


S.U.P.E.R.M.A.N. for macOS Software Updates (macOS Script)

What is S.U.P.E.R.M.A.N.?

S.U.P.E.R.M.A.N. (Software Update Policy Enforcement with Recursive Messaging and Notification) can be an innovative feature within FileWave's Unified Endpoint Management (UEM) tool. Designed to optimize the macOS software updates and upgrades experience, S.U.P.E.R.M.A.N. empowers education organizations, corporations, and state and local government entities to enforce software update policies seamlessly across their diverse endpoint environments.

 Note: Sometimes this tool is referred to as super or superman.

When/Why Use S.U.P.E.R.M.A.N.?

Keeping macOS devices up-to-date with the latest software updates and upgrades is crucial to ensure optimal performance, security, and compatibility. However, managing software updates across a large number of devices can be challenging, especially in organizations with diverse endpoint environments. Here's why you should consider using S.U.P.E.R.M.A.N.:


- Streamlined Software Updates: S.U.P.E.R.M.A.N. simplifies the process of managing macOS software updates and upgrades. It ensures that all devices in the organization have the latest software versions, reducing the risk of security vulnerabilities and compatibility issues.
- Automated Policy Enforcement: With S.U.P.E.R.M.A.N., administrators can define software update policies once and enforce them across all macOS devices automatically. This automation saves time and effort while ensuring consistency in software version deployment.
- Recursive Messaging and Notification: S.U.P.E.R.M.A.N. leverages recursive messaging and notification capabilities to actively prompt users to initiate the update process. This feature encourages end-user participation in keeping their devices up-to-date, reducing potential delays in updates.
- Optimal End-User Experience: By allowing users to initiate the update process, S.U.P.E.R.M.A.N. ensures that updates don't disrupt critical tasks. End-users can conveniently schedule updates during non-productive hours, minimizing interruptions to their workflow.
- Enhanced Security and Compliance: Outdated software can expose endpoints to security risks. S.U.P.E.R.M.A.N. helps organizations maintain a secure environment by enforcing timely software updates, ensuring compliance with data protection regulations.

Mac Computers With Intel

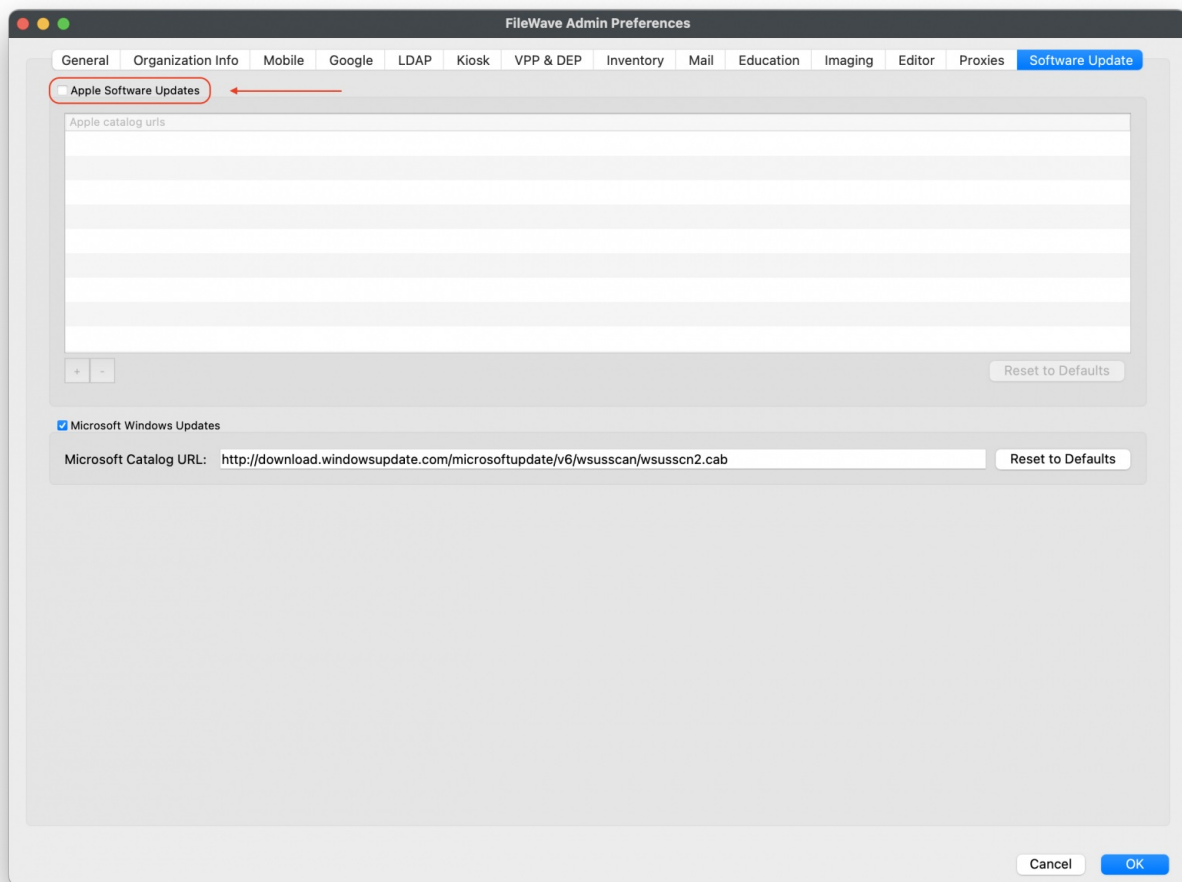
- macOS update and upgrade workflows validated on macOS 10.14 and later. Earlier versions of macOS may work, but have not been validated.
- The `super` script must run with system (root) privileges, but otherwise no additional credentials or MDM service is required.

Mac Computers With Apple Silicon

- The `super` script must run with system (root) privileges.
- To enforce automatic macOS updates or upgrades without using an MDM requires providing `super` with the local credentials of an existing account.
- If no credentials are provided to `super` then macOS updates and upgrades can not be enforced on Mac computers with Apple silicon. In this case `super` prompts the user to provide their local account password; [Apple Silicon Local Credentials](#) · [Macjutsu/super Wiki](#) · [GitHub](#)

 Prior to macOS 11, Software Updates were pulled from Apple's catalogue (not MDM App Store) through FileWave. To achieve this, the client will overwrite the current Software Update plist whilst running and then revert the file once done. This will likely conflict with any Super process, causing Super to fail if the two occur at the same time. You can however get around this.

There is a client setting that could be pushed with Superpref to disable non MDM updates, but easier still, if you disable Legacy Apple Software Updates entirely from the FileWave Central > Preferences > Software Updates tab. This will allow clients not request to attempt non-MDM checks anymore.



Worth noting, that every Model Update, Inventory, Verify, the fwupd client process overwrites this file, since it needs to report which updates are appropriate, so this isn't a case of they might only conflict if you have associated updates, this happens always if not disabled.

How to Use S.U.P.E.R.M.A.N.?

The script must run with system (root) privileges, if you create the Fileset and add the `super` script, it will be run at system (root) privileges just as any other script created in FileWave Filesets.

Name	Size	Access	User	Group	Verification	ID	Mod
var		rw-rw-rw-	root	wheel		227	
FileWave		rw-rw-r--	root	admin		13716	
filewave_icon.png	48.5 kB	rw-r--r--	Typical User ID 501	staff	Self Healing	755946	7/1
scripts		rw-rw-r--	root	wheel		228	
414070		rw-rw-r--	root	wheel		755945	
install_super.sh	456.9 kB	r-x-----	root	wheel	Self Healing	755943	7/1
uninstall_super.sh	2.2 kB	r-x-----	root	wheel	Self Healing	755944	7/1

Above is the completed Fileset ready for deployment. You may also download and review the Fileset here:

[Superman.fileset.zip](#)

✓ For macOS 10.13 (Ventura) and above you should also deploy: [Profile - Superman Managed Login Item.fileset.zip](#)

The Fileset includes both an installation and uninstallation script. If you remove the Fileset from your devices, the uninstallation script will run, removing all content.

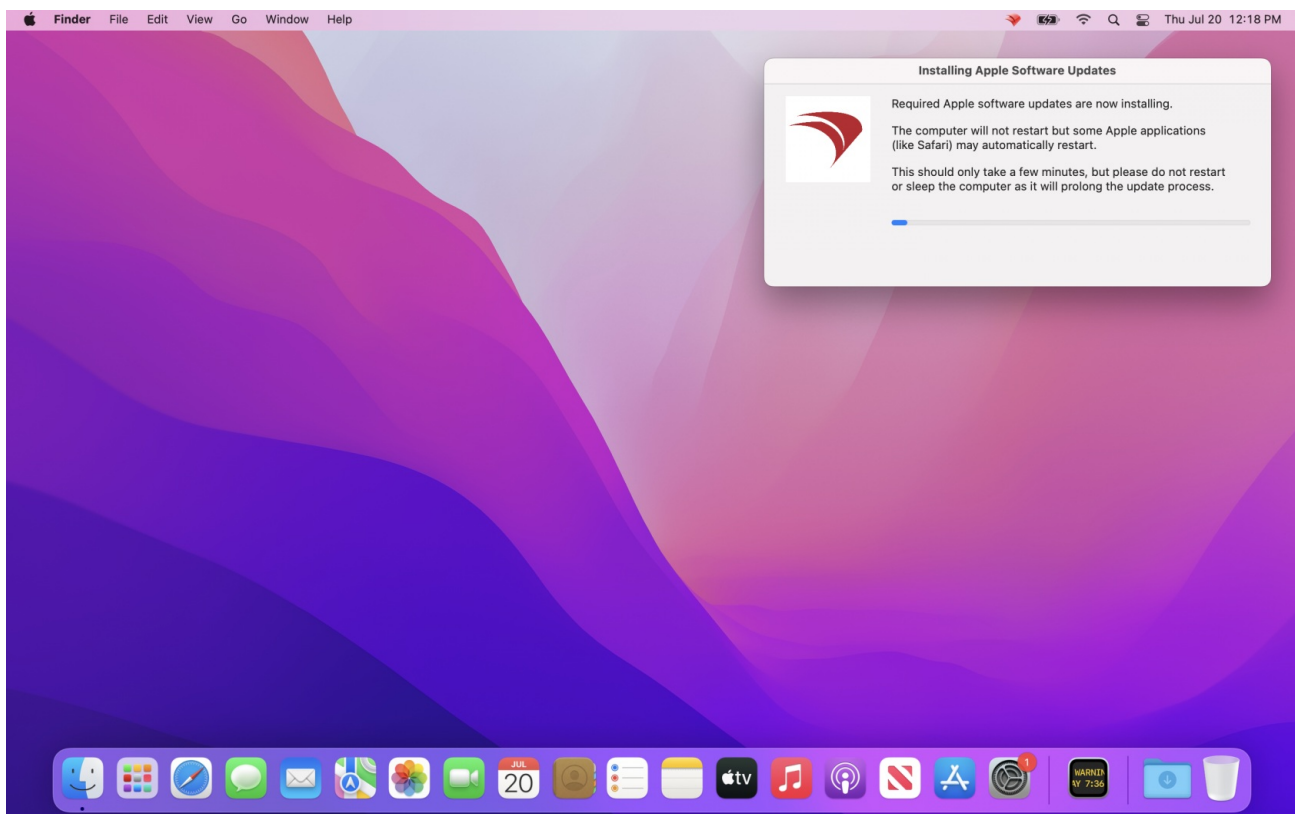
Deployment and installation will be run silently. You may review the log file to troubleshoot or confirm that the installation has been completed. The main `super` workflow log is the `super.log`. This log is located at `/Library/Management/super/logs/super.log` as set by the `$superLOG` script parameter. When run from the command line, `super` also outputs the content of the `super.log` in real time. Example log below:

```
install_super.sh.log Open with Console

----- HEADER - Date: (Wed Jul 19 2023) - Time: (15:31:07) -----
Wed Jul 19 15:31:08 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: **** S.U.P.E.R.M.A.N. 3.0 INSTALLATION ****
Wed Jul 19 15:31:08 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Installation: Copying file: /Library/
Management/super/super
Wed Jul 19 15:31:08 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Installation: Creating search path link: /usr/
local/bin/super
Wed Jul 19 15:31:08 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Installation: Creating LaunchDaemon helper: /
Library/Management/super/super-starter
Wed Jul 19 15:31:08 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Installation: Setting permissions in: /
Library/Management/super
Wed Jul 19 15:31:08 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: **** S.U.P.E.R.M.A.N. 3.0 STARTUP ****
Wed Jul 19 15:31:09 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Status: Current GUI user name is "dummy".
Wed Jul 19 15:31:09 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: No custom display icon found, copying
default icon from: /System/Library/PrivateFrameworks/SoftwareUpdate.framework/Versions/A/Resources/SoftwareUpdate.icns
Wed Jul 19 15:31:09 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: Apple Silicon Mac computer running
macOS 13.4.1-22F70820d.
Wed Jul 19 15:31:09 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: Last macOS startup was Wed Jul 19
15:30.
Wed Jul 19 15:31:09 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Warning: Automatic macOS update/upgrade
enforcement on Apple Silicon computers requires authentication credentials.
Wed Jul 19 15:31:09 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: User authentication is required to
perform a macOS update/upgrade.
Wed Jul 19 15:31:19 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: Unable to locate jamf binary at: /
usr/local/bin/jamf
Wed Jul 19 15:31:19 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: Attempting to download and install
IBM Notifier.app...
Wed Jul 19 15:31:20 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Startup: Found that super is installing,
restarting with new LaunchDaemon...
Wed Jul 19 15:31:20 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: Exit: LaunchDaemon com.macjutsu.super.plist is
scheduled to start right now.
Wed Jul 19 15:31:20 FileWave-Support-C02DNSB8Q6L4 install_super.sh[634]: **** S.U.P.E.R.M.A.N. 3.0 EXIT ****

----- FOOTER - Date: (Wed Jul 19 2023) - Time: (15:31:20) - Exit code: (0) -----
```

Once installed the end user will be prompted with your configured settings. Here is an example prompt with customized icon, and enforcing the non-system updates on a macOS Monterey device:

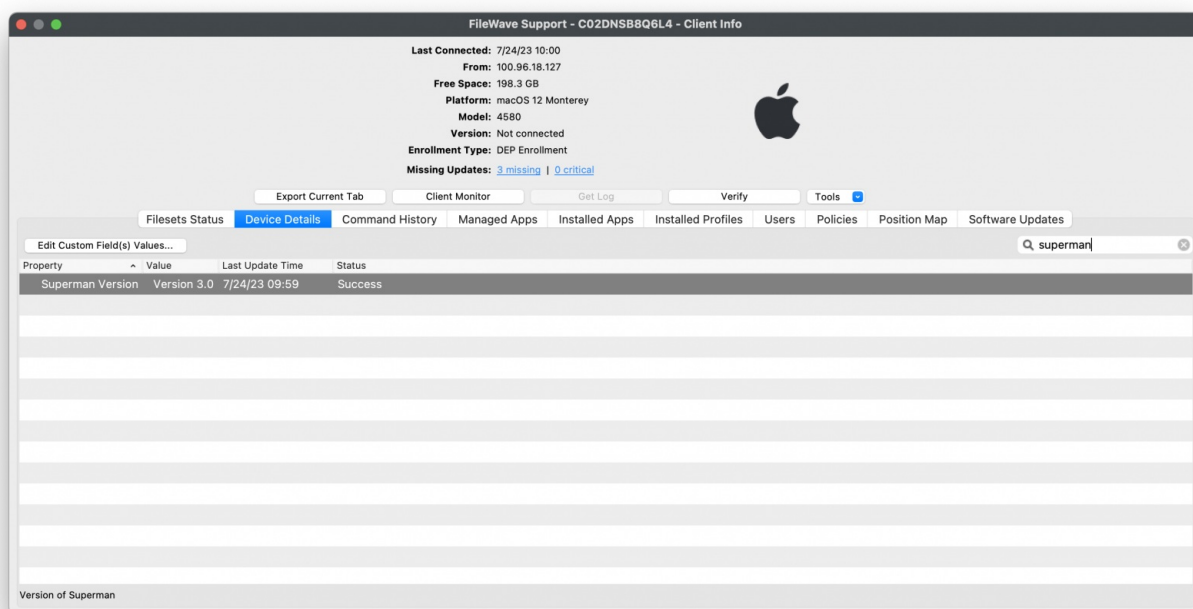


New release of S.U.P.E.R.M.A.N.

With `super` version 3.0, the default dialogs and notifications are handled by [IBM Notifier.app version 2.9.1](#) and macOS upgrade workflows leverage [erase-install.sh version 27.3](#). If either of these items are not found on the local system they are automatically installed by `super` via direct download from GitHub.

Staying up-to-date with the latest version of `super` can be managed with FileWave. Checking the version of `super`, use [Custom Fields](#) to check which version of `super` you have installed on your macOS devices. Below may you download, unzip and import the Custom Field for checking which version is installed:

[Superman customfield.zip](#)

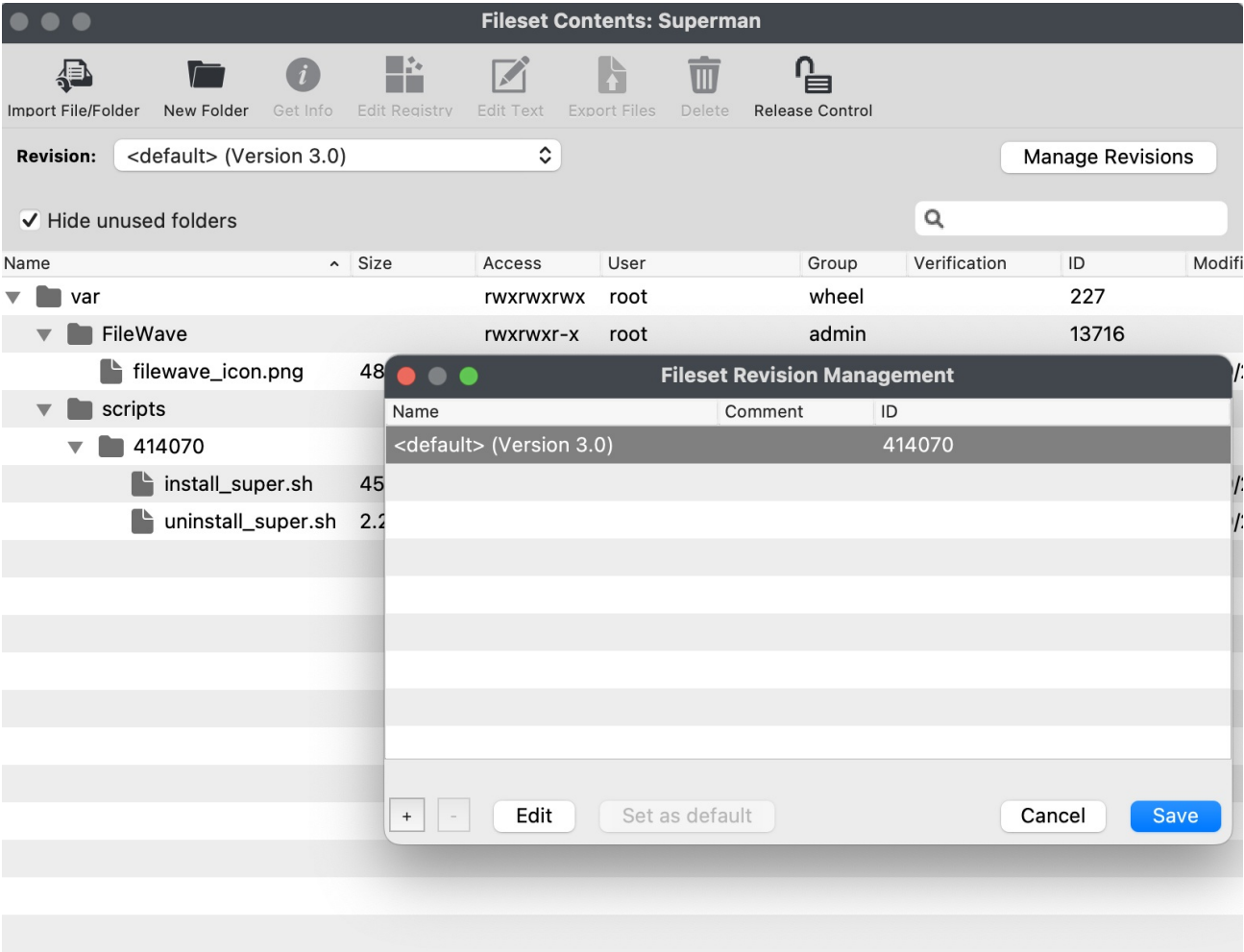


To update the Filesset and deploy the latest release of `super`, the use of [Filesset Revisions](#) works great for this Filesset.

- Highlight your `super` Filesset, then double-click to open its contents. Select Manage Revisions and create a new Filesset

Revision and choose to duplicate everything.

- Replace the install_super.sh with the newest version available from the web page: [Getting Started · Macjutsu/super Wiki · GitHub](#). Save and test deployment to a macOS device and use the custom field to verify latest version installed.



Customizing S.U.P.E.R.M.A.N.

There are many options to customize your deployment. Displaying your customized icon for your organization, or setting up deferral dates or the amount of deferrals before software update requirement.

✔ When customizing the icon, be sure to copy your icon into a folder within the Fileset and specific the correct location path

▼ Option Parameters		
	Option parameter	Description
Allow macOS Upgrades		
	--allow-upgrade or -M	With this option enabled <code>super</code> leverages <code>erase-install.sh</code> to find compatible macOS upgrade versions. For a newer macOS upgrade is available then the <code>super</code> workflow attempts to download and install the upgrade.
Target macOS Version		
	--target-upgrade=12	With this option enabled <code>super</code> does not select any major macOS upgrades newer than the targeted version. For example, if you specified <code>--target-upgrade=12</code> then the <code>super</code> workflow never attempts to install upgrades to macOS 13 or newer.
Allow macOS RSR		
	--allow-rsr-updates or -R	If this option is enable then the <code>super</code> workflow appears


		to the user as a normal macOS update.
Enforce non-System Updates		
	--enforce-non-system-updates or -N	With this option enabled, if non-system Apple software updates are found, they are immediately downloaded and installed.
Skip All Updates		
	--skip-updates or -S	Skip checking for, downloading, or installing any Apple software updates or upgrades, even if they are available.
Display Customization		
	--display-icon=/path/icon.png	Location of the icon image, if the local path contains any special characters or spaces then you should surround the text with single ' quotes.
Deferral Behavior		
	--focus-defer=7200	The number of seconds to defer automatically if the system is in user-enabled Focus/Do Not Disturb or when a process has requested that the display not go to sleep
	--menu-defer=300,1800,3600,7200	Display a deferral time pop-up menu in the non-deadline update restart dialog that allows the user to override the default
	--Error-defer=7200	The number of seconds to defer if <code>super</code> detects an error in the workflow
	--recheck-defer=86400	The number of seconds to defer if no software updates are available or allowed. Enabling this option results in <code>super</code> acting as a permanent agent that checks for software updates on a regular basis.
	--delete-deferrals	This option can not be set via a MDM configuration profile. However, any other deferral options that are specified via a <code>super</code> MDM configuration profile remain in effect.
Deferral Count Deadlines		
	--focus-count=5	The maximum number of automatic deferrals allowed if the system is in user-enabled Focus/Do Not Disturb or when a process has requested that the display not go to sleep
	--soft-count=5	The maximum number of user selected deferrals allowed before showing a soft deadline dialog
	--hard-count=5	The maximum number of user selected deferrals allowed before the computer automatically restarts for updates without asking the user for approval
Deferral Days Deadlines		
	--focus-days=3	The maximum number of days that automatic deferrals are allowed if the system is in user-enabled Focus/Do Not Disturb or when a process has requested that the display not go to sleep
	--soft-days=5	The maximum number of deferral days allowed before showing a soft deadline dialog.
	--hard-days=7	The maximum number of days allowed before before the computer automatically restarts for updates without asking the user for approval.
	--zero-day=2022-09-01:12:00	Instead of having the days deadline counter automatically select the day zero date, this option sets a specific date and time as day zero.

Deferral Date Deadlines		
	--focus-date=2022-09-03:12:00	The last date and time when automatic deferrals are allowed if the system is in user-enabled Focus/Do Not Disturb or when a process has requested that the display not go to sleep
	--soft-date=2022-09-05:12:00	The last date and time before showing a soft deadline dialog.
	--hard-date=2022-09-07:12:00	If this date and time have passed the computer automatically restarts for updates without asking the user for approval.
Apple Silicon Local Credentials		
	--local-account='labadmin' --local-password='ThisIs@Test'	An existing local (standard or admin) user account name and password with volume ownership privileges that can be used to authenticate the local <code>softwareupdate</code> command.

MDM Configuration Profile

If there are specific `super` options you plan to set "permanently" then you should consider deploying these settings via a MDM configuration profile. In addition to over-the-air deployment, using a MDM configuration profile also allows you to enforce your options. In other words, if a specific `super` option is deployed via a MDM configuration profile then it cannot be ignored or changed via local command options.

The MDM configuration profile specification allows for custom settings deployed via application specific preference domains. In the case of `super`, the preference domain is `com.macjutsu.super`.

 Please note: Do not deploy the example `.mobileconfig` file, as this is an example for all options and contains conflicts with other settings that can cause errors if deployed as is.

Example options for the `.mobileconfig` listed below.

▼ Example `.mobileconfig`

```
<?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">
  <dict>
    <key>PayloadUUID</key>
    <string>D819E6B3-72BA-4202-874D-FE7C6783C984</string>
    <key>PayloadType</key>
    <string>Configuration</string>
    <key>PayloadOrganization</key>
    <string>Macjutsu</string>
    <key>PayloadIdentifier</key>
    <string>D819E6B3-72BA-4202-874D-FE7C6783C984</string>
    <key>PayloadDisplayName</key>
    <string>S.U.P.E.R.M.A.N. All Settings Example</string>
    <key>PayloadDescription</key>
    <string>This is an example of all possible super managed preferences. DO NOT DEPLOY. THIS EXAMPLE CONTAINS
    CONFLICTING SETTINGS THAT WILL CAUSE ERRORS IF DEPLOYED.</string>
    <key>PayloadVersion</key>
    <integer>1</integer>
    <key>PayloadEnabled</key>
    <true/>
    <key>PayloadRemovalDisallowed</key>
    <true/>
    <key>PayloadScope</key>
    <string>System</string>
    <key>PayloadContent</key>
    <array>
      <dict>
        <key>PayloadDisplayName</key>
        <string>Custom Settings</string>
```



```
<key>PayloadIdentifier</key>
<string>22539B64-64B4-4973-AA52-B6B105B4C014</string>
<key>PayloadOrganization</key>
<string>Software</string>
<key>PayloadType</key>
<string>com.apple.ManagedClient.preferences</string>
<key>PayloadUUID</key>
<string>22539B64-64B4-4973-AA52-B6B105B4C014</string>
<key>PayloadVersion</key>
<integer>1</integer>
<key>PayloadContent</key>
<dict>
  <key>com.macjutsu.super</key>
  <dict>
    <key>Forced</key>
    <array>
      <dict>
        <key>mcx_preference_settings</key>
        <dict>
          <key>AllowRSRUpdates</key>
          <true/>
          <key>AllowUpgrade</key>
          <true/>
          <key>BatteryLevel</key>
          <string>50</string>
          <key>BatteryTimeout</key>
          <string>3600</string>
          <key>DefaultDefer</key>
          <string>120</string>
          <key>DeferDialogTimeout</key>
          <string>60</string>
          <key>DisplayAccessoryDefault</key>
          <string>https://raw.githubusercontent.com/Macjutsu/super/main/Super-Friends/Display-
Accessory-Example.html</string>
          <key>DisplayAccessoryType</key>
          <string>HTML</string>
          <key>DisplayAccessoryUpdate</key>
          <string>https://raw.githubusercontent.com/Macjutsu/super/main/Super-Friends/Display-
Accessory-Example.html</string>
          <key>DisplayAccessoryUpgrade</key>
          <string>https://raw.githubusercontent.com/Macjutsu/super/main/Super-Friends/Display-
Accessory-Example.html</string>
          <key>DisplayAccessoryUserAuth</key>
          <string>https://raw.githubusercontent.com/Macjutsu/super/main/Super-Friends/Display-
Accessory-Example.html</string>
          <key>DisplayIcon</key>

<string>/System/Library/CoreServices/CoreTypes.bundle/Contents/Resources/BurningIcon.icns</string>
          <key>DisplayRedraw</key>
          <string>20</string>
          <key>DisplaySilently</key>
          <false/>
          <key>EnforceNonSystemUpdates</key>
          <true/>
          <key>ErrorDefer</key>
          <string>120</string>
          <key>FocusCount</key>
          <string>5</string>
          <key>FocusDate</key>
          <string>2022-07-01</string>
          <key>FocusDays</key>
          <string>2</string>
          <key>FocusDefer</key>
          <string>120</string>
          <key>FreeSpaceTimeout</key>
          <string>3600</string>
          <key>FreeSpaceUpdate</key>
          <string>15</string>
          <key>FreeSpaceUpgrade</key>
          <string>35</string>
```

```

<key>HardCount</key>
<string>5</string>
<key>HardDate</key>
<string>2022-07-03:03:03</string>
<key>HardDays</key>
<string>6</string>
<key>HelpButton</key>
<string>https://support.apple.com/en-us/HT201541</string>
<key>IconSizeIbm</key>
<string>128</string>
<key>IconSizeJamf</key>
<string>128</string>
<key>InstallNow</key>
<true/>
<key>JamfProID</key>
<string>$JSSID</string>
<key>MenuDefer</key>
<string>120,200,300</string>
<key>OnlyDownload</key>
<true/>
<key>PolicyTriggers</key>
<string>trigger1,trigger2</string>
<key>PreferJamfHelper</key>
<false/>
<key>RecheckDefer</key>
<string>120</string>
<key>RestartWithoutUpdates</key>
<true/>
<key>SkipUpdates</key>
<true/>
<key>SoftCount</key>
<string>5</string>
<key>SoftDate</key>
<string>2022-07-02:02:02</string>
<key>SoftDays</key>
<string>4</string>
<key>SoftDialogTimeout</key>
<string>60</string>
<key>TargetUpgrade</key>
<string>12</string>
<key>TestMode</key>
<true/>
<key>TestModeTimeout</key>
<string>60</string>
<key>UserAuthMDMFailover</key>
<string>HARD,INSTALLNOW,BOOTSTRAP</string>
<key>UserAuthTimeout</key>
<string>600</string>
<key>VerboseMode</key>
<true/>
<key>WarningButton</key>
<string>https://support.apple.com/en-us/HT201222</string>
<key>ZeroDay</key>
<string>2022-07-01</string>
</dict>
</dict>
</array>
</dict>
</dict>
</dict>
</array>
</dict>
</plist>

```

Uninstalling S.U.P.E.R.M.A.N.

The [Super-Friends folder contains a `Remove-Super.sh` script](#) that deletes all `super` related items except for the `erase-`

`install.sh` items used to facilitate macOS upgrade workflows.

However, if `erase-isntall.sh` was used as part of a macOS upgrade workflow it too can be easily removed by deleting the contents of the `/Library/Management/erase-install/` folder.

The script has been included in the Fileset on this KB article as well so you can have it executed simply by removing the Association of the Fileset to your devices. If you don't want that behavior then simply remove the script from the Fileset.

Related Content

- [Home · Macjutsu/super Wiki \(github.com\)](#)
- [Getting Started · Macjutsu/super Wiki · GitHub](#)
- [MDM profile example - Macjutsu/super Wiki](#)

🔄Revision #47

★Created 19 July 2023 18:08:35 by Andrew Kloosterhuis

✍Updated 11 April 2024 17:04:14 by Zachary Butterfield