- macOS User Guide – Apple Support (UK)
- Set up content caching on Mac – Apple Support (UK)

Digging Deeper

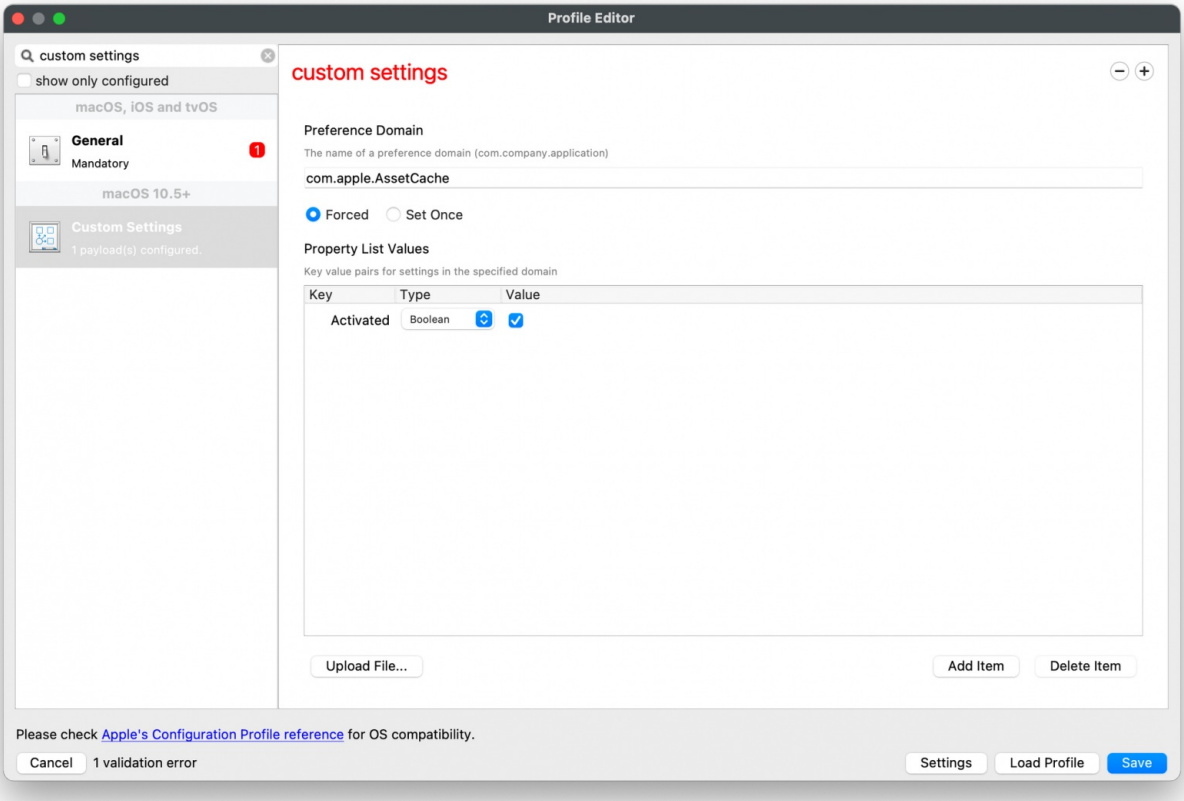
Caching configuration is stored in the following file:

```
/Library/Preferences/com.apple.AssetCache.plist
```

A Custom Field could be created, using the following code, to determine devices which have Content Caching Enabled – Boolean 0(False) 1(True)

```
defaults read /Library/Preferences/com.apple.AssetCache.plist Activated
```

A device could be set to cache, by way of a Custom Settings Payload:


Profile Editor
custom settings
show only configured
macOS, iOS and tvOS
General
Mandatory
1
macOS 10.5+
Custom Settings
1 payload(s) configured.
custom settings
Preference Domain
The name of a preference domain (com.company.application)
com.apple.AssetCache
Forced
Set Once
Property List Values
Key value pairs for settings in the specified domain
Key
Type
Value
Activated
Boolean
Upload File...
Add Item
Delete Item
Please check Apple's Configuration Profile reference for OS compatibility.
Cancel
1 validation error
Settings
Load Profile
Save

Bypassing DPI for Apple Traffic in MDM Communication

What

This article explains the importance of bypassing Deep Packet Inspection (DPI) on network traffic directed to Apple's IP range (17.0.0.0/8) to ensure seamless communication between Apple devices and the FileWave Mobile Device Management (MDM) solution.

When/Why

Deep Packet Inspection is a network packet filtering technique that examines the data part (and possibly also the header) of a packet as it passes an inspection point, to determine what to do with the packet based on its content. This is often employed in firewalls, intrusion prevention systems, and content filters to scrutinize traffic for security and compliance purposes.

However, when managing Apple devices via an MDM solution like FileWave, it's crucial to ensure uninterrupted communication with Apple's network. The DPI can interfere with the SSL traffic to and from Apple's servers, thus hindering the communication between your managed devices and the MDM server. This is particularly vital for the initial device setup, software updates, and continuous management operations.

How

To prevent any interference with the communication between Apple devices and FileWave MDM, it's advised to configure your network's firewall and content filters to bypass or disable Deep Packet Inspection for traffic destined to or originating from the IP range 17.0.0.0/8. Here are general steps:

1. Access Firewall/Content Filter Settings:
   - Log in to your firewall or content filter management interface.
2. Create a Bypass Rule:
   - Navigate to the section where you can create rules or policies.
   - Create a new rule to bypass DPI for the IP range 17.0.0.0/8.
3. Verify Configuration:
   - After setting the rule, verify the configuration by testing the communication between your MDM and an Apple device.
   - You can also check the logs to ensure traffic is flowing correctly without any SSL manipulation.

Related Links

- Deep Packet Inspection (Wikipedia) - Overview of Deep Packet Inspection.
- Apple's Managed Devices - Understanding Apple's Managed Devices and their communication.
- Default TCP and UDP Port Usage - FileWave port usage.

Digging Deeper

Understanding the technical intricacies of network traffic inspection and its implications on MDM communication is crucial for ensuring a seamless operation of managed Apple devices. Disabling DPI for specified traffic ensures that the necessary communication between your FileWave MDM server and managed Apple devices remains uninterrupted, providing a stable and reliable management infrastructure.

Hardware Encryption Capabilities for Apple Hardware

What

From a security perspective, it is important to understand the encryption capabilities of devices.

When/Why

In FileWave 14.6.0 some reporting was added to report on HardWareEncryptionCaps (https://developer.apple.com/documentation/devicemanagement/securityinforesponse/securityinfo?changes=latest_minor) as reported through Apple's MDM framework.

How

- Hardware Encryption Capabilities has been added as a field for iOS 4+ and tvOS 6+ devices to report the supported encryption.
- Passcode Present had its description updated to explain how it ties to Hardware Encryption Capabilities and also is for iOS 4+ and tvOS 6+.
- Is Recovery Lock Enabled was added for macOS devices to reflect if Recovery Lock is enabled on Apple Silicon running macOS 11.5+.

Digging Deeper

HardwareEncryptionCaps is an integer that indicates the underlying hardware encryption capabilities of the device, which is one of the following values:

- `1`: Block-level encryption
- `2`: File-level encryption
- `3`: Both block-level and file-level encryption

This value is available in iOS 4 and later, and tvOS 6 and later.

> For a device to have data protection, `HardwareEncryptionCaps` must be `3` and `PasscodePresent` must `true`.

macOS Sonoma / iOS 17 support in FileWave 15.1.0+

What

iOS, iPadOS and tvOS 17 will be released on Monday, 18th of September. macOS 14 Sonoma was released on the 26th of September. FileWave 15.1.0 released on Thursday, September 21, 2023.

When/Why

As an administrator, it is important to know if these new Operating Systems are supported in order to schedule device upgrade or new purchase.

How

iOS, tvOS, iPadOS 17

Generally, devices running iOS, tvOS, iPadOS can safely be upgraded at any time. FileWave 15.1.0 will bring new management features for these devices.

macOS 14 Sonoma

Generally, a new version of macOS requires a new version of FileWave : desktop agent and server needs an update to report properly the OS version, likewise various features like SIP detection within Filesets.

In addition, macOS 14 introduces changes related to application sandboxing which prevent FileWave 15.0 desktop agent or earlier versions to upgrade without a device restart to complete the upgrade. FileWave 15.1 has been modified to adapt to these changes, therefore it is then recommended to upgrade to FileWave 15.1 before upgrading to macOS Sonoma.

In case devices are upgraded to macOS Sonoma before FileWave is upgraded to 15.1, you have to know that:

- These devices won't report the macOS version properly until they are upgraded to FileWave 15.1 or newer client.
- A device restart will be required to upgrade FileWave client to version 15.1. This can be achieved automatically by enabling the "requires reboot" option if you go to Properties for the upgrade Fileset.

As a reminder, released version of Operating Systems can be supported. Testing existing and new features is made with Beta versions which are subject to change. These OS will be supported after the release and going through our QA process.

Related Content

- FileWave Downloads
- macOS 14 Compatible Devices (Custom Field)
- iOS 17 Compatible Devices (Query)

MDM Lost Mode (Apple)

Starting with iOS 9.3, supervised devices were able to be set to "MDM Lost Mode." Missing devices can be locked, displaying a message, phone number, and footnote. FileWave 11.1+ integrated this new feature with the "Missing" state. Changing device state to "Missing" will automatically send the new commands. You can configure text that will be displayed on the device in the Organizational Info portion of FileWave Admin Preferences. These strings are optional; however, we recommend that you specify a phone number or message. FileWave will display "Lost device" on an iOS device that is set to missing if nothing is provided in the settings. Now will the release of iOS 10.3 and FileWave 12.0+ you have the option to "Play Lost Mode Sound" on your devices. After you have set your device to missing, simply right click the it and select "Play Lost Mode Sound (iOS 10.3+)" the only way to turn that off will then to take the device out of "Missing".

Microsoft Enterprise SSO plug-in for Apple devices

What

The Microsoft Single Sign-On (SSO) plug-in for Apple devices is a software extension that allows users to log in to Microsoft services on their Apple devices without needing to enter their credentials each time. This plug-in enables users to authenticate once and use Microsoft services seamlessly across multiple applications and services. For more information, you may visit, Microsoft Enterprise SSO plug-in for Apple devices.

When/Why

Using the Microsoft SSO plug-in for Apple devices offers several advantages. First, it saves time by eliminating the need to enter login credentials each time a user needs to access Microsoft services. This can be particularly useful for users using Microsoft services on their Apple devices.

Second, the Microsoft SSO plug-in provides an added layer of security. Users can use multi-factor authentication (MFA) to secure their login credentials and protect their data from unauthorized access. The plug-in provides a more secure way to access Microsoft services on Apple devices than standard login credentials.

Finally, the Microsoft SSO plug-in offers a more streamlined and user-friendly experience. Users can easily switch between different Microsoft services without needing to log in again and quickly access their files and data on any device.

How

Below are the following requirements and configuration creation steps for deployment.

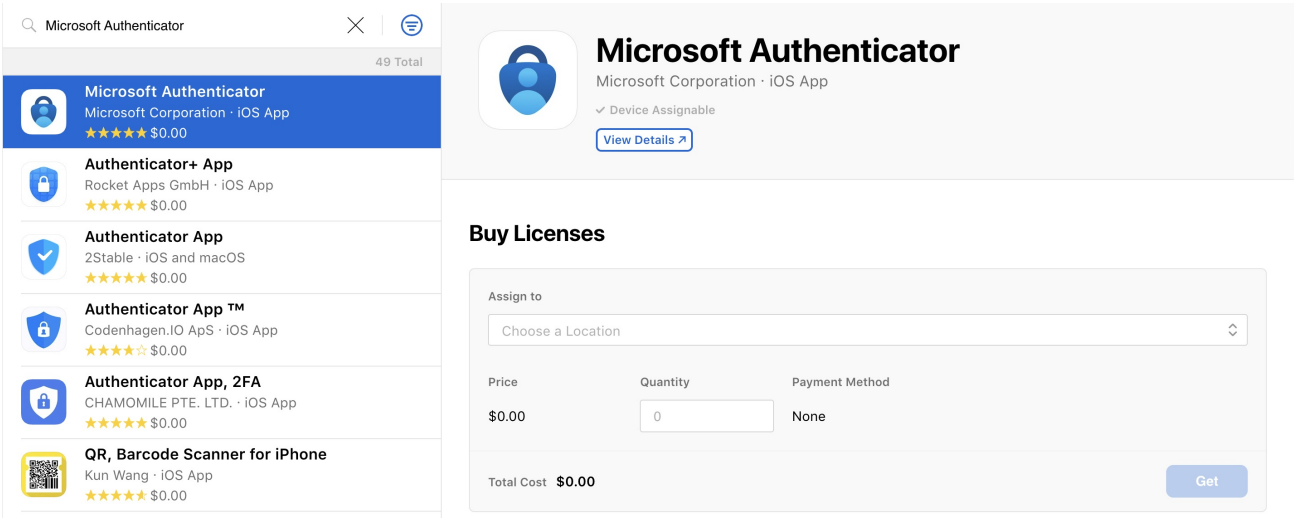
Requirements:

- The device must support and have an installed app that has the Microsoft Enterprise SSO plug-in for Apple devices:
  - iOS 13.0 and later: Microsoft Authenticator app
  - iPadOS 13.0 and later: Microsoft Authenticator app
  - macOS 10.15 and later: Intune Company Portal app
- The device must be enrolled in MDM, i.e. DEP enrolled.
- Configuration must be pushed to the device to enable the Enterprise SSO plug-in. Apple requires this security constraint.

> Please Note: On macOS devices, Apple requires the Company Portal app be installed. Users don't need to use or configure the Company Portal app, it just needs to be installed on the device. You may download here: Download the Company Portal app installer package.

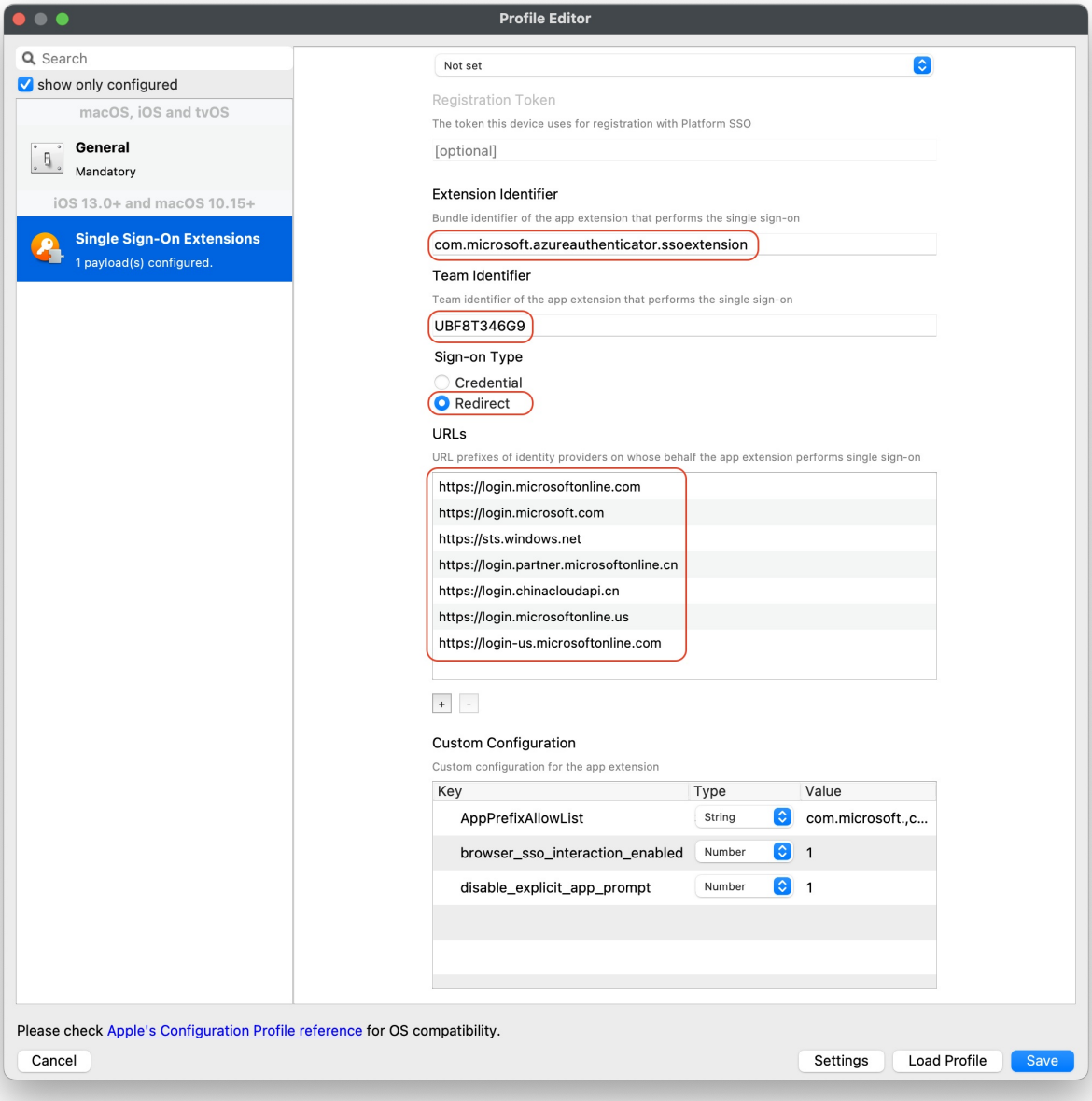
Microsoft Authenticator app deployment:

You may acquire and deploy the Microsoft Authenticator app via your ASM/ABM account. A similar method as any VPP application, search the ASM/ABM, enter in the number of licenses for the VPP application, and click on GET.



Creating the Configuration profile to be deployed to your devices:

1. Open FileWave Central
2. Select Filesets from the left side menu
3. Select New Desktop Fileset
4. Click on Profile
5. Enter in the name of the Profile, example: Microsoft Single-Sign On
6. Select the Single Sign-On Extensions payload
7. Enter in the following for specified payload:
    1. iOS settings:
        - Extension ID: `com.microsoft.azureauthenticator.ssoextension`
        - Team ID: This field isn't needed for iOS but you can use `UBF8T346G9`
    2. macOS settings:
        - Extension ID: `com.microsoft.CompanyPortalMac.ssoextension`
        - Team ID: `UBF8T346G9`
    3. Sign-On Type:
        - Type: Redirect
    4. URL identity providers:
        - `https://login.microsoftonline.com`
        - `https://login.microsoft.com`
        - `https://sts.windows.net`
        - `https://login.partner.microsoftonline.cn`
        - `https://login.chinacloudapi.cn`
        - `https://login.microsoftonline.us`
        - `https://login-us.microsoftonline.com`
    5. Optional Custom Configurations (Not required):
        - Enable SSO for all apps with specific bundle IDs or prefix IDs: Key:AppPreFixAllowList - Type:String - Value:com.microsoft., com.apple., or com.business.travelapp
        - Sign in with browser that don’t use MSAL and Safari: Key:browser_sso_interaction_enabled - Type:Number - Value:1
        - Disable OAuth 2 app prompts: Key:disable_explicit_app_prompt - Type:Number - Value:1



Deployment

Next deploy the Microsoft Authenticator app and Configuration profile on a few devices. If you're not deploying the Microsoft Authenticator app using an app policy, then users must install it manually. Users don't need to use the Authenticator app, it just needs to be installed on the device.

Users sign in to any supported app or website to bootstrap the extension.

> Bootstrap is the process of signing in for the first time, which sets up the extension. After users sign in successfully, the extension is automatically used to sign in to any other supported app or website.
>
> Meaning the end users will need to sign into their Microsoft account for their first time manually for the extension to authenticate successfully.

You can test single sign-on by opening Safari in private mode (opens Apple's web site) and opening the https://portal.office.com site. If configured successfully, no username and password will be required.

Related Content

- Microsoft SSO for macOS devices

Understanding Similar and Identical Software Update Names in FileWave for Apple Devices

What

As a Unified Endpoint Management tool, FileWave manages a wide range of devices, including Apple devices such as macOS, iOS, iPadOS, and tvOS. In the Software Updates list for these devices, you may notice updates that have similar or identical names, such as "iPadOS 16.1," "iPadOSs 16.1," and "iOS 16.1." This can be confusing, as it may seem like there are duplicate updates or that the updates are intended for different devices.

When/Why

This is due to how Apple publishes updates for its devices. Different devices, as well as different versions of macOS and iOS, may have updates with slightly different names. For example, an update for iPadOS may have the same version number as an update for iOS, but the names will be slightly different to reflect the intended device.

How

To ensure that all of your Apple devices are updated with the latest patches, it is important to enable all variations of the patch for a specific version (e.g., 16.1) that you are trying to update. This will ensure that all relevant devices receive the necessary updates. Ensuring that all of your Apple devices are up to date is crucial for maintaining the security and functionality of your organization's technology.

Related Content

- Apple MDM Software Updates
- View - Software Updates