



Inventory Management 201:

Take the next steps with Jamf

We all have inventory: hardware, operating system versions, serial numbers, application licenses.

Keeping track of which devices are given to whom, and when, along with all the relevant details might be easy when your organization is small. But as an organization grows, this process becomes increasingly labor intensive, and the data itself risks becoming stale. (Did I remember to check which iPads got an OS upgrade?)

As businesses and schools increasingly turn to remote work/distanced learning, having physical access to each device to check and update inventory becomes even more difficult. Once you factor in the nature of remote work — and the corresponding need for a focus on security and compliance — you realize you have to do something more than tracking hardware, software and configuration inventory manually in a spreadsheet.

Enter Jamf Pro. Jamf Pro automatically collects hundreds of data points about your Apple devices — both “out of the box” info about the devices and what’s installed on them, as well as customized data with Jamf Pro’s extension attributes.

Inventory should start with enrollment

From the moment a user touches a device, your inventory software should be collecting information on the device and placing it into an easily accessible database by apps, device type, OS, permissions and packages.

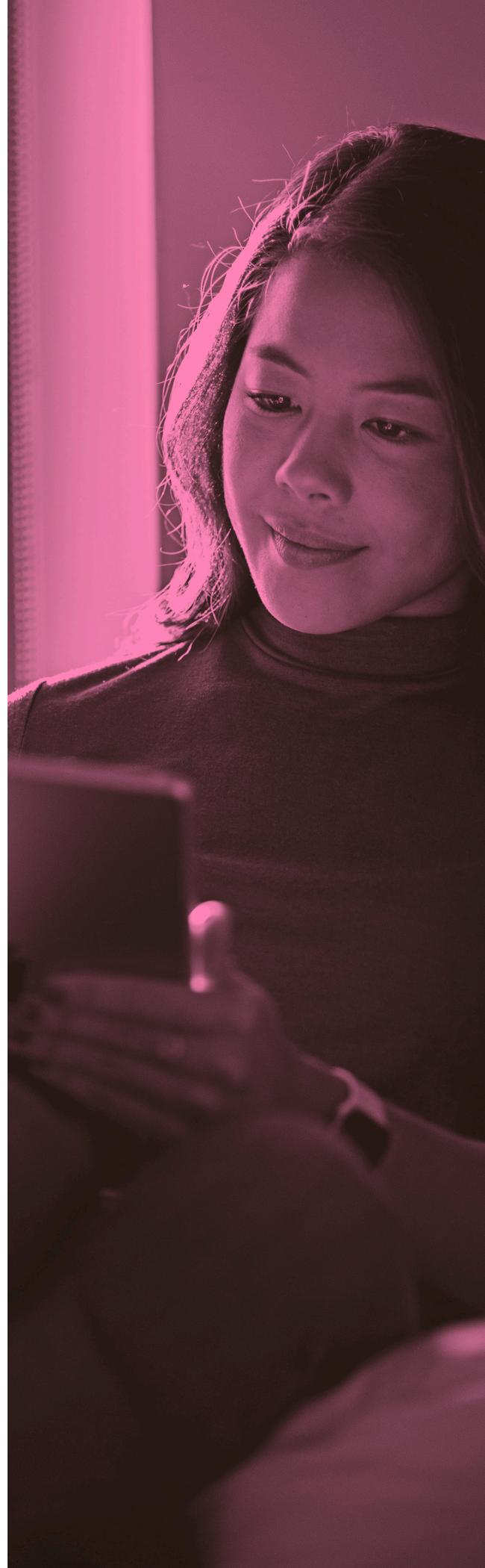
The best way to do this for Apple devices is to pre-enroll your devices with Apple Business Manager, and ensure that Jamf Pro can continue to automate device tracking throughout the life of that device.

Once you have all of this information Jamf Pro has gathered, you can take action based on all manner of inventory data.

What is inventory management? A brief recap of ideas and terms

Computer [inventory management](#) is a set of best practices and workflows that track an organization's IT devices and their contents. The best inventory management should answer these questions for you and more:

- How many devices do you actually have? What types of devices?
- Are they all in use, or are some simply stacked in a closet, depreciating?
- Are all of your organization's devices up to date?
- Which apps are on which devices?
- Who has access to what?
- Does anyone have an overloaded machine, hasn't updated their password or hasn't performed a backup?
- Is your fleet secure? Are you meeting all of your [CIS benchmarks](#)?
- Who has required protection software, such as Jamf Protect, and who has not?



Inventory management terms

Fleet: all of the devices you're managing (macOS, iOS, iPadOS and tvOS devices).

Inventory: attributes of your fleet of devices, such as:

- Device Type: MacBook Pro, iPad, iPhone, Apple TVs
- FileVault 2 Partition Encryption State: Encrypted, Not Encrypted
- Last Inventory Update: (this will show the date/time the device last updated information)

Static Groups: devices in groups chosen manually by an administrator; the membership does not change unless the admin changes it intentionally.

Smart Groups: devices in Smart Groups change dynamically based on criteria pulled from inventory. For example, you could create a Smart Group for "Find all macOS devices that are FileVault encrypted" or "find iPads running any version of iOS 14." As your fleet changes, your Smart Group memberships change based on how they fit into that criteria, automatically.

Extension attributes: additional inventory data populated manually or via a script. A few examples: "a date field reflecting the date a device was retired from inventory" (this could be manually entered as a part of the deprovisioning workflow) or "an integer field detailing the binary version of a tool/app that normally isn't reflected in the Applications folder" (populated via script and dynamically changed whenever the app is updated).

Scope: defining which devices receive which management components (apps, configurations, policies, restrictions, etc.)

Configuration profiles: settings that use Apple's mobile device management (MDM) framework and cannot be changed by end users (i.e. "deploy a setting to require the macOS screen saver starts after 15 minutes of inactivity" or "deploy the office Wi-Fi settings to all iPads). You can manage iOS, iPadOS and tvOS entirely by profiles.

Policies: settings and application deployments for macOS devices that fall outside the MDM framework. The Jamf binary handles policies and accounts for the things that MDM doesn't do for macOS (i.e. "run a script on macOS devices").

Supervised device: supervision of a device provides a higher level of MDM control. It's not required for a device to be managed by Jamf Pro, but more options are available for control and restriction of supervised devices than those that are not supervised. In macOS 11 or later, a Mac computer is considered supervised when a user performs a device enrollment into MDM.

When the serial numbers of the devices appear in Apple School Manager, Apple Business Manager or via Apple's Automated Device Enrollment, they are automatically supervised when enrolled in Jamf Pro:

- iPhone and iPod touch with iOS 13 or later
- Apple TV with tvOS 13 or later
- iPad with iPadOS 13.1 or later
- Mac computers with macOS 10.14.4 or later



Application Programming Interface (API): a programming language that allows apps to speak to each other to accomplish tasks. A script can use the Jamf Pro API to read and update information on a device record automatically without end-user interaction.

Webhooks: a way for one app to provide other apps with real-time information. APIs require an app or script to reach out for data very frequently in order to get timely information. By listening for events to happen, webhooks make sharing information between apps much more efficient.



If you can track it, you can take action on it!

Tracking the status of your devices provides you all sorts of reporting capabilities:

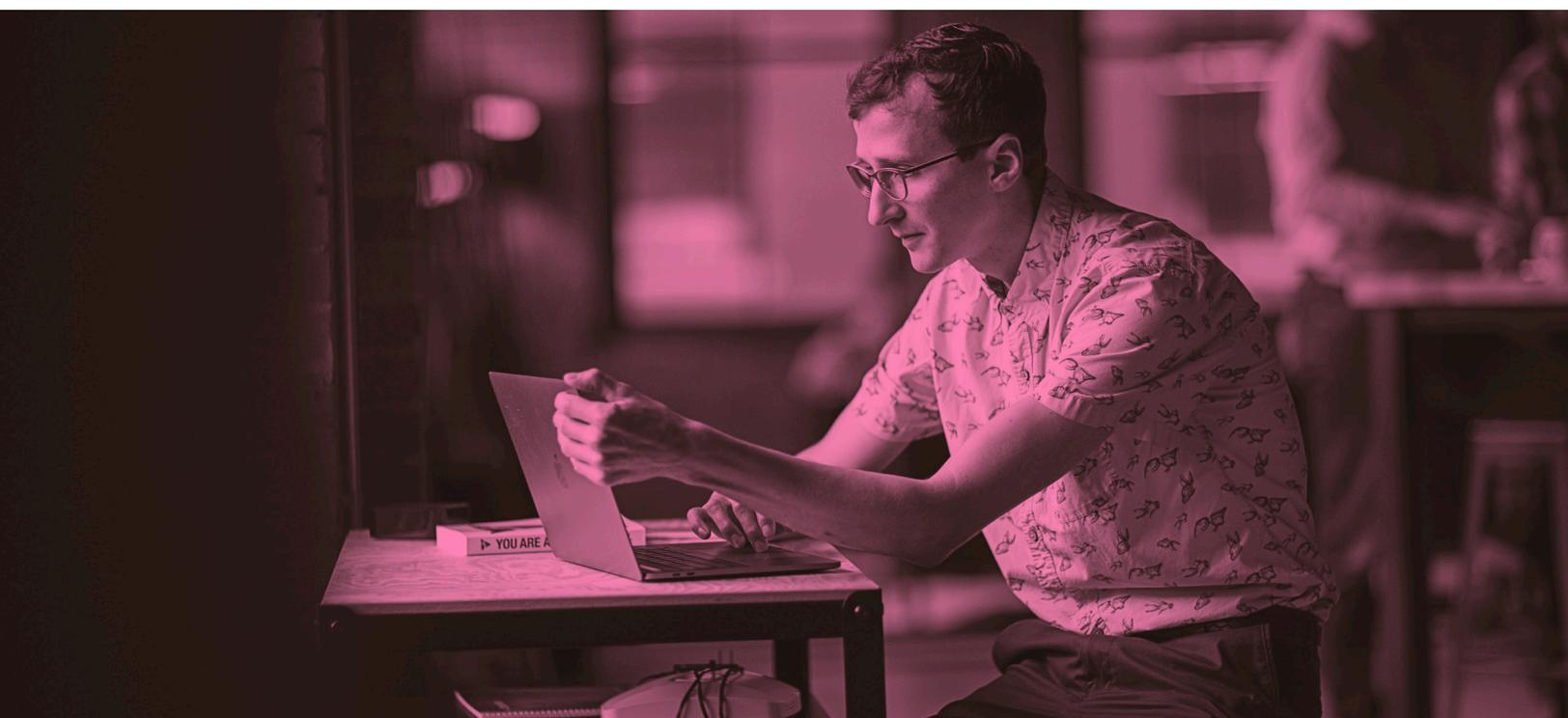
- Report on devices that are coming off warranty to find out how many MacBooks you have that might be in need of replacement.
- A report sorted by version will show you which operating systems are most prevalent in your environment
- Concerned that folks might have a vulnerable version of a major application? Report on and prioritize your updating.



But say you didn't just want reports - what if you wanted to actually take action?

The same approach that allows you to report also allows you to remediate:

- If you find devices running older software, you can push out an update.
- Out-of-date operating systems? Serve up the update in Self Service, and notify users via the Notification Center.
- Want to force device compliance for your organization's security requirements? Pair Jamf Pro with [Jamf Protect](#): endpoint protection built exclusively for Mac.





Let's also create a Smart Group to look for enrolled Macs that are encrypted but do not have a valid Recovery Key stored in Jamf Pro:

Computers : Smart Computer Groups

← **New Smart Computer Group**

Computer Group Criteria

AND/OR	CRITERIA	OPERATOR	VALUE		
<input type="checkbox"/>	FileVault 2 Individual Key Validation	is <input type="checkbox"/>	Invalid	⋮	<input type="checkbox"/> Delete
<input type="button" value="+ Add"/>					

Now that we have insight into the FileVault 2 status of the fleet, we can start the encryption process. We build a policy that starts the process:

Computers : Policies

← **Enable FileVault Encryption**

Options Scope Self Service User Interaction Show in Jamf Pro Dashboard

General

Disk Encryption Individual Encryption >

Disk Encryption

Action Action to take on computers

Apply Disk Encryption Configuration

Disk Encryption Configuration Disk encryption configuration to use to enable FileVault 2

Individual Encryption

Require FileVault 2 Require users to enable FileVault 2 based on one of the following events

And we limit the scope of this policy to the Smart Group we've just created for Macs that aren't encrypted:

The screenshot shows the Jamf Pro interface for configuring the 'Enable FileVault Encryption' policy. The 'Scope' tab is selected, and the 'Targets' section is expanded. The 'Target Computers' dropdown is set to 'Specific Computers' and the 'Target Users' dropdown is set to 'Specific Users'. Below this, a table lists the target configuration:

TARGET	TYPE
FileVault - Eligible AND Not Encrypted	Smart Computer Group

Now, Macs that are not encrypted will funnel into the Smart Group and receive a policy that will get them all encrypted. And if you enroll a Mac next week that's brand new, Jamf Pro will automatically detect that FileVault encryption is not enabled, it will appear in the Smart Group and the policy can automatically enable FileVault.

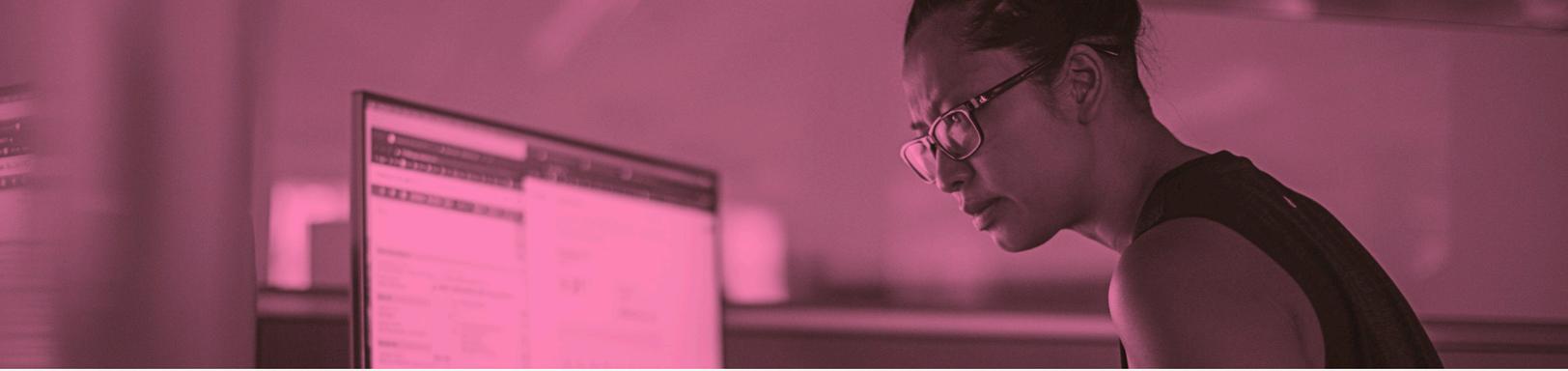
Extension attributes: adding in your own defaults

Jamf Pro collects more than 200 pieces of data about your Mac devices, but sometimes IT needs to collect information that's not on the list.

That's where extension attributes come in. For example, IT might need to periodically audit the date and time Jamf Protect last checked in to the Jamf Protect server. By default, Jamf Pro does not collect this information, but extension attributes allow IT to create a custom inventory field and run reports against that information.

IT admins can write their own extension attributes, but to save time, Jamf Pro contains templates for some of the most popular scenarios. To get started, navigate to Settings > Computer Management > Extension Attributes, and click "+ New From Template" to look at existing templates.

Within the Jamf Applications section of the extension attributes templates, select "Jamf Protect - Last Check-in." This is a script that queries the Jamf Protect command-line tool looking for the last check in and records that information within Jamf Pro..



The screenshot shows the Jamf Pro interface for configuring an extension attribute. The page is titled "Settings : Computer Management > Extension Attributes" and "Jamf Protect - Last Check-in".

- Display Name:** Jamf Protect - Last Check-in
- Enabled:** Enabled (script input type only)
- Description:** Displays the date and time of the last agent check-in for Jamf Protect
- Data Type:** Integer
- Inventory Display:** Extension Attributes
- Input Type:** Script

The script is shown in a terminal window with the following code:

```
1 #!/bin/sh
2
3 #Jamf Protect Location
4 jamfProtectBinaryLocation="/usr/local/bin/protectctl"
5
6 if [ -f "$jamfProtectBinaryLocation" ]; then
7   jamfProtectLastCheckin="$($jamfProtectBinaryLocation info | awk -F 'Last Check-in:' '{print $2}' | xargs)
8 else
9   jamfProtectLastCheckin="Protect binary not found"
10 fi
11
12 echo "<result>$jamfProtectLastCheckin</result>"
```

This extension attribute will now automatically report the last check-in date and store that information in the Mac computer's inventory record in Jamf Pro. From here, IT admins can run reports to get insight into when the Macs are checking in, and run remediation for Macs that haven't checked in recently.

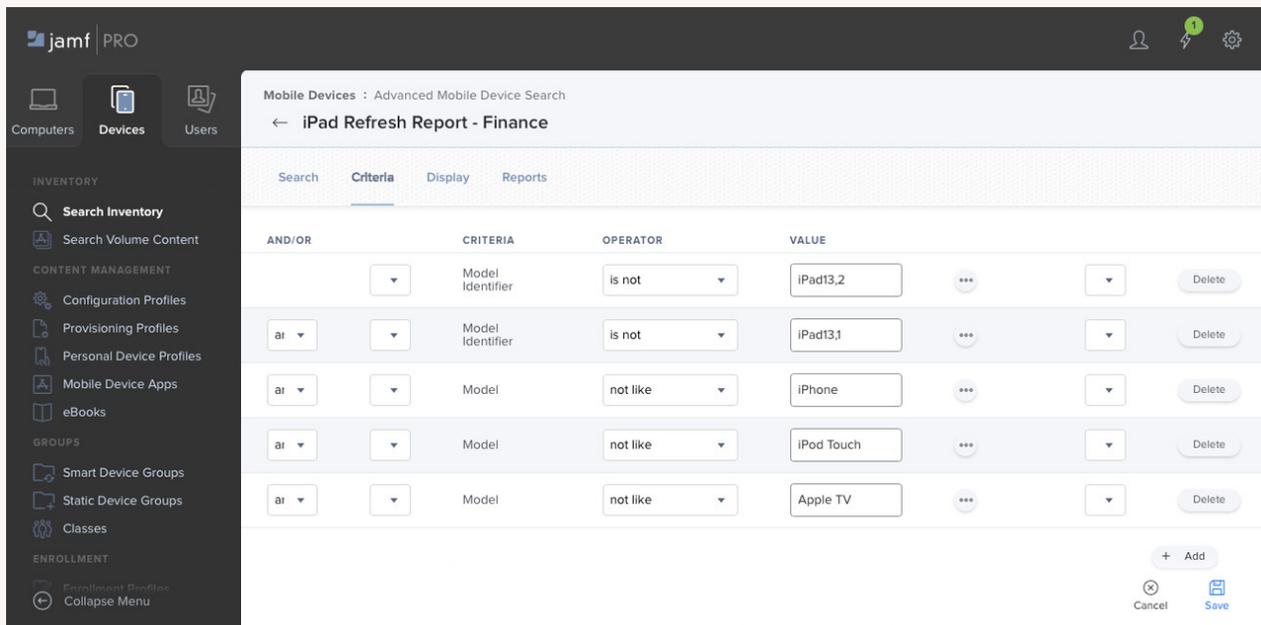
The screenshot shows the Jamf Pro interface for a specific computer, "test's MacBook Pro". The page is titled "Computers" and "test's MacBook Pro".

- Inventory:** General (test's MacBook Pro), Hardware (13-inch MacBook Pro (Early 2015)), Operating System (macOS 11.2.3), User and Location, Security, Purchasing, Storage (1 Drive), Extension Attributes (highlighted), Disk Encryption (Not Encrypted), Applications (57 Applications), Fonts (356 Fonts), Profiles (10 Profiles).
- Extension Attributes:** Jamf Protect - Last Check-in: 08:19:2021 4:12:45 PM GMT

Hardware Refresh: iPad refresh + Unified Model Deployment (UDM)

Say your organization has existing devices that are due to be refreshed. These devices are a few years old, and keeping track of inventory manually would mean working with your finance department to dig up purchase orders, putting that data into Excel and manually comparing. IT admins can use Jamf Pro to create a report of all eligible devices as well as to create a report for finance to have visibility on the devices that need to be refreshed.

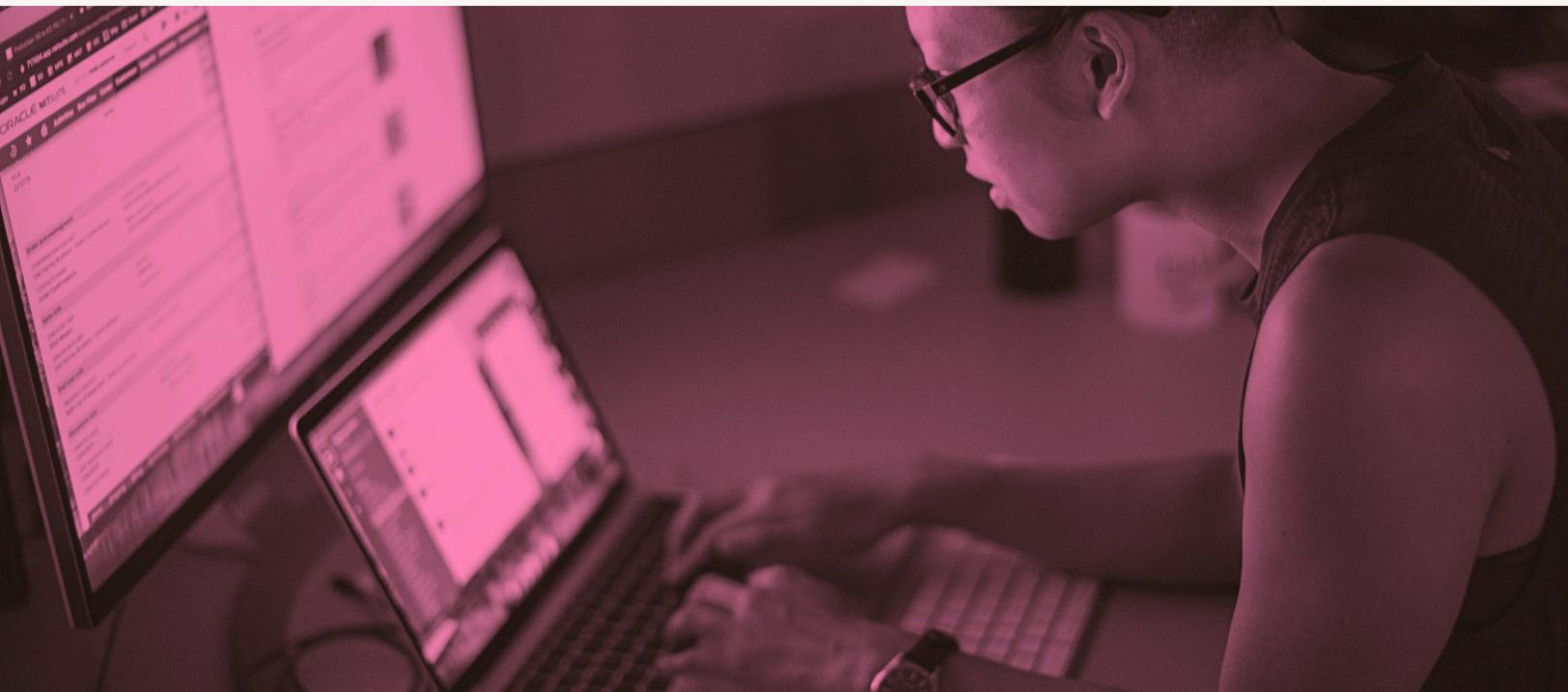
To start, we will create a saved advanced device search that looks for any iPads that are not the model your organization is moving to, along with other devices that will be replaced by the new iPads.



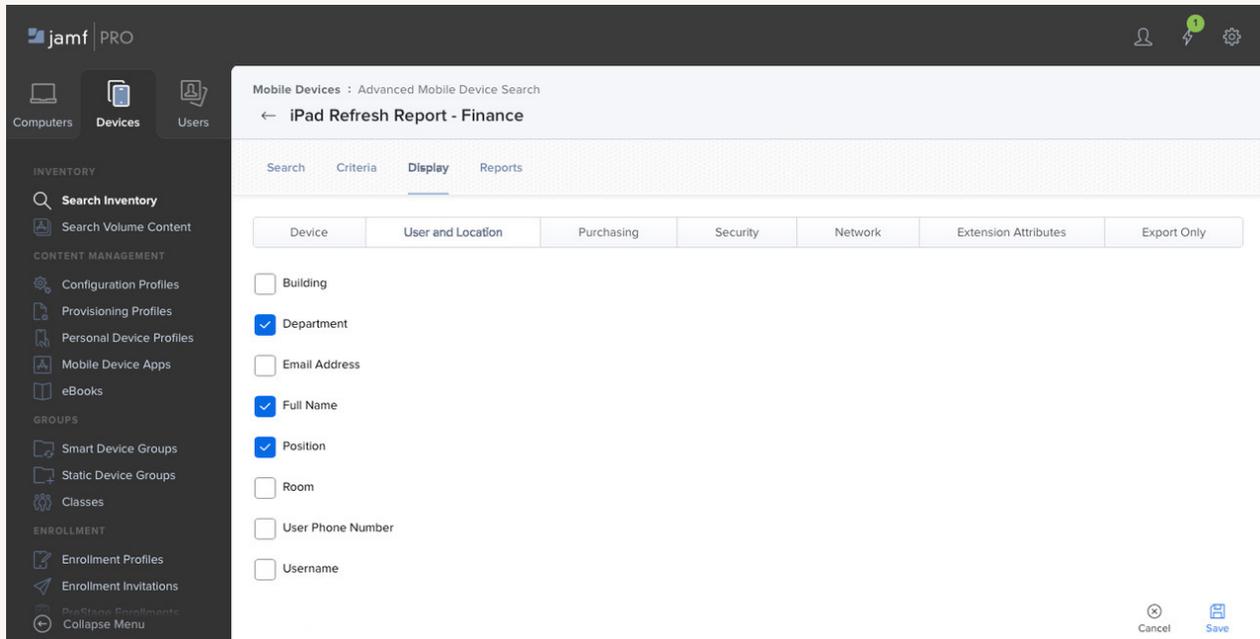
The screenshot shows the Jamf Pro interface with a sidebar on the left containing navigation options like 'Computers', 'Devices', and 'Users'. The main content area is titled 'Mobile Devices : Advanced Mobile Device Search' and 'iPad Refresh Report - Finance'. It features a search criteria table with columns for AND/OR, CRITERIA, OPERATOR, and VALUE. The table lists several criteria for identifying devices to be refreshed.

AND/OR	CRITERIA	OPERATOR	VALUE			
	Model Identifier	is not	iPad13,2	...	▼	Delete
and	Model Identifier	is not	iPad13,1	...	▼	Delete
and	Model	not like	iPhone	...	▼	Delete
and	Model	not like	iPod Touch	...	▼	Delete
and	Model	not like	Apple TV	...	▼	Delete

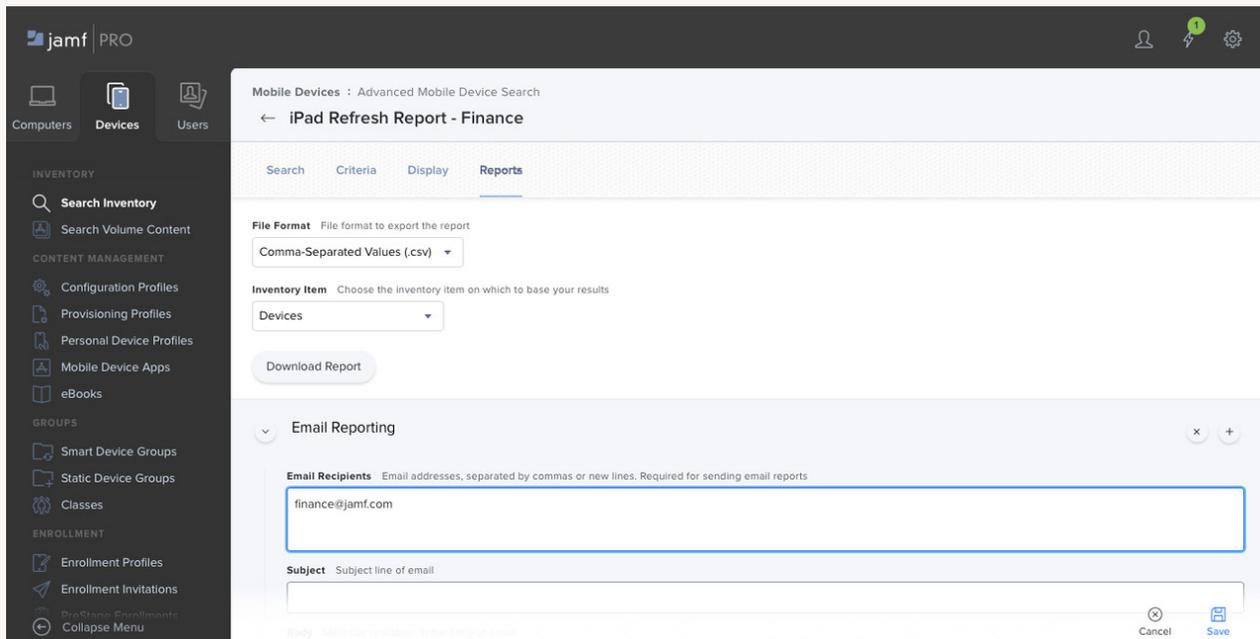
At the bottom right of the table, there are buttons for '+ Add', 'Cancel', and 'Save'.

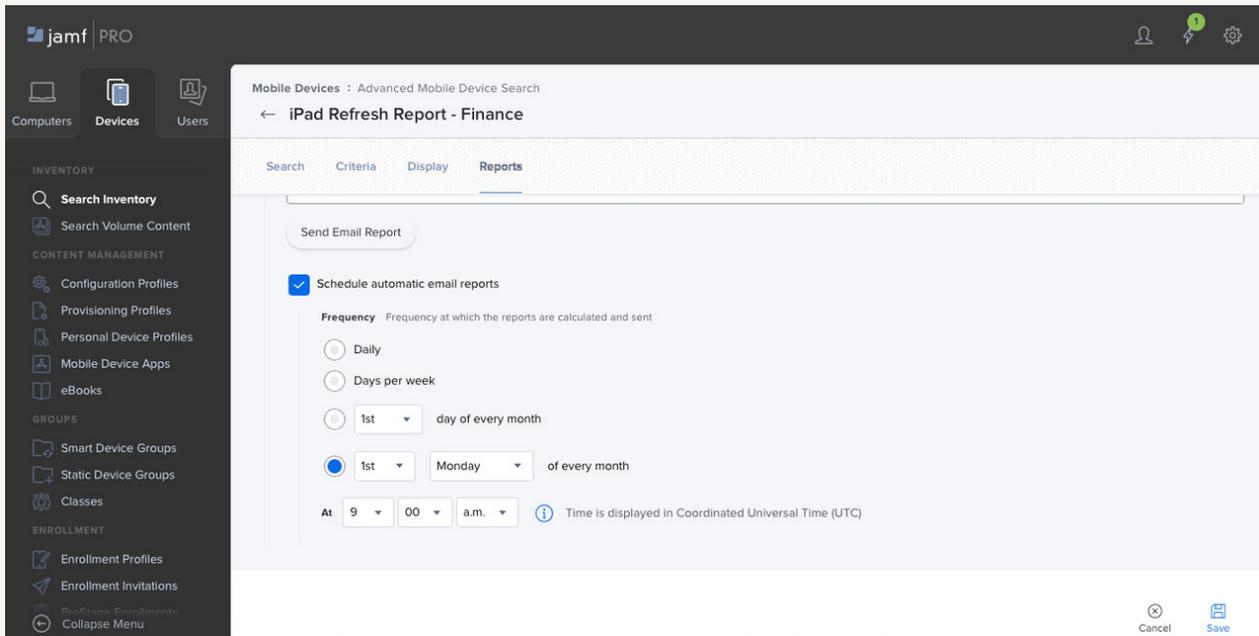


The report can be customized with information that finance will find helpful:



Here, the report is set to be delivered on the first of each month so that finance can space out the purchasing based on device age:





The report not only contains the list of devices but also the display information chosen when the saved advanced device search was created. Now IT and finance can coordinate the deployment of the new iPads as well as the retrieval of the old equipment.

58 Mobile Devices in iPad Refresh Report - Finance

Display Name	Model	Serial Number	Position	Full Name	Department
SalesDLX	iPad mini 4 (Wi-Fi)	DLXQ	Account Executive	Johnny Appleseed	Sales
SalesDLX	iPad Pro (12.9-inch Wi-Fi)	DLXS3	Account Executive	Johnny Appleseed	Sales
SalesDLX	iPad mini 4 (Wi-Fi)	DLXSJ	Systems Engineer	Johnny Appleseed	Sales
SalesDMP	iPad Pro (9.7-inch Wi-Fi)	DMPR	Systems Engineer	Johnny Appleseed	Sales
SalesGCV	iPad 5th Generation (Wi-Fi)	GCVV	Manager	Johnny Appleseed	Sales
DevDLXM	iPad Air (Wi-Fi)	DLXM	Developer I	Johnny Appleseed	Developer
DevDMPV	iPad Pro 2nd Generation (10.5-inch Wi-Fi)	DMPV	Developer I	Johnny Appleseed	Developer
DevF9FV9	iPad 5th Generation (Wi-Fi)	F9FV9	Developer II	Johnny Appleseed	Developer
DevGCTV	iPad 5th Generation (Wi-Fi)	GCTV	Developer III	Johnny Appleseed	Developer
DevF9FT6	iPad 5th Generation (Wi-Fi)	F9FT6	Senior Developer	Johnny Appleseed	Developer
DevDMPR	iPad Pro (9.7-inch Wi-Fi)	DMPR	Manager	Johnny Appleseed	Developer
DevDMQS	iPad Air 2 (Wi-Fi)	DMQS	Manager	Johnny Appleseed	Developer
FinDMPS	iPad Pro (9.7-inch Wi-Fi)	DMPS	Accountant	Johnny Appleseed	Finance
FinDMPR	iPad Pro (9.7-inch Wi-Fi)	DMPR	Controller	Johnny Appleseed	Finance
FinDLXSJ	iPad mini 4 (Wi-Fi)	DLXSJ	CFO	Johnny Appleseed	Finance/Executive
ExecDMP	iPad 6th Generation (Wi-Fi)	F9FW	CIO	Johnny Appleseed	Executive
ExecDMP	iPad Air 2 (Wi-Fi)	DMPR	CEO	Johnny Appleseed	Executive
ExecDLXS	iPad mini 3 (Wi-Fi)	DLXN	CTO	Johnny Appleseed	Executive

Using these steps, your organization is able to save valuable time and potential communication headaches by automating your information-gathering. IT is able to plan for the new devices and ensure they are finding and assisting the correct users while finance receives regular reports from IT for their purchasing needs.

Patch Management

Keeping software up-to-date across the fleet is a key part of endpoint security. Maintaining a wide array of software titles and updates can be difficult for IT admins. Applications can often come from multiple sources and be installed over a wide range of computers. If your organization has increasing security and operational requirements to ensure endpoints are secure, you'll need to be up-to-date with patches. Additionally, with a growing remote workforce, IT admins need a flexible solution to enable and empower end users.

Enter Jamf Pro's Inventory and Patch Management.

Jamf Pro automatically inventories installed applications— those included with the macOS as well as third-party applications installed by users or an MDM.

Category	Item Name	Version	Path	Status
Applications 57 Applications	Self Service.app	10.311	/Applications/Self Service.app	No
	Siri.app	1.0	/System/Applications/Siri.app	No
	Stickies.app	10.2	/System/Applications/Stickies.app	No
	Stocks.app	3.4	/System/Applications/Stocks.app	No
	System Information.app	11.0	/System/Applications/Utilities/System Information.app	No
	System Preferences.app	14.0	/System/Applications/System Preferences.app	No
	Terminal.app	2.11	/System/Applications/Utilities/Terminal.app	No
	TextEdit.app	1.16	/System/Applications/TextEdit.app	No
	Time Machine.app	1.3	/System/Applications/Time Machine.app	No
	TV.app	11.5	/System/Applications/TV.app	No
Package Receipts 30 Receipts	VoiceMemos.app	2.2	/System/Applications/VoiceMemos.app	No
	VoiceOver Utility.app	10	/System/Applications/Utilities/VoiceOver Utility.app	No
	zoom.us.app	5.4.7 (59780.1220)	/Applications/zoom.us.app	No

Say that you need to ensure Macs are running the latest version of Zoom. Jamf Pro's built-in patch management system can help IT make this happen.

The screenshot shows the Jamf Pro interface with a sidebar on the left containing navigation options like 'Computers', 'Devices', 'Users', 'Inventory', 'Content Management', 'Groups', 'Enrollment', and 'Settings'. The main panel is titled 'Computers - Patch Management' and displays a table of software titles. Each row includes a plus icon, the software name, publisher, version, and installation date.

Software Title	Publisher	Version	Installation Date
YubiKey Manager	Yubico	1.2.3	06/21/2021 at 04:48 AM
Zeplin	Zeplin	3.23.2	07/22/2021 at 01:07 PM
Zoom Client for Meetings	Zoom Video Communications	5.75 (1123)	08/12/2021 at 08:27 AM
Zoom Plugin for Microsoft Outlook	Zoom Video Communications	5.7.0	06/29/2021 at 02:23 AM
Zoom Rooms	Zoom Video Communications	5.75 (3865.0810)	08/13/2021 at 02:48 PM
zoom.us	Zoom Video Communications, Inc. (Legacy Definition)	5.75 (1123)	08/12/2021 at 08:28 AM
Zotero	Corporation for Digital Scholarship	5.0.96.2	05/24/2021 at 04:56 AM
Zulu OpenJDK 10	Azul	10.3.5	07/13/2021 at 06:43 AM
Zulu OpenJDK 11 (LTS)	Azul	11.50.19	07/22/2021 at 09:10 AM
Zulu OpenJDK 12	Azul	12.3.11	07/13/2021 at 06:43 AM
Zulu OpenJDK 13 (MTS)	Azul	13.42.17	07/22/2021 at 09:09 AM
Zulu OpenJDK 14	Azul	14.29.23	07/13/2021 at 09:16 AM
Zulu OpenJDK 15 (MTS)	Azul	15.34.17	07/22/2021 at 09:03 AM
Zulu OpenJDK 16 (JTS)	Azul	16.29.15	07/26/2021 at 08:45 AM

Let's choose Zoom Client for Meetings from the built-in list. Now we have a report showing which Macs have Zoom, and which version is installed on each.

The screenshot shows the Jamf Pro interface for the 'Zoom Client for Meetings' patch report. A circular progress indicator shows 0% for the latest version (5.7.5). Below it, a table lists the installed versions on two devices:

NAME	LAST CHECK-IN	INSTALLED VERSION
demo's MacBook Pro	2021/04/20 at 10:25 AM	5.2.2 (45106.0831)
test's MacBook Pro	less than a minute ago	5.2.2 (45106.0831)

Summary statistics:

- 0 Latest Version (5.7.5) (1123)
- 2 Other Version

Version Number: 5.2.2 (45106.0831) | Number of Devices: 2

Jamf Pro keeps a list of known version definitions for applications in the Patch Management library. In this case, we see all the recent versions of Zoom, as well as information about the app (OS requirements, etc).

The screenshot shows the 'New Patch Management Software Title' definition for zoom.us. The interface displays a list of 165 version definitions. The table below shows the first few entries:

VERSION	RELEASE DATE	INCREMENTAL UPDATE	REBOOT REQUIRED	DEPENDENCIES	MINIMUM OS	APPS THAT MUST QUIT
5.7.5 (1123)	08/12/2021 at 2:00 AM	Not Required	No		10.9	zoom.us
5.7.4 (898)	07/26/2021 at 11:00 AM	Not Required	No		10.9	zoom.us
5.7.3 (809)	07/19/2021 at 11:00 AM	Not Required	No		10.9	zoom.us
5.7.1 (499)	06/28/2021 at 8:00 AM	Not Required	No		10.9	zoom.us
5.7.0 (446)	06/21/2021 at 11:00 AM	Not Required	No		10.9	zoom.us
5.6.7 (1020)	06/07/2021 at 10:19 PM	Not Required	No		10.9	zoom.us
5.6.6 (950)	05/25/2021 at 8:00 AM	Not Required	No		10.9	zoom.us
5.6.4 (765)	04/26/2021 at 8:00 AM	Not Required	No		10.9	zoom.us
5.6.3 (706)	04/19/2021 at 8:00 AM	Not Required	No		10.9	zoom.us
5.6.1 (560)	03/29/2021 at 8:00 AM	Not Required	No		10.9	zoom.us
5.6.0 (536)	03/23/2021 at 12:00 PM	Not Required	No		10.9	zoom.us
5.5.5 (1348.0305)	03/08/2021 at 8:00 AM	Not Required	No		10.9	zoom.us
5.5.4 (13130.0228)	03/01/2021 at 1:10 PM	Not Required	No		10.9	zoom.us
5.5.2 (12513.0205)	02/05/2021 at 9:49 PM	Not Required	No		10.9	zoom.us
5.5.1 (12484.0202)	02/03/2021 at 3:14 PM	Not Required	No		10.9	zoom.us
5.5.0 (12467.0131)	02/01/2021 at 6:55 PM	Not Required	No		10.9	zoom.us

Since we identified Macs running the non-latest version of Zoom, we can create a patch policy to remediate this.

The screenshot shows the configuration page for a patch policy named "Update Zoom". The policy is enabled. The target version is set to 5.7.5 (1123). The release date is 08/12/2021 at 2:00 AM. The policy requires no incremental updates, no reboots, and has no dependencies. The apps that must quit are zoom.us, and the minimum OS is 10.9. The distribution method is "Make Available in Self Service". There are checkboxes for "Allow Downgrade" and "Patch Unknown Versions".

General | Scope | User Interaction

Display Name Display name for the patch policy
Update Zoom

Enabled

Target Version Version to deploy
5.7.5 (1123) *Only versions that have an associated package can be selected. You must use the Definition tab to add a package to a specific version.*

Release Date 08/12/2021 at 2:00 AM

Requires Incremental Update No

Reboot Required No

Dependencies N/A

Apps That Must Quit zoom.us

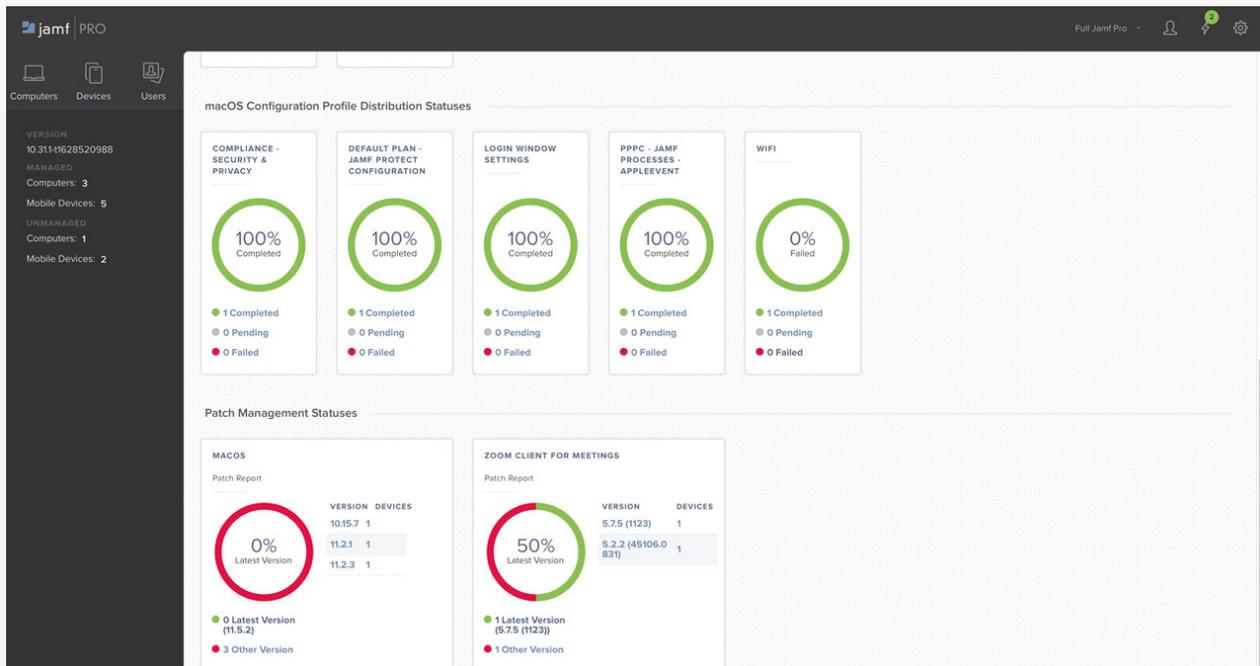
Minimum OS 10.9

Distribution Method Method to use for distributing the software title update
Make Available in Self Service *Requires Self Service v10.0.0 or later*

Allow Downgrade
Allow the software title to be downgraded

Patch Unknown Versions
If unknown versions (e.g. beta version, version not available in Jamf Pro) are detected on computers, deploy the target version

Now that we have our patch policy defined and properly scoped, the last step will be to audit the patch upgrading process. We can easily do this with the Jamf Pro dashboard.



Targeted Hardware Updates (aka Rosetta 2)

Let's say as a precursor for moving your fleet to M1-enabled Mac devices you need to know how many M1 and Intel Macs there are currently in your fleet. Jamf Pro inventory allows IT admins to build smart groups based on architecture type.



Jamf Pro automatically collects the architecture type of Macs enrolled and displays them in the individual computer inventory record as, for example,

Processor Type: Apple M1

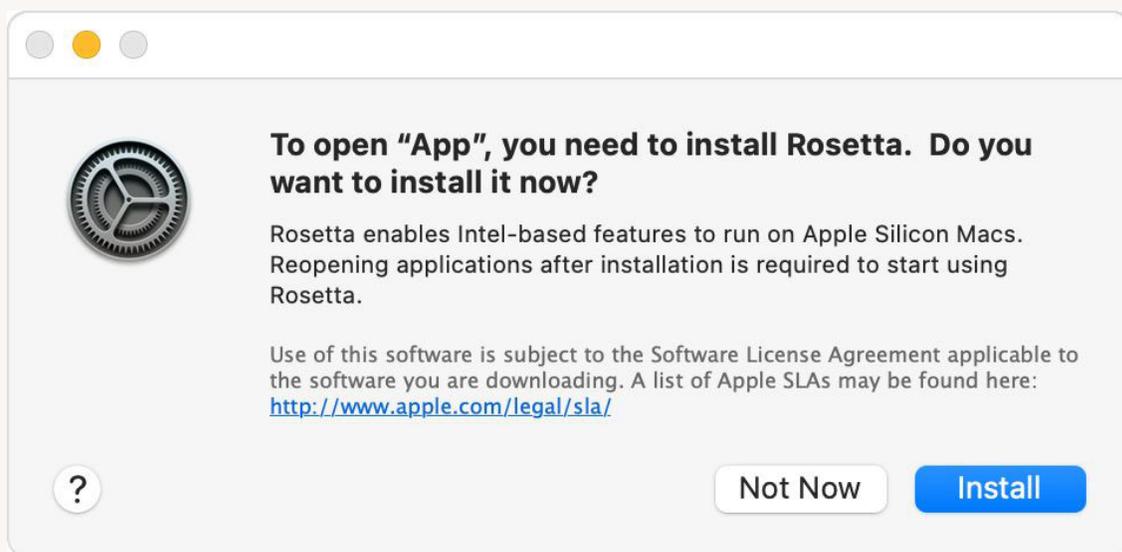
Architecture Type: arm64

You can automate reporting and deployments to Apple Silicon Macs using Smart Groups.

You can create a Smart Group targeting Apple Silicon in your organization's environment. Simply create a Smart Group titled 'Apple Silicon' that uses the advanced criteria or 'Architecture Type' and use the ellipses to choose arm64:

CRITERIA	OPERATOR	VALUE
Architecture Type	is	arm64

Now IT can install Rosetta 2 on these Macs. Rosetta 2 is a utility needed to run Intel applications on M1 and future Apple Silicon Macs. It's not installed by default.



This process is similar to the FileVault example above – build out a policy that installs Rosetta 2, and scope it to the Smart Group we just created. We can also use this same inventory information and Smart Group to make Apple Silicon-specific versions of applications available to end users in Self Service.

Automate tedious workflows with APIs and Webhooks

Now, what if you wanted to dynamically rename all your iOS devices using the same format, such as a combination of the end user's surname and department or building, or a teacher's surname and grade level? While Jamf Pro's interface allows you to change a device's name and even set device names en masse, it would be time-consuming to uniquely set each device name.

Here's where Jamf Pro API and webhooks come in. The combination of these two powerful tools built into Jamf Pro can automate this task for you.

First, webhooks can 'listen' for any time a new iOS device enrolls.

MobileDeviceEnrolled

This event is triggered when a mobile device is enrolled or re-enrolled into Jamf Pro. For more information on mobile device enrollment, see the [Administrator's Guide](#).

JSON

XML

```
{
  "event": {
    "bluetoothMacAddress": "string",
    "deviceName": "string",
    "icciID": "string",
    "imei": "string",
    "ipAddress": "string",
    "jssID": integer,
    "model": "string",
    "modelDisplay": "string",
    "osBuild": "string",
    "osVersion": "string",
    "product": "string",
    "room": "string",
    "serialNumber": "string",
    "udid": "string",
    "userDirectoryID": "string",
    "username": "string",
    "version": "string",
    "wifiMacAddress": "string"
  },
  "webhook": {
    "eventTimestamp": integer,
    "id": integer,
    "name": "string",
    "webhookEvent": "MobileDeviceEnrolled"
  }
}
```

Once the enrollment event occurs, the webhook payload is sent to a callback URL, which can then process the event and use the API to first get the currently-assigned username and building/department.

Finds a subset of data for a mobile device
Subset values can also be appended using an ampersand to return multiple subsets (e.g. /subsets/General&Location)

147 `https://api.jamfcloud.com/JSSResource/mobiledevices/id/id/subset/subset` Try It

PATH PARAMS

id `int32`
ID to filter by: 55

subset `string`
Subset to filter by: Location

cURL Swift Ruby Python

```
curl --request GET \  
--url \  
https://api.jamfcloud.com/JSSResource/mobiledevices/id/  
/55/subset/Location \  
--header 'Authorization: Basic ZGVtbzpz0cnlpdG91dA  
=='
```

200 OK Metadata Examples

```
<?xml version="1.0" encoding="UTF-8"?><mobile_device><  
location-<username>user49</username><realname>User 49<  
/realname><real_name>User 49</real_name><email_address  
>User49@email.com</email_address><position/><phones>612  
-605-6625</phones><phone_numbers>612-605-6625</phone_num  
ber><department><building>Minneapolis</building><room  
>301 S 4th Ave&#13;  
Suite 1075</room></location></mobile_device>
```

RESPONSE 200 B

OK

Once you find the necessary information, the API can send an 'update mobile device' command to set the device name automatically.

mobiledevicecommands

- 141 Finds all mobile device commands
- 142 Finds a mobile device command by UUID
- 143 Finds all mobile device commands by command name
- 144 Finds all mobile device commands for specified command
- 145 Creates a new mobile device command
- 146 Creates a new mobile device command

146 `https://api.jamfcloud.com/JSSResource/mobiledevicecommands/command`

BODY PARAMS

general `object`

command `string`
Command to send device

device_name `string`
Device name to set (Required for DeviceName command)

cURL Swift Ruby Python

```
curl --request POST \  
--url \  
https://api.jamfcloud.com/JSSResource/mobiledevicecomm  
ands/command \  
--header 'Content-Type: application/xml' \  
--data '{"general":{"always_enforce_last_mode":tr  
ue,"lost_mode_with_sound":false,"disallow_proximity_se  
tup":false}}'
```

201 Created

Created

Using these tools, we can fully automate the unique naming convention for each device whenever they enroll or re-enroll into Jamf Pro. [Learn more about Jamf Pro APIs and webhooks.](#)

You can't fix what you can't see

Jamf Pro gives you insight with accurate, up-to-date inventory information that is key to scaling your IT practice.

Admins can fix existing problems, anticipate fixes for new ones and patch out-of-date-software titles with Jamf Pro. With our patented Smart Groups, admins can create a "set it and forget it" management tool that automatically pushes the apps and settings as soon as new devices are enrolled.

Not a Jamf customer? See what's possible and how to apply this knowledge with a free trial.

[Request Trial](#)

Or contact your preferred Apple Reseller.