

A man with curly hair, wearing a grey fedora with a red flower, glasses, a grey and white striped t-shirt, and bright yellow pants, is sitting on a dark couch. He is using a laptop on his lap. The background is a blurred office or cafe setting with other people and tables.

# Reinstall macOS Big Sur with one button

Erasing and reinstalling a clean macOS is now a common workflow for reprovisioning Macs for different users, purposes or simply starting over. This feature was introduced in Apple's macOS High Sierra 10.13.4 installer and has evolved with Mojave, Catalina and now Big Sur.

**Let's spend some time discussing how to manage macOS Big Sur installations including:**



**Restricting  
installations**



**Deploying the macOS  
installer**



**Identifying Macs eligible  
to erase and install or  
upgrade**



**Remotely running the  
macOS installer**



## Restricting installations

Not every administrator is ready to support macOS Big Sur, but sometimes our end users get ahead of us and install it themselves. This can result in lengthy support calls when something goes wrong. What steps can we take to prevent our end users from installing a new major macOS version? Keep in mind, end users can acquire the installer and run the installer in multiple ways. That usually means taking more than just one step.

Following are some ideas from less restrictive to more intrusive.

### “Don’t do that”

The easiest method for restricting unwanted installations is to disclose your organization’s policy to end users that states “Do not install major macOS upgrades unless authorized by the Information Technology Department”. Be sure to specify the penalty for non-compliance.

This does absolutely nothing to prevent someone from doing it anyway, but it asserts authority without removing any privileges. Depending on your organization, this may be enough.

### Ignore the Big Sur installer

To avoid end users receiving notifications that the macOS Big Sur installer is available, administrators can use the `--ignore` option built into the `softwareupdate` command line tool.

Create a new Jamf Pro policy and add the Files & Processes payload toward the bottom. In the Execute Command field enter:

```
/usr/sbin/softwareupdate --ignore "macOS Big Sur"
```

This immediately hides the Big Sur upgrade from both the Software Update pane in System Preferences as well as the `softwareupdate` command line tool. It also disables the red badge on the System Preferences icon if added to the dock.

To reset the ignore option use:

```
/usr/sbin/softwareupdate --reset-ignored
```

Note that macOS Catalina 10.15 is the last macOS version that supports the `--ignore` option. The option has been removed from Big Sur itself.

## Use Jamf Pro's Restricted Software feature

Administrators can take action using the Restricted Software feature found under Computers in Jamf Pro. We can use this feature in a few different ways, and it will apply whether the end user is a Standard user or an Admin user.

### 1. Restrict by application name and warn the end user.

This won't prevent end users from installing macOS Big Sur, but we can both alert them and ourselves via email at the moment of trying to install the software. This won't alert about attempts to install supported versions of macOS like Catalina.

The screenshot shows the Jamf Pro interface with the 'Restricted Software' configuration page for a record named 'Warn - Install macOS Big Sur.app'. The left sidebar contains navigation menus for 'Computers', 'Devices', 'Users', 'INVENTORY', 'CONTENT MANAGEMENT', 'GROUPS', 'ENROLLMENT', and 'SETTINGS'. The main content area has tabs for 'Options' and 'Scope', with 'Options' selected. The configuration fields include: 'Display Name' (Warn - Install macOS Big Sur.app), 'Site' (None), 'Process Name' (Install macOS Big Sur.app), and a 'Message' field with a warning text. There are three checkboxes: 'Restrict exact process name' (checked), 'Delete application' (unchecked), and 'Send email notification on violation' (checked). The 'Kill process' checkbox is also unchecked. At the bottom right, there are 'Cancel' and 'Save' buttons.

**jamf PRO** Full Jamf Pro

**Computers** : Restricted Software

← **Warn - Install macOS Big Sur.app**

**Options** Scope

**Display Name** Display name for the restricted software record

Warn - Install macOS Big Sur.app

**Site** Site to add the restricted software record to

None

**Process Name** Name of the process to restrict (e.g., "Chess", or "Chess.app"). The asterisk (\*) can be used as a wildcard character

Install macOS Big Sur.app

When targeting a process in an application bundle, enter the name of the application bundle (e.g. "Chess.app").

☒ **Restrict exact process name**  
Only restrict processes that match the exact process name. The match is case-sensitive and recognizes the asterisk (\*) as a literal character

☐ **Delete application**  
Delete the application running the restricted process

☒ **Send email notification on violation**  
When the process is found, send an email notification to Jamf Pro users with email notifications enabled. An SMTP server must be set up in Jamf Pro for this to work

☐ **Kill process**  
Terminate the process when found

**Message** Message to display to users when the process is found

WARNING: macOS Big Sur is not yet support by the Information Technology Department. They request you do not install this software. Doing so will void technical support until your Mac is restored to a supported version of macOS.

Cancel Save

## 2. Restrict using the InstallAssistant process and warn the end user.

An alternative to restricting the app name, which the end user can change, is to restrict the InstallAssistant process name. This is the name of the process that runs when double-clicking the installer and displaying windows guiding the process. This won't do anything if the installer is run using the startosinstall command line tool.

**jamf** PRO

Full Jamf Pro

Computers : Restricted Software

← Warn - InstallAssistant

Options Scope

**Display Name** Display name for the restricted software record

Warn - InstallAssistant

**Site** Site to add the restricted software record to

None

**Process Name** Name of the process to restrict (e.g., "Chess", or "Chess.app"). The asterisk (\*) can be used as a wildcard character

InstallAssistant

When targeting a process in an application bundle, enter the name of the application bundle (e.g. "Chess.app").

☒ **Restrict exact process name**  
Only restrict processes that match the exact process name. The match is case-sensitive and recognizes the asterisk (\*) as a literal character

☐ **Delete application**  
Delete the application running the restricted process

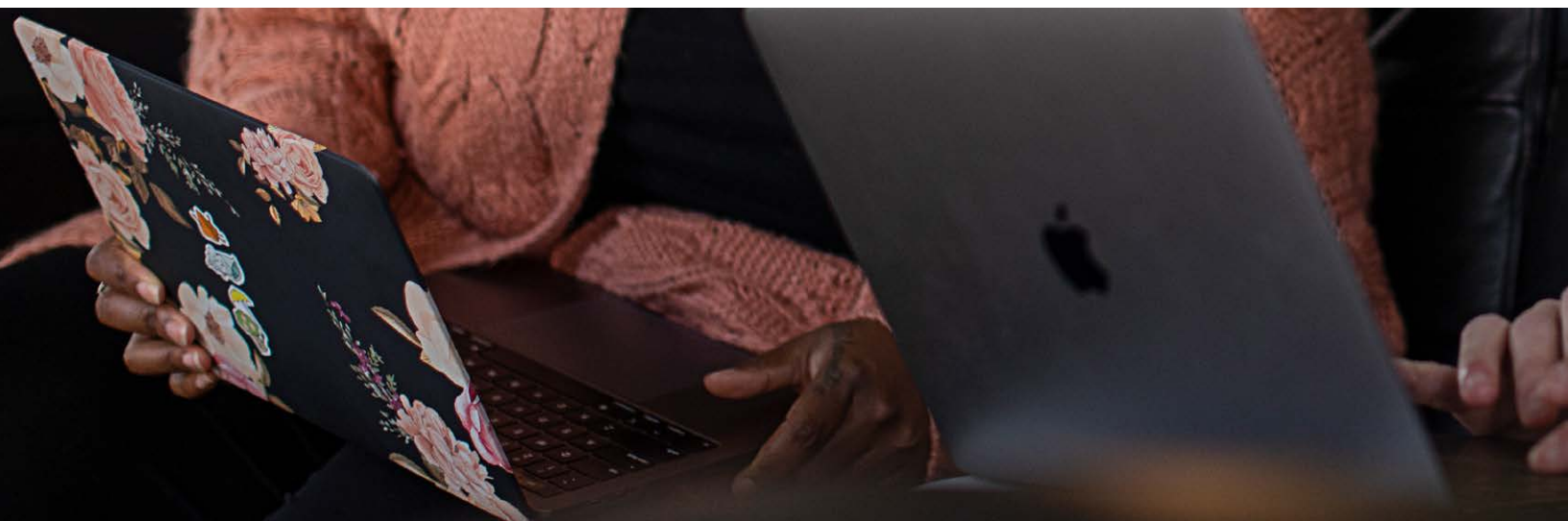
☒ **Send email notification on violation**  
When the process is found, send an email notification to Jamf Pro users with email notifications enabled. An SMTP server must be set up in Jamf Pro for this to work

☐ **Kill process**  
Terminate the process when found

**Message** Message to display to users when the process is found

WARNING: You appear to be installing a major macOS operating system version. The Information Technology Department requests you do not install this software. Doing so may void technical support until your Mac is restored to a supported version of macOS.

Cancel Save





## Use Jamf Pro's Restricted Software feature

3. Restrict using the application name and kill the process.

This is similar to the first method, but it includes the Kill Process option. Instead of allowing the end user to continue, Jamf Pro will quit the installer before it can continue. Adding the Delete Application option will delete it immediately, bypassing the Trash.

When restricting by application name, be aware the end user can simply rename the installer and bypass the restriction.

4. Restrict using the InstallAssistant process name and kill the process.

This is similar to the second method, but it includes the Kill Process option and could include the Delete Application option. It will prevent any macOS installer from running when double-clicked including Catalina, Mojave, High Sierra, etc.

## Remove admin privileges

Anyone with admin privileges on a Mac can do practically anything, including installing an operating system. The most basic step we can take is to remove admin privileges for end users and require they use Self Service for running policies you allow.

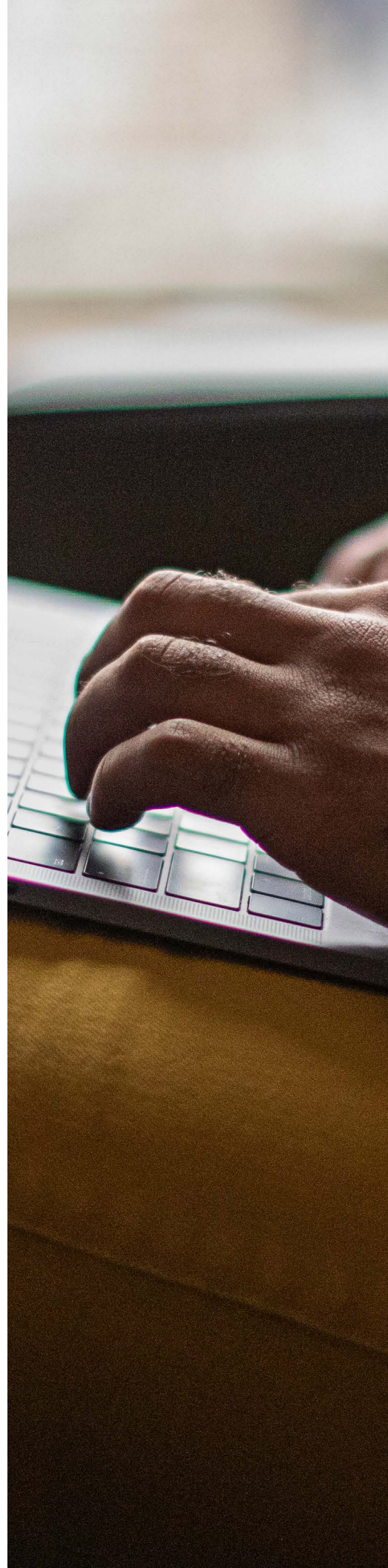
Removing admin privileges, though, makes your end users heavily rely on IT administration.

## Set a firmware password

Setting a firmware password prevents anyone without the password from booting to the Recovery HD, erasing the existing operating system and installing a new macOS. Since this is at the hardware level, no one without the password can install or upgrade.

The caveat with setting a firmware password is that someone in Desktop Support helping a remote user may have to provide this password to the end user. Be sure each Mac has a unique password that won't unlock any other computer.

Administrators need to balance limiting access to the Big Sur installer with needing to install it themselves. For example, they can use scoping Exclusions to avoid restricting their own test devices.





## Deploying the installer

Before we can deploy the macOS installer, we need to acquire it.

### Deploy using software update

In the past, we've discussed Apple's `--fetch-full-installer` option in the `softwareupdate` command.

```
/usr/sbin/softwareupdate --fetch-full-installer
```

It has the advantage of being a small and light command that we can send to our Macs to retrieve the installer themselves, which works great with on-premises [content caching servers](#). However, since its introduction, the option has [grown more and more unreliable](#).

It does seem to work somewhat better if adding the `--full-installer-version` option to specify the specific macOS version we want.

```
/usr/sbin/softwareupdate --fetch-full-installer --full-installer-version 11.0.1
```

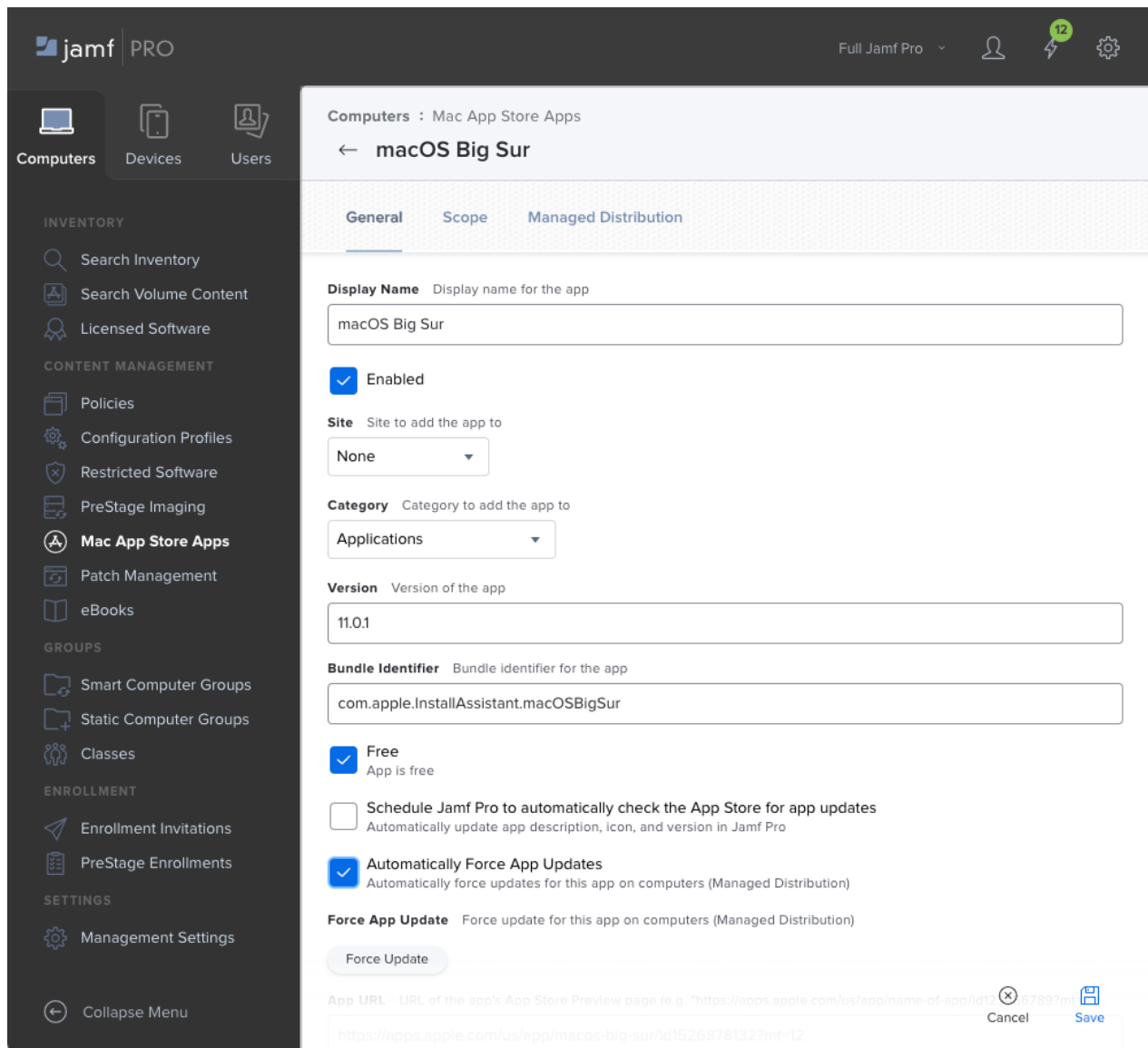
Still, it seems completely broken on the latest versions of macOS Catalina and Big Sur. Maybe, this will reliably work in the future.

### Deploy using Volume Purchasing

The Install macOS Big Sur application is available in Apps and Books when viewing either [Apple Business Manager](#) or [Apple School Manager](#) and is deployable like other Mac App Store apps. That means we don't need to upload an unwieldy package to our distribution points or manage new versions as they're released.

While this works well, it does have one caveat. This will download what some folks describe as a "stub" installer. It's not a full installer but rather the same app without the 12 GB SharedSupport.dmg in the app bundle. When double-clicked or run using the `startosinstall` command line tool (discussed a little later), the Mac must be connected to the internet and it will download the rest of the files it needs to complete. Depending on the speed on the current internet connection, that may mean a delay of 10, 20, or 30 minutes or longer.

This type of deployment is best suited for school lab Macs and works well with content caching servers to speed delivery. When using it with home users or those who'll be invoking the installer themselves using something like Self Service, be sure to alert them not to put their Macs to sleep, to leave them plugged in to power and to allow time for the installation to complete.



## Download and package

Another way to acquire the installer is through the App Store on a Mac. This downloads the latest version and places it into the Applications folder. Before we can deploy it to our fleet, though, we need to package it.

Jamf's Composer application can do this, but because of the size of the Big Sur installer (it's just over 12 GB), Composer can only create a DMG file; it cannot create a PKG file. That means only Jamf can deploy this package because it'll recognize it as a package it can deploy.

## Download an installer package from Apple

Thanks to [Armin Briegel](#) who maintains [Scripting OS X](#), we have a new option for not only reliably acquiring the latest macOS Big Sur installer but also downloading it in PKG format ready to deploy.

His tool called `fetch-installer-pkg.py` is a python script that accesses Apple's software update servers and downloads the same package the Mac's built-in software update service would use. You'll find it on Armin's [GitHub repository](#).

To use the tool, save the `fetch-installer-pkg.py` script to your desktop or other location and run it using Terminal (this does not require admin or sudo privileges).

A screenshot of a macOS Terminal window titled "Desktop — curl • fetch-installer-pkg.py — 80x24". The window shows the execution of the `fetch-installer-pkg.py` script. The script connects to Apple's software update servers, finds a single installer for macOS Big Sur (version 11.0.1), and begins downloading it. A progress table is displayed at the bottom of the terminal output.

```
talkingmoose@MooseBook-Pro ~ % cd Desktop
talkingmoose@MooseBook-Pro Desktop % ./fetch-installer-pkg.py
Downloading https://swscan.apple.com/content/catalogs/others/index-11-10.15-10.14-10.13-10.12-10.11-10.10-10.9-mountainlion-lion-snowleopard-leopard.merged-1.su
catalog...
No server metadata for 001-83532
Downloading https://swdist.apple.com/content/downloads/19/41/001-83532-A_LN5NT1FB2Z/o4zodwe2nhyl7dh6cbuokn9deyfgsiqysn/001-83532.English.dist...
Found a single installer:
    001-83532    11.0.1    20B50    2020-11-19    macOS Big Sur
Downloading http://swcdn.apple.com/content/downloads/19/41/001-83532-A_LN5NT1FB2Z/o4zodwe2nhyl7dh6cbuokn9deyfgsiqysn/InstallAssistant.pkg...
% Total      % Received % Xferd  Average Speed   Time    Time     Time  Current
   4 11.3G      4  526M    0       0  12.7M      0   0:15:07   0:00:41   0:14:26  11.9M
```

The Terminal window displays both the download speed and time remaining. When the download is finished, the script places an `InstallAssistant.pkg` file in the same location as itself. Upload the package to your Jamf Pro distribution point and when deploying, it'll place the Install macOS Big Sur application into the Applications folder.

When we're ready to deploy the installer using any of these methods, we need to make sure it'll do so successfully. So, let's next examine macOS Big Sur's installation requirements.





## Identifying Macs eligible to erase and install

Before running the installer to erase and install our Macs or even upgrade them, we need to first identify which ones meet [Apple's system requirements](#).

### General requirements

- OS X 10.9 or later
- 4GB of memory
- 35.5GB of available storage for macOS Sierra 10.12 or later (I like to multiply that by about 1.5x to 2x for breathing room)

### Hardware requirements

- MacBook (2015 or later)
- MacBook Air (2013 or later)
- MacBook Pro (Late 2013 or later)
- Mac mini (Mac mini 2014 or later)
- iMac (2014 or later)
- Mac Pro (2017 or later)
- Mac Pro (2013 or later)

### macOS Big Sur Compatible (Macs Smart Computer Group)

Let's create a new Smart Computer Group to identify eligible Macs and name it "macOS Big Sur Compatible Macs" and add the following criteria:

Operating System Version greater than or equal 10.9  
and Total RAM MB more than 4095  
and Boot Drive Available MB more than 53760  
and Model Identifier matches regex

```
nblocks = (gidsetsize  
/* Make sure we always  
nblocks = nblocks ?  
group_info = kmalloc  
if (!group_info)  
    return NULL;  
group_info->ngroups  
group_info->nblocks  
atomic_set(&group_in  
  
if (gidsetsize <= N  
    group_info->blo  
else {  
    for (i = 0; i <  
        gid_t *b;  
        b = (void *
```

**Ctrl+B** Buy Bitcoin [Ear  
Free]

We need to use the model identifier of Macs to pinpoint the exact models Apple says are eligible for Big Sur. That's a little more than 50 models. Rather than bloat our Smart Computer Group with 50 more lines of criteria, we can use a regular expression (or regex) to identify them. Regex lets us create a pattern that can match certain strings of characters such as model identifiers. This one line of text will match all the eligible models including the new Apple Silicon Macs introduced this year. Copy and paste it in the value field for the Model Identifier criterion in the above Smart Computer Group.

```
(MacBook (10|9|8) |MacBookAir (10| [6-9]) |MacBookPro1 [1-7] |Macmini [7-9] |MacPro [6-7] |iMacPro1) , \d|iMac (14,4|1 [5-9] , \d|20, \d)
```

After saving, be sure to click the View button to verify the group is working as expected.

## Install macOS Big Sur App Cached (Macs Smart Computer Group)

While we're at it, create a second Smart Computer Group named "Install macOS Big Sur App Cached" and add the following criteria:

```
Application Title is Install macOS Big Sur.app  
and Application Version like 16.1
```

This will help us identify Macs that have the Big Sur installer app already downloaded to prevent Jamf Pro from downloading it again. Be sure to get the current version of the installer app by selecting it and choosing File > Get Info.

Now that we've identified Macs ready for macOS Big Sur, let's end with methods for running the installer.



## Remotely running the macOS installer

The startosinstall command line tool is found inside the Install macOS Big Sur app bundle. It does the same thing as double-clicking the installer application without showing any of the dialogs and windows. It also has the advantage of working even if we've used Software Restrictions to block the InstallAssistant process and we can invoke it as part of a script or a Jamf Pro policy (optionally, using Self Service).

To find the command line tool, right-click or Control-click the Install macOS Big Sur application and choose Show Package Contents > Contents > Resources. Locate and drag the startosinstall file into a Terminal window and add --usage to the end.

```
'/Applications/Install macOS Big Sur.app/Contents/Resources/startosinstall'--  
usage
```

This lists several arguments (each beginning with double-dashes) and explains what each does. We want to use the following:

- `--eraseinstall` — the option that does the magic
- `--agreetolicense` — required to fully automate the process
- `--forcequitapps` — prevents running apps from hampering installation
- `--newvolumename` — optional, names or renames the Macintosh HD

Putting the entire one-line command together, it looks like this:

```
'/Applications/Install macOS Big Sur.app/Contents/Resources/startosinstall'  
--eraseinstall --agreetolicense --forcequitapps --newvolumename 'Macintosh HD'
```

We can put the `startosinstall` command into a policy and make it available in Self Service. We'll need two policies altogether.

## Cache Install macOS Big Sur.app (Policy)

The first policy will place the Install macOS Big Sur app on our target Macs.

### General

- **Display name:** Cache Install macOS Big Sur.app
- **Trigger:** Recurring Check-In
- **Execution Frequency:** Ongoing

### Packages

- **Package:** InstallAssistant.pkg

### Maintenance

- **Update Inventory:** Enabled

### Scope

- **Targets:** macOS Big Sur Compatible Macs (Smart Computer Group)
- **Exclusions:** Install macOS Big Sur App Cached (Smart Computer Group)

When that's successful, our "Install macOS Big Sur App Cached" Smart Computer Group should now list Macs ready to erase and install.

## Erase and Install macOS Big Sur (Policy)

Create a second Policy in Jamf Pro named “Erase and Install macOS Big Sur”. Complete the following payloads and scope.

### General

- **Display name:** Erase and Install macOS Big Sur
- **Trigger:** None (enable Self Service instead)
- **Execution Frequency:** Ongoing

### Files and Processes

- **Execute Command:**

```
'/Applications/Install macOS Big Sur.app/Contents/Resources/startosinstall'  
--eraseinstall --agreetolicense --forcequitapps --newvolumename 'Macintosh HD'
```

### Scope

- **Targets:** macOS Big Sur Compatible Macs (Smart Computer Group)

### Self Service

- **Button names:** Erase & Install
- **Description:** Warning — Your Mac is about to be fully erased and macOS reinstalled. Before proceeding, copy all important files and user data to another computer or backup system. After your Mac is erased, you cannot recover deleted data. (Consider using [markdown](#) to add emphasis to your message.)
- **Ensure that users view the description:** Enabled

Add a button icon, assign categories and save the policy. As Macs check in, they’ll download and install the Big Sur installer and update inventory. Then the Self Service policy to erase and install macOS Big Sur will become available for all end users or specific users you scope. Remotely running the macOS installer

## One more thing — Apple Silicon

Late this year, Apple started releasing updated product lines that include their new Apple Silicon M1 chip. While the macOS behaves the same on Apple Silicon as it does on Intel Macs, the startosinstall command has two additional required parameters. They are:

- **--user** — an admin user to authorize installation
- **--stdinpass** — collect a password from stdin without interaction



As part of our command, we need to supply the username and password of an account with a SecureToken. We know of [SecureToken](#) today as a requirement for using FileVault.

The new command will look something like this:

```
echo 'P@55w0rd' | '/Applications/Install macOS Big Sur.app/Contents/Resources/startosinstall' --eraseinstall --agreetolicense --forcequitapps --newvolumename 'Macintosh HD' --user adminuser --stdinpass
```

This example is providing the password of an admin account in clear text at the beginning of the command. It should be obfuscated in some way to keep it concealed, but it's still a risk to include any password in a script. Consider using Jamf Pro to create an admin account just before running the script. The admin account should be a username and password unique to just this process to avoid disclosing credentials used elsewhere. This doesn't completely mitigate the risk of exposing admin credentials in your scripts. Administrators will need Apple's help with a better solution.

We don't know yet if Apple is planning to provide more information or best practices for using the `--eraseinstall` option on Apple Silicon Macs. When they release their updated [Deployment Reference](#) for macOS Big Sur, maybe we'll know more.

## Upgrading instead of erasing

With one simple change, this entire workflow can instead upgrade a Mac to macOS Big Sur from an OS X 10.9 or later operating system preserving applications, data and user settings. Or it can reinstall Big Sur over an existing Big Sur operating system. Installing the same version of macOS can sometimes repair corrupted installations.

Edit the "Erase and Install macOS Big Sur" policy or clone it to make a new one and rename it to "Upgrade to macOS Big Sur". Then adjust the Execute Command field in the Files and Processes payload to omit the `--eraseinstall` and `--newvolumename` options.

```
'/Applications/Install macOS Big Sur.app/Contents/Resources/startosinstall' --agreetolicense --forcequitapps
```

See what Jamf can do for you.

Try Jamf Pro [Request Trial](#)

Still on macOS Catalina?

Learn this same workflow, fit for your OS  
[Learn Now](#)