# Apple MDM

With the exception of macOS, all Apple devices are managed purely through MDM.  macOS devices on the other hand, relies upon the FileWave Client, however, when MDM enrolled, benefit from the additional features

- Address Stalled MDM Commands
- Apple MDM Lost Mode
- Apple MDM Command History

# Address Stalled MDM Commands

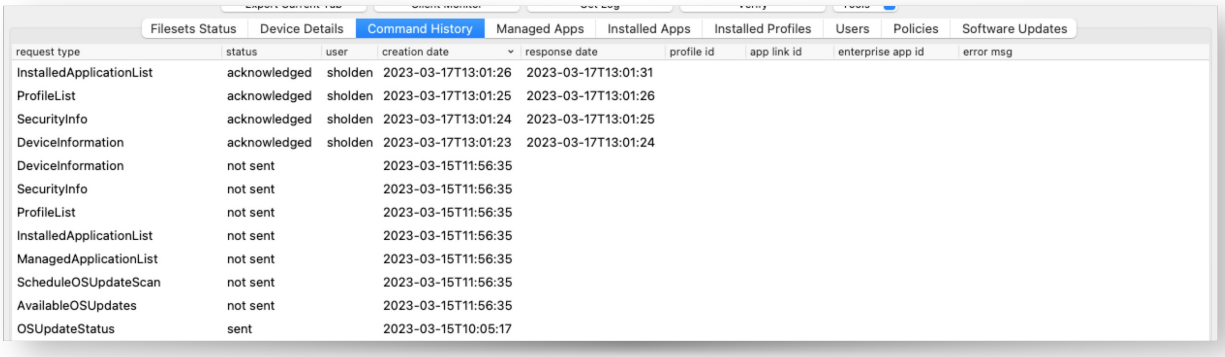
## Description

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

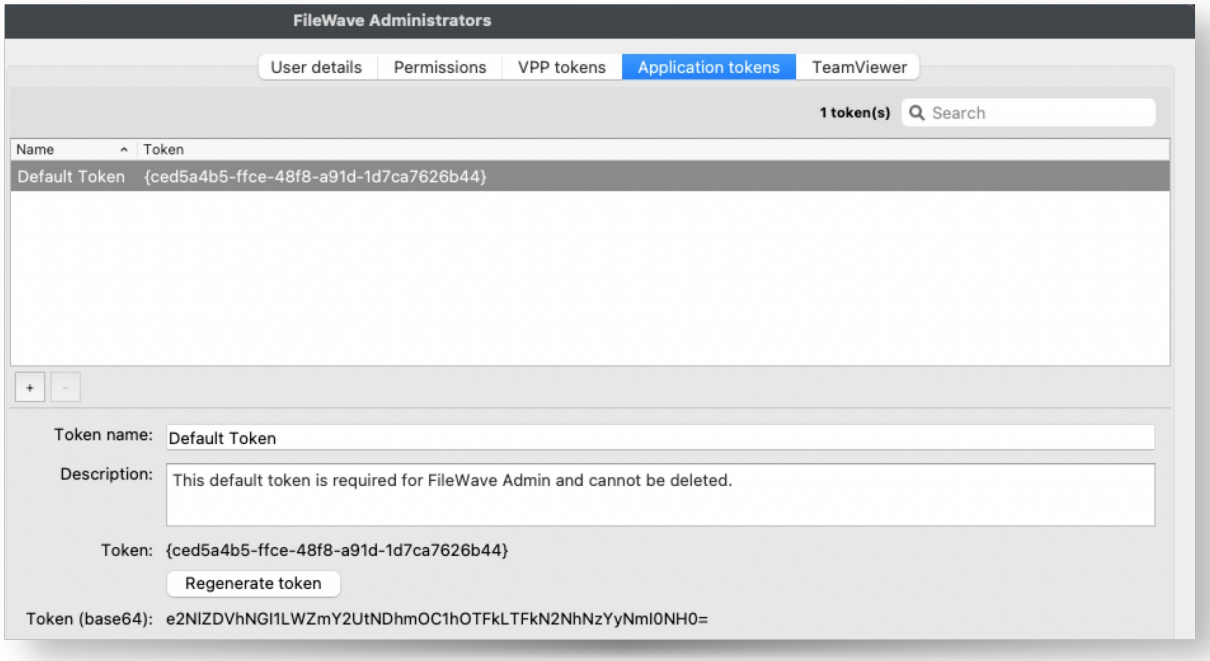
1. Copy the provided Script to the FileWave Server
2. Edit the top of the script, providing the FileWave Administrator Authorisation Token and Server URL values

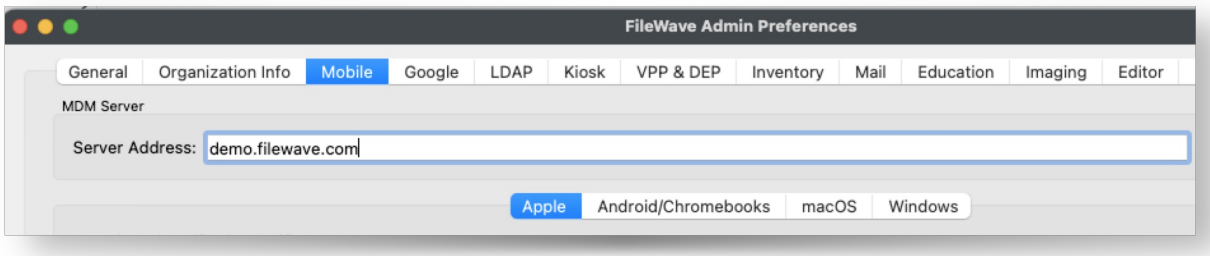
```
#!/bin/zsh

# 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
```

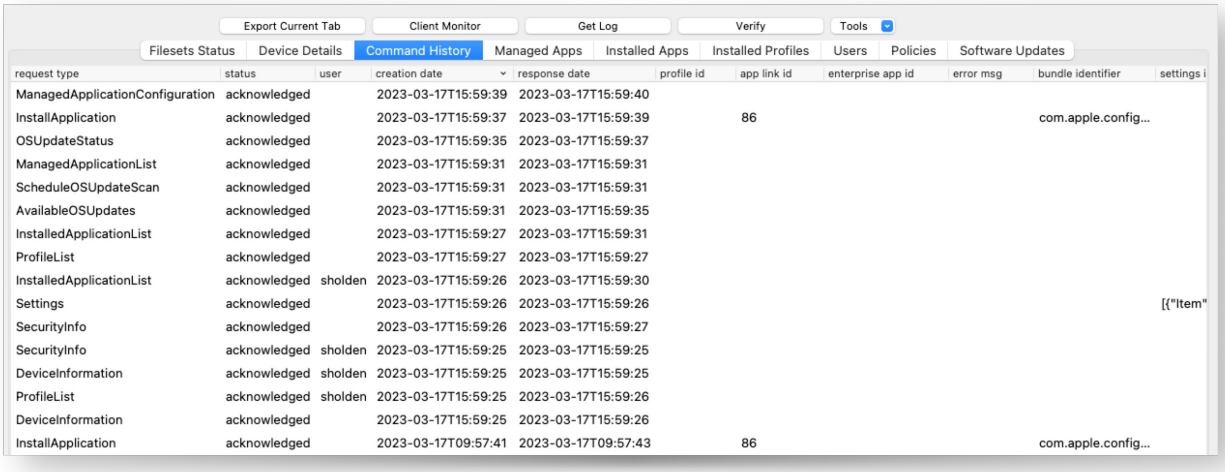
Since this script was written using ZSH, then if not already installed, the ZSH shell will require installing. For macOS servers, ZSH is default, however on CentOS servers it is likely not yet installed. To instal ZSH use the following command:

```
yum install zsh
```

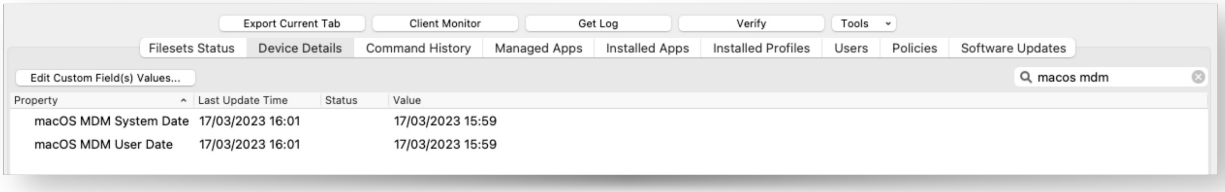
## Testing the Script

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:

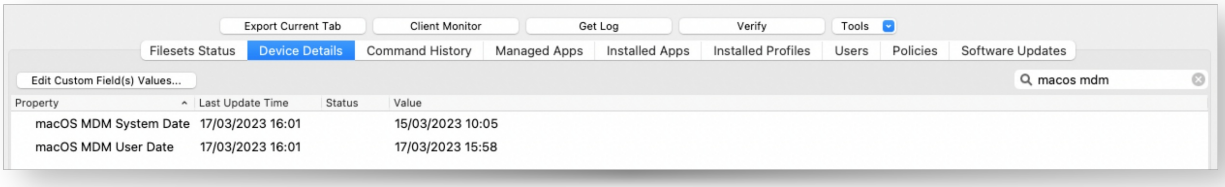



The second image shows the Custom Fields populated with those values, note both date values match:

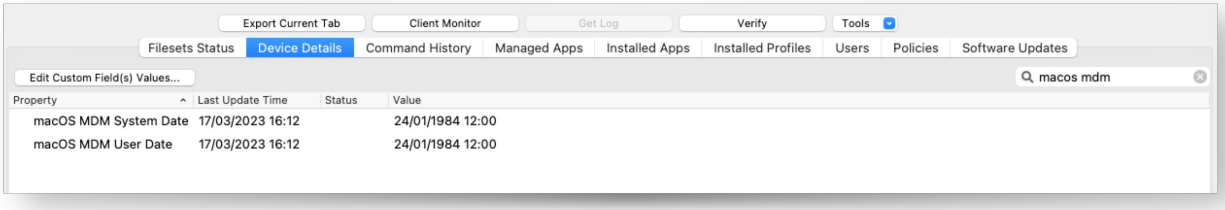



The following image is an example of a device where the device was communicating, but is no longer acknowledging MDM System Channel commands, note the System date value

is more than 2 days older:




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




## 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

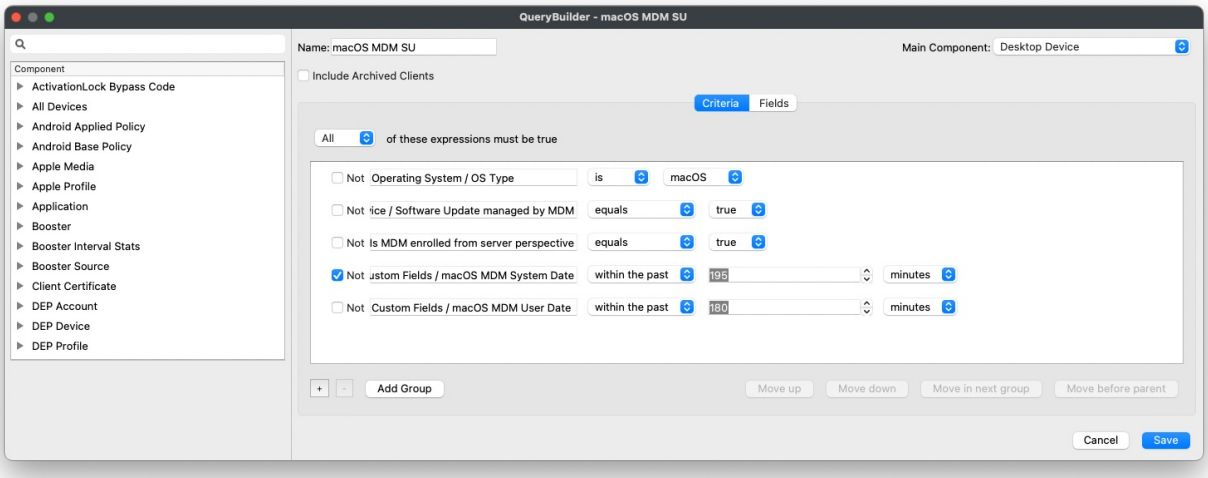
> ⚠ Do not run the cronjob too often. However, in the same right, if a device were addressed and subsequently MDM stalled again between the two cronjob actions, since the device would not have left and re-entered the Smart Group, the Fileset will not reinstall, unless manually re-triggered.

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 MDM Lost Mode

## What

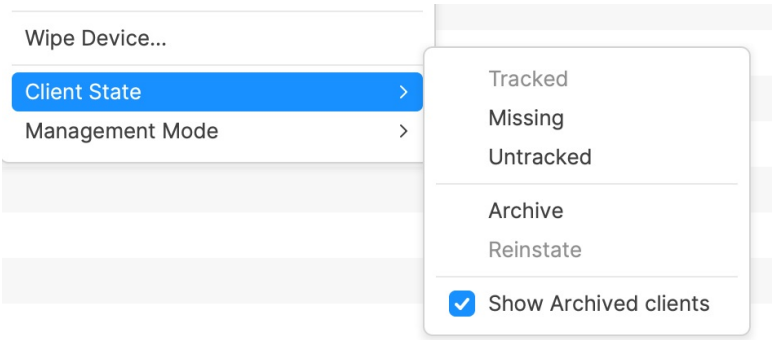
Lost Mode locks devices from use until Lost Mode is disabled.  This enhances the security whilst devices are in an unknown location.

## Why

On occasion, users can misplace devices.  To assist the protection of data and prevent anyone from using the device until relocated, an MDM command may be sent to Lock the device.

## Information

Lost Mode is enabled by setting a device into the 'Missing' Client State, via the right click contextual menu:




To disable Lost Mode, select any other Client State, Tracked or Untracked.  A Model Update is required after altering the Client State before the MDM command will be sent to the device.

Devices require network to receive the command to enable or disable Lost Mode.

If a device is Locked and unable to receive network due to the surrounding W-Fi networks not yet being configured, consider connecting the device to ethernet (adaptor may be required).

## Locating Devices

Whilst locked, as well as devices being unusable, additional features help locate the device

### Location

After being set as Lost, location data will be sent back to the FileWave Server, where the device has a network connection.  Location may be viewed on a map from the Client Info view as well as related location data being available from Inventory Queries.

### Sound

Location is all very well, but what if the device is in a bag, cupboard or similar.  To further assist retrieval of the device, when in the Lost Mode 'missing' state, an additional option will be available from the right click menu: 'Play Lost Mode Sound'.  Previous set volume is not a consideration and the device will make an audible set of tones.

# Apple MDM Command History

## What

Any Apple devices that are MDM enrolled should receive MDM Requests.  Each Client Info lists the requests for that device.

## Why

MDM Command History exists to display all requests queued and sent to devices, along with the result of those requests; additionally showing if the request was designed for the User of the device or the System

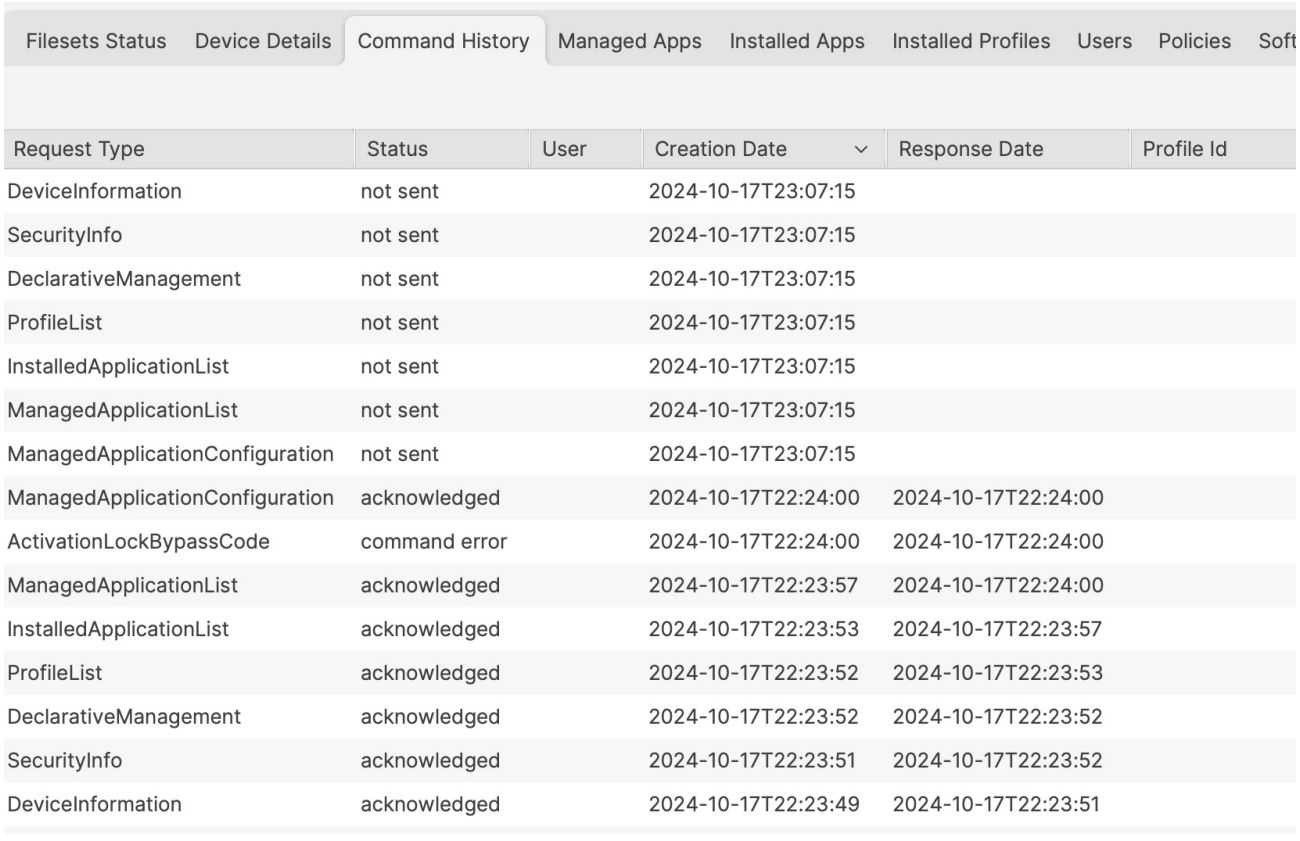
> When referring to Users and MDM, Apple allow for any amount of directory users to be managed, but only one local user is considered to be managed.  This is the first local user to log into the device after enrolment.  System requests impact all users.

The Request Types include:

- Inventory
- Profile instals and uninstalls
- VPP App instals and uninstalls
- Commands, e.g. erasing a device, renaming the device, etc.

## Information

Opening the Client Info for an Apple MDM enrolled device and selecting the Command History tab should show something similar to the below image:




| Request Type | Status | User | Creation Date | Response Date | Profile Id |
|---|---|---|---|---|---|
| DeviceInformation | not sent | | 2024-10-17T23:07:15 | | |
| SecurityInfo | not sent | | 2024-10-17T23:07:15 | | |
| DeclarativeManagement | not sent | | 2024-10-17T23:07:15 | | |
| ProfileList | not sent | | 2024-10-17T23:07:15 | | |
| InstalledApplicationList | not sent | | 2024-10-17T23:07:15 | | |
| ManagedApplicationList | not sent | | 2024-10-17T23:07:15 | | |
| ManagedApplicationConfiguration | not sent | | 2024-10-17T23:07:15 | | |
| ManagedApplicationConfiguration | acknowledged | | 2024-10-17T22:24:00 | 2024-10-17T22:24:00 | |
| ActivationLockBypassCode | command error | | 2024-10-17T22:24:00 | 2024-10-17T22:24:00 | |
| ManagedApplicationList | acknowledged | | 2024-10-17T22:23:57 | 2024-10-17T22:24:00 | |
| InstalledApplicationList | acknowledged | | 2024-10-17T22:23:53 | 2024-10-17T22:23:57 | |
| ProfileList | acknowledged | | 2024-10-17T22:23:52 | 2024-10-17T22:23:53 | |
| DeclarativeManagement | acknowledged | | 2024-10-17T22:23:52 | 2024-10-17T22:23:52 | |
| SecurityInfo | acknowledged | | 2024-10-17T22:23:51 | 2024-10-17T22:23:52 | |
| DeviceInformation | acknowledged | | 2024-10-17T22:23:49 | 2024-10-17T22:23:51 | |

Request Types are defined by Apple and details may be viewed on the Apple Development Pages

Status and Error messages are those reported by Apple, with the exception of 'not sent'.  The possible values are:

| Status | Details | Apple Request Response |
|---|---|---|
| not sent | The command is queued and awaiting for the device to reply to the APNs request | - |
| acknowledged | Device has processed the request | Acknowledged |
| not now | Device has received the request, but is unable to | NotNow |

| | | |
|---|---|---|
| | process the request at this time | |
| command error | An error has occurred | Error or CommandFormatError |

As of FileWave 15.5, the status of Command Queue requests are now accessible from within standard Inventory Queries

### 'not sent'

Before any requests may be sent to a device, the FileWave Server sends an Apple Push Notification (APN) request to Apple. Apple queue these APNs requests and only after the device checks in with Apple and pulls the APNs request, will the device then check-in with the defined server.

APNs request is nothing more that a notification for the device to check-in with the defined server.

Whilst waiting for the device to receive the APNs request and check-in, the Command History will display 'not sent'

Where requests are 'not sent' or 'not now', new requests will not be added to the queue for the same Request Types, since there is already a queued request waiting. The 'Creation Date' displays the time and day the request was added to the queue.

## Response Commands

Once the APNs request has been received by the device, on check-in, queued requests may then be sent to the device.

### 'acknowledged'

The request has been received by the device, processed and reported back to the FileWave Server as completed.

### 'not now'

Some requests will not be accepted, until the device is in a certain state. For example, a user may need to be logged into the device to process the provided request.

Hence, the request has been received and the device has responded, but the request is awaiting the desired state before it will finish processing the request and report back as much.

### 'command error'

In some instances, the request may not be able to complete, due to an error. Apple have two defined error status values:

- 'Error' - An error occurred (the device will report error details in the response to the request)
- 'CommandFormatError' - The request protocol was incorrect, e.g. a malformed request

The 'Error Msg' column shown in the Command History view reports information provided by the device back to FileWave Server and contains information as set out by Apple.

Apple's developer page demonstrates greater depth on MDM requests:

Sending MDM Commands to a Device

## User vs System

The user column contains the channel used for the request. Where the user column is blank, this implies the request is a System Channel request. Otherwise the name of any managed users existing on the device will be displayed for those requests.

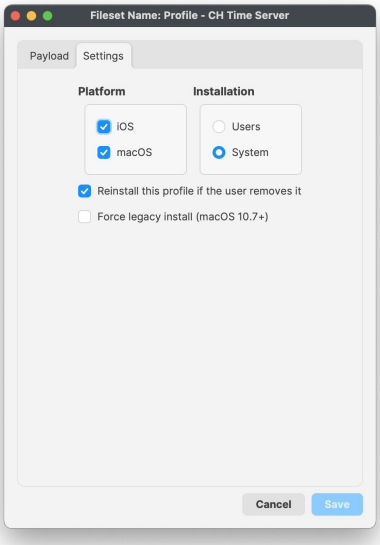
Some Request Types are required for both User and System, e.g.


DeviceInformation may report inventory regarding the System and the User. As such there are multiple requests, one per Managed

User and one for System.

## Profile Installation

Profile Settings defines whether a Profile should be Installed for the System or User:




As with other Request Types, if the Profile is configured for System Installation, the User channel column should be blank, whilst those set as User should show a Request Type of 'InstallProfile' per managed user.

The below image shows installation of two differing Profiles, one set as System and the other set as User:

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

| Request Type | Status | User | Creation Date | Settings Items | Response Date | Profile Id |
|---|---|---|---|---|---|---|
| InstallProfile | acknowledged | | 2024-10-17T23:55:25 | | 2024-10-17T23:55:26 | fw1063.home.22e28158-118c-4956-8640-1f4119( |
| InstallProfile | acknowledged | sholden | 2024-10-09T22:14:08 | | 2024-10-09T22:14:08 | ml1063.local.9645a261-8941-4827-b0b8-21b97a |

There was a period of time where all Profiles could be set as either User or System regardless. Apple enforced 'correct' Installation channels several major versions back. As such, if Profiles were delivered before such change, it may be possible that the User column may show a User despite the Profile being set as System. This should no longer be the case for newly delivered requests

Although altered values in a Profile Payload will just cause the Profile to be updated on a device, changing the Installation channel from User to System or vice versa, will cause the Profile to be uninstalled and then re-installed using the newly defined Installation channel.

Only some Profile Payloads may be defined as either User or System. If one option is greyed out, the Profile Setting cannot be changed.

# Commands

Some requests are commands designed to action an event, e.g. rename a device, erase a device, etc. If the request is altering a setting, the Request Type reported is 'Settings' and the 'Settings Items' column will display details regarding the setting to be altered.

The below image shows a request to alter the name of the device:

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

| Request Type | Status | User | Creation Date | Settings Items |
|---|---|---|---|---|
| Settings | not sent | | 2024-09-17T21:00:59 | [{"Item": "DeviceName", "DeviceName": "macOS12VM"}] |

An Update Model will be required before the Setting request is added to the device's request queue.

Further details regarding 'Command Policy' Fileset Payloads can be viewed in the following KB:

Profile Editor Command Policy

# Troubleshooting

Some typical items to consider when reviewing Command History:

- The Command History tab will not be present if the device is not MDM enrolled (i.e. a macOS device with only the FileWave Client installed)
- A profile does not show on the device, yet command is acknowledged. This is typically seen where a Profile is set as User, but the currently logged in user is a local user, but not the local managed user. Compare the Command History User column with the currently logged in user.
- Where possible, it only might be prudent to set Profiles as System rather than user, if all local users require management (Note: this may also impact Administrators logging into systems, when attempting to fix a user's issue)
- Where an error has occurred, review the 'Error Msg' column. If the error message does not appear clear, consider raising a ticket with the FileWave Support team.
- As noted above, a change of Profile Setting causes a Profile to be removed, before being delivered on the newly defined Installation channel (User or System)l. If the Profile in question is required for network connection (for example), there is no way for the device to receive the newly set Profile, since the device will no longer have a working network connection.