

# 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.

Device Information									
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

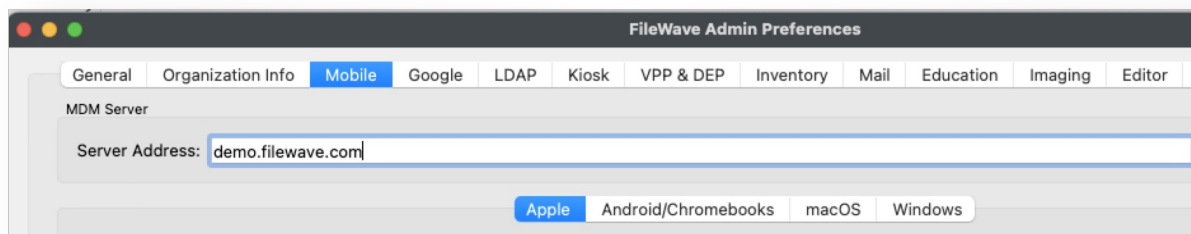
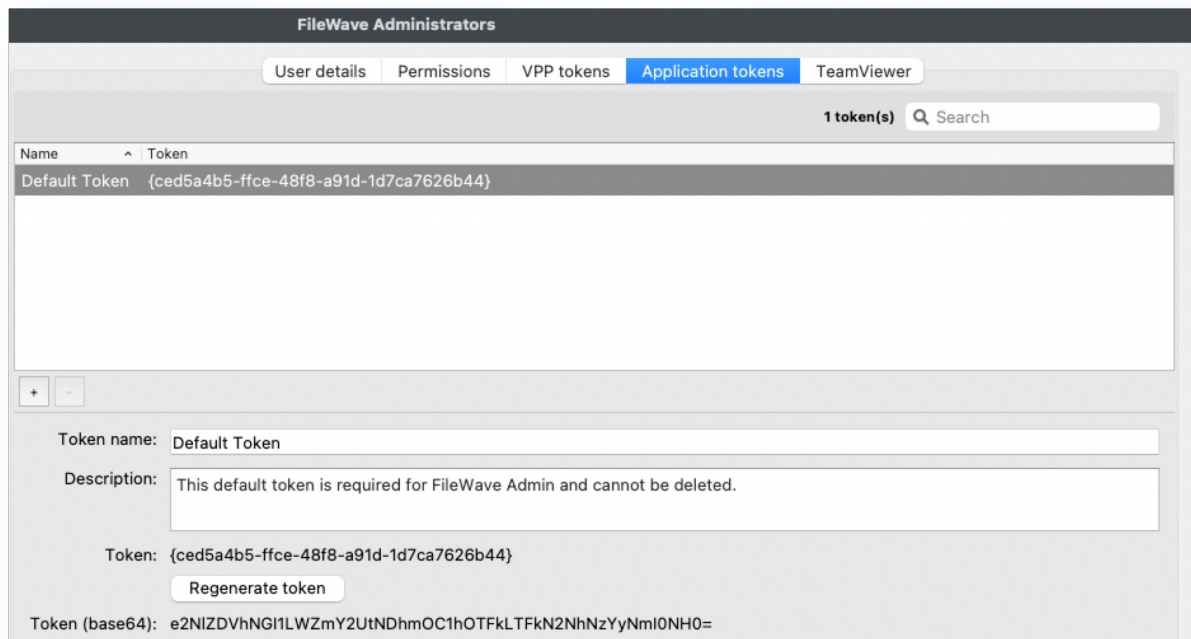
1. Copy the provided Script to the FileWave Server
2. Edit the top of the script, providing the FileWave Administrator Authorisation Token and Server URL values

```
#!/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:

Export Current Tab Client Monitor Get Log Verify Tools										
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	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:

Export Current Tab Client Monitor Get Log Verify Tools										
Filesets Status Device Details Command History Managed Apps Installed Apps Installed Profiles Users Policies Software Updates										
Edit Custom Field(s) Values... macos mdm										
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:

Export Current Tab Client Monitor Get Log Verify Tools										
Filesets Status Device Details Command History Managed Apps Installed Apps Installed Profiles Users Policies Software Updates										
Edit Custom Field(s) Values... macos mdm										
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

Export Current Tab Client Monitor Get Log Verify Tools										
Filesets Status Device Details Command History Managed Apps Installed Apps Installed Profiles Users Policies Software Updates										
Edit Custom Field(s) Values... macos mdm										
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

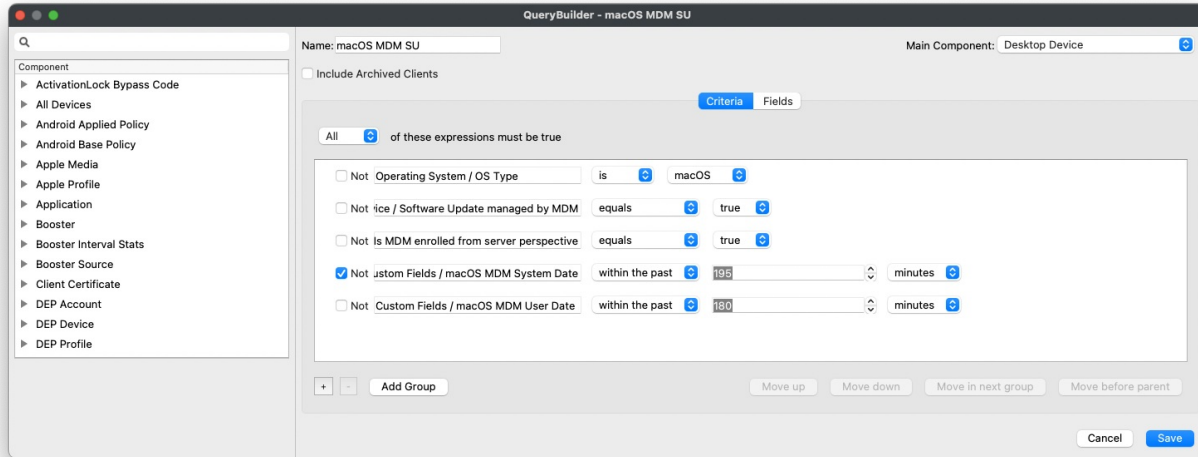
Do not run the cronjob too often. However, in the same right, if a device were addressed and subsequently MDM stalled again between the two cronjob actions, since the device would not have left and re-entered the Smart Group, the Fileset will not reinstall, unless manually re-triggered.

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.

Revision #9

★ Created 14 July 2023 13:53:16 by Josh Levitsky

✎ Updated 13 December 2023 17:50:15 by Josh Levitsky