

# Apple管理者のためのスクリプト活用術(中級編)

一般的なタスクの自動化を目指す管理者に向けた中級ガイド

「Apple管理者のためのスクリプト活用術(入門編)」で学んだスクリプトの 基礎知識を次のレベルまで上げたいApple管理者の方は、ぜひこのガイド をご覧ください。

Appleデバイスの管理で頻繁に行う3つのタスクをスクリプトを使って自動化するための、シンプ ルで強力な3つのツールをご紹介します。



# ファイルの書き込みと読み出し

時として、後にアクセスが必要になる情報をMacに残しておく必要がある場合があります。例えば、Macがいつ準備および 導入され、誰によって構築されたのかを知ることができるプロビジョニングレシートを残すといったケースが考えられます。

そのための方法をいくつか見ていきましょう。

### ターミナルを使用する

入門編でご紹介した「echo」コマンドを覚えていますか? このコマンドを使うと、入力したものを表示することができます。例えば:

echo 'Hello, World!'

と入力してリターンキーを押すと、以下のように表示されます。

Hello, World.

#### echoコマンドでテキストファイルに保存する

「リダイレクト」の記号である >> を使うと、ファイルへの保存が可能です。ファイルが存在しない場合は新規作成され、すでに存在する場合はechoの内容をファイルの末尾に追加します(「>>」ではなく「>」を使った場合、ファイルの内容が上書きされます)。

echo Technician: Martin Moose >> ~/Desktop/provision.txt
echo "Date: \$( /bin/date +%y-%m-%d )" >> !\$
echo Department: Graphics >> !\$

リターンキーを押すと、デスクトップにファイルが現れます。

ファイルを選択してスペースバーを押すと、クイックルックで確認することができます。



# ファイルに現在の日付を追加する

以下のコマンドを使用すると、現在の日付を自動的にファイルに追加することができます。

```
echo "Date: $( /bin/date +%y-%m-%d )"
```

「%y-%m-%d」は、年、月、日を表します。

キー操作の手間を省くために、直前のコマンドの最後の引数を実行するショートカットを使って、テキストをファイルにリ ダイレクトします。

>> !\$

次にファイルに部署名を書き込みます。

```
echo Department: Graphics >> !$
```

この時点でクイックルックでファイルを確認すると、すべてのテキストが追加した順番通りに入っているのがわかるはずです。



もう一度読み出すには、catコマンドを入力し、続けてファイルのパスを指定します。

/bin/cat ~/Desktop/provision.txt

すると、ファイルの内容が直接ターミナルに表示されます。

Technician: Martin Moose Date: 20-06-13 Department: Graphics

## ここまでのまとめ

echo	ターミナルウィンドウに入力された内容を出力
>>	内容をファイルにリダイレクトする、または既存ファイルに追加
>	内容をファイルにリダイレクトする、または既存ファイルを上書き
\$( command )	コマンドを実行
!\$	最後の引数を実行
cat	ファイルを読み込む

### Defaultsコマンド

catコマンドを使うとファイル全体が読み込まれますが、その中の一部の情報だけを確認したい場合はどうすれば良いのでしょうか?

この場合、環境設定フォルダのplist (プロパティファイル)を読み込む際に使用される「defaults」コマンドを使います。

このコマンドは、plistを新規作成するために使うこともできます。

ファイルを新規作成するには、まず「defaults write」でファイルの場所を指定します。そして次に、このコンピュータを構築 した技術者を示すための「Technician」という単語と、続いて技術者の名前を入力します。

/usr/bin/defaults write ~/Desktop/provision.plist Technician 'Martin Moose'

次にリターンキーを押します。

デスクトップにファイルが表示されたら、クイックルックで確認してみてください。前回とはまったく異なっているはずです。

```
<?ml 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="1.0">
<dict>
<key>Technician</key>
<string>Martin Moose</string>
</dict>
</plist>
```

「defaults」コマンドでは、多くの書式が追加されますが、必要な情報はすべてそこにあるはずです。

次は日付と部署を追加して、もう一度ファイルを見てみましょう。

```
/usr/bin/defaults write ~/Desktop/provision.plist Technician 'Martin Moose'
/usr/bin/defaults write ~/Desktop/provision.plist Date $ ( /bin/date '+%y-%m-%d' )
/usr/bin/defaults write ~/Desktop/provision.plist Department 'Graphics'
```

これで、ファイルに3つの情報がすべて追加されました。

```
<?ml 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="1.0">
<dict>
<key>Date</key>
<string>20-06-13</string>
<key>Department</key>
<string>Graphics</string>
<key>Technician</key>
<string>Martin Moose</string>
</dict>
</dict>
```

plistの場合、それぞれの情報(例:技術者、日付、部署)は「key」として表示され、各keyの下にその値が表示されています (defaultsコマンドを使うと、どのような順番でkeyを追加しても自動的にアルファベット順に表示されます)。

plistは最初に作成したファイルよりもかなり複雑に見えますが、本領を発揮するのはここからです。

「defaults read」でファイルを読み込み、「Technician」、「Date」、または「Department」を指定して、1項目だけの値を返す ことができます。これは拡張属性などの作成に特に便利で、このコマンドを含む短いスクリプトを使って結果をJamf Pro にエコーバックすることが可能です。

例えば、以下のような文字列を入力すると、

/usr/bin/defaults read~/Desktop/provision.plist Date

ターミナルに日付だけが表示されます。

20-06-13



# Jamf Proでスクリプトパラメータを使用する

# スクリプトパラメータとは?

まずは簡単なスクリプトを例に取って見ていきましょう。

#!/bin/zsh

この場合のシバンは「zsh」です。

ここで全体が見やすいようにechoを使って空行を追加しておきます。

echo

次に、パラメータの変数 (1~5の数字) を含むechoステートメントを追加 します。

echo \$1 \$2 \$3 \$4 \$5

以前にも触れましたが、スクリプトに「\$」が含まれている場合、それは変数またはプレースホルダーが存在することを意味しています。

スクリプトの全体像は以下のようになります。

#!/bin/zsh
echo
echo \$1 \$2 \$3 \$4 \$5

このスクリプトに「parameters.zsh」と名前を付けてデスクトップに保存します。

スクリプトファイルを新規作成する場合は、必ずターミナルで「chmod」 というコマンドを使って実行可能な状態にする必要があります。これを 行わないと、ターミナルはそれを実行可能なスクリプトではなく、ただの テキストファイルとみなしてしまいます。 ····

BBEditのようなテキスト エディタはもちろんのこと、Macに 内蔵されているテキストエディット を使用して書くこともできます。スク リプトでは直線型の引用符を使用 するため、曲線型の引用符に自動 変換するオートコレクト機能はすべ てオフにしておきましょう(スクリプ トエディタの場合は自動的に行わ れます)。

スクリプトは、自分が好

きなコードエディタや

各スクリプトの先頭には 必ずシバン(#!)を入力し ます。この後に/bin/zsh (またはzshの代わりにbashやその 他のスクリプトタイプ)と入力するこ とで、インタープリタとしてシェルス クリプト(またはその他のスクリプト タイプ)を使うようターミナルに指 示することができます。

「モード変更」を意味するコマンドである「chmod」を入力し、次に「実行可能」を意味する「+x」を入力してから、スクリプ トをウィンドウ内にドラッグしてください。

このスクリプトを実行し「Mary had a little lamb」と追加すると、「Mary had a little lamb」がエコーバックされます。

chmod +x ~/Desktop/parameters.zsh Mary had a little lamb Mary had a little lamb これだけでは少し退屈なので、次はもっと楽しい例を用いて見ていきましょう。

スクリプトに戻り、もう1つechoコマンドを追加します。今回は変数の順番を入れ替えてみます。



これを保存して、ターミナルでスクリプトをもう一度実行し、「Mary had a little lamb」を追加してみてください。

~/Desktop/parameters.zsh Mary had a little lamb

するとこのようになります。



最初のものは正しく表示されていますが、2番目は語順が変わっています。これは、変数である\$1、\$2、\$3、\$4、\$5が、スクリプトの後に追加した各項目の順番を決めていることを表しています。

最初の単語である「Mary」は「\$1」に相当しています。

同じく、「had」は2番目の単語で「\$2」に相当します。

「Mary had a little lamb」に含まれる各単語は、「スクリプトパラメータ」と呼ばれ、スクリプトの後に入力された順番によって参照することがでいます。

スクリプト内の変数の順番を変えると、スクリプトを実行したときに語順が変わります。

000		parameters.zsh					
ф	parameters.zsh ×						
_	Users > mmoose > Desktop > 🔳 parameters.zs	h					
Q	1 #!/bin/zsh 2						
	3 echo						
ş	4 5 echo \$1 \$2 \$3 \$4 \$5						
Å	7 echo						
<u>_</u> 0	9 ech \$5 \$3 \$1 \$2 \$4						
	11 echo						
			mmoose — -zsh — 80×24				
		mmoose@MooseBookPro ~ % / amb	/Users/mmoose/Desktop/parameter	rs.zsh Mary H	nad a	little	1) =
		mmoose@MooseBookPro ~ % ( mmoose@MooseBookPro ~ % ) amb	chmod +x /Users/mmoose/Desktop/parameter /Users/mmoose/Desktop/parameter	/parameters.z rs.zsh Mary H	zsh nad a	little	1]
		Mary had a little lamb					
		mmoose@MooseBookPro ~ % / amb	/Users/mmoose/Desktop/parameter	rs.zsh Mary H	nad a	little	1)
		Mary had a little lamb					
	de.	lamb a Mary had little	<b>.</b>				
	40	mmoose@MoseBookPro %	<b>▶</b> \$4				
8	¢2 -	1 40					
503	-p-3	\$1 <b>*</b> *					
0.00							
<b>⊗</b> 0⊿	20						

さて、スクリプトパラメータがスクリプトに続く項目の順番を指すことがわかったところで、これをJamf Proで利用する方 法について見ていきましょう。



# 既存のスクリプトを探す

Jamf Nationでは、他のJamf Pro管理者が作成したスクリプトを参照することができます。jamfnation.comにア クセスし、[resources] → [Jamf Pro add-ons] → [Scripts]を選択します。

ここでは、様々なスクリプトを閲覧したり検索したりすることができます。

例えば、「タイムゾーン」のスクリプトパラメータを使う場合は、 以下の手順で行います。

- •スクリプトをダウンロード
- ファイルを開く
- •すべてのコンテンツを選択
- •コピー

Jamf Proインスタンスで以下を選択してください。

#### [設定] → [コンピュータ管理] → [スクリプト] → [新規]

- ・新規スクリプトに「Set time zone」と名前を付ける
- ・「スクリプト」タブをクリックし、スクリプトを貼り付ける

 「オプション」タブをクリックし、パラメータ4フィールドに「Time Zone」と入力(名前はどんなものでも構いませんが、理に適った名 前をつけることをお勧めします。この場合、スクリプトを実行して 4番目のパラメータである\$4に希望するタイムゾーンを設定します)
 保存する

- •「スクリプト」タブを再度クリック
- コピーしたいタイムゾーンをリストアップするコマンドを検索

なぜ、パラメータ1では なくパラメータ4を使う のでしょうか? パラメータラベルの上に小さく 「Parameters 1 through 3 are predefined as mount point, computer name and username (パラメータ1~3はマウントポイン ト、コンピュータ名、ユーザ名とし てあらかじめ定義されています)」 と表示されています。

Jamf Proでは、スクリプトがそれ を使用しているかどうかにかかわ らず、常に最初の3つのパラメー タとしてこれらの情報が送信され ます。

この3つのパラメータはユーザ側 で変更できませんが、それ以外に 最大8つのパラメータがあり、自由 に設定できます。

VERSION 10.22.0-t1591219900	General Script Options Limitations	
MANAGED Computers: 5	Script Contents	
Mobile Devices: 6	Default Mode 🔹 Default Theme 👻	-T T• ~ Commands
UNMANAGED Computers: 0	49 # 50 # SYNOPSIS	
Mobile Devices: 6	<pre>sudo setTimeZone.sh <mountpoint> <computername> <currentusername> <timezone> If the \$timeZone parameter is specified (parameter 4), this is the time sone that will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the \$timeZone parameter is specified for parameter 4, the hardcoded value in the script will be the script sets the system time zone as reflected in the Date &amp; Time preference pane with the script sets the system time zone entries can be found by running the command:     for Mac OS X 10 for left         /usr/sbin/systemsetup -listtimezones         for Mac OS X 10 for left         /usr/sbin/systemsetup -listtimezones         for Mac OS X 10 for left         //system/Library/CoreServices/RemoteManagement/ARDAgent.app/Contents/Support/system         The system time zone will be set according to the value specified in the parameter \$timeZone     } } </timezone></currentusername></computername></mountpoint></pre>	e set. used. th the the the msetup -listtimezones one.

# コマンドの実行方法

- ターミナルを開く
- ・権限を昇格して実行するためのコマンドである「sudo」を入力し、以下の文字列を貼り付ける

/usr/sbin/systemsetup -listtimezones

これにより、利用可能なタイムゾーンのリストが、スクリプトに適したフォーマットで表示されます。ここでは、アメリカの 中部標準時を代表する都市であるシカゴを選び、コピーします。

# ポリシーの作成

次はこのスクリプトを実行するためのポリシーを作成してみましょう。

- Jamfインスタンスの左側に表示されているメニューか「ポリシー」をクリックし、「カテゴリ」を選択(ここでは「Provisioning」を選びます)
- ポリシーに「Set Central Time Zone」という名前をつけて「Provisioning」カテゴリに追加

	⊜ jamfpro.talkingmoose.net	Ċ	0	80	+
<b>™ jamf</b>   PRO			Full Jamf Pro 🕤 🔒		
Computers : Policies Computers Devices Users · New Policy					
INVENTORY Options Scope S	elf Service User Interaction				
Search Volume Content	General				
CONTENT MANAGEMENT	Display Name Display name for the policy Set Central time zone				
Policies     Configuration Profiles     Software Updates     Not Configured	Enabled				
Restricted Software     Scripts     O Scripts	None •				
Mac App Store Apps     Printers     Printers	Category Category to add the policy to Provisioning				
GROUPS Disk Encryption	Trigger Event(s) to use to initiate the policy				
C Smart Computer Groups	User a user loss in to a computer A losin hose	checks for policies must be confi	igured in Jamf Pro for this to work	rie	
Classes Local Accounts ENROLLMENT	Upper Logout When a user logs out of a computer. A logout he	ook that checks for policies must	be configured in Jamf Pro for this to	work	
Enrollment Invitations     PreStage Enrollments	Network State Change When a computer's network state changes (e.g. when the IP address changes)	, when the network connection cf	hanges, when the computer name ch	anges,	
SETTINGS Directory Bindings OBindings	Enrollment Complete Immediately after a computer completes the enrollment	rollment process			

- ・後にスクリプト内でこのポリシーを呼び出せるように、カスタムトリガー(例:setcentraltimezone)を設定
- 「スクリプト」オプションを選択し、ターミナルコマンドからコピーしたタイムゾーンを貼り付ける

		≅ jamfpro.talkingmoose.net Č	• • • +
<b>™ jamf</b>   PRO		Full Jamf Pro	<u> </u>
Computers Devices Users	New Policies		
INVENTORY	lons Scope Self Service	User Interaction	
Q         Search Inventory           Image: Search Volume Content         Image: Search Volume Content	scripts > Script	Scripts	
CONTENT MANAGEMENT	Printers Printers	Set time zone	× +
Policies     Configuration Profiles	Disk Encryption lot Configured	Priority Priority to use for running the script in relation to other actions           After         *	
Restricted Software     PreStage Imaging	Dock Items Dock Items	Parameter Values Values for script parameters. Parameters 1-3 are predefined as mount point, comp Time Zone	uter name, and username
Mac App Store Apps     Z	ocal Accounts Accounts	America/Chicago	
GROUPS	Management Accounts lot Configured	Parameter 5	
Image: Smart Computer Groups         Image: Static Computer Groups	Directory Bindings Bindings	Parameter 6	
() Classes ENROLLMENT	EFI Password lot Configured	Parameter 7	
Enrollment Invitations     PreStage Enrollments	Restart Options lot Configured	Parameter 8	
SETTINGS 🔀 N	Maintenance lot Configured	Parameter 9	
Collapse Menu		Parameter 10	Cancel Save

このフィールドには、スクリプトを貼り付けたときに「オプション」タブで追加したラベルが表示されているはずです。 ラベルを見ることで、ここに何を入れるべきかがわかります。

あとは、ポリシーの適用範囲を決めて保存するだけです。

これで「Provisioning」カテゴリに新たなポリシーが追加されました。

スクリプトパラメータは非常に便利なツールです。

スクリプトパラメータを受け付けるスクリプトを書いておけば、異なるタイムゾーンのために何度でも再利用することができます。タイムゾーンごとに新しいポリシーを作成し、同じスクリプトを追加してタイムゾーンの値を記入するだけです。

# jamfHelperとosascriptでダイアログを作成する

ダイアログは、エンドユーザに対してメッセージを表示するだけでなく、情報をリクエストする際にも便利です。

ここでは、ダイアログを作成するための2つの方法と、それぞれメリットをご紹介します。

#### jamfHelper

jamfHelperはコマンドラインツールです。

ターミナルを開いて:

- ・ [ライブラリ] → [Application Support] → [JAMF] → [bin] → [jamfHelper]と進む
- ・ jamfHelperを右クリックして「パッケージの内容を表示する」を選択
- [Contents] → [MacOS]を選択すると、コマンドラインツールが表示されます。
- ターミナルにドラッグする

入力されたコマンドラインの末尾に「-help」と追加してリターンキーを押してください。

/Library/Application\ Support/JAMF/bin/jamfHelper.app/ Contents/MacOS/jamfHelper -help

これで、jamfHelperを使うために必要なすべての情報を見ることができるようになります。

	mmoose — -zsh — 81x40
mmoose@MooseBookPro ~ % /Libr tents/MacOS/jamfHelper -help	ary/Application\ Support/JAMF/bin/jamfHelper.app/Con <sup> m</sup>
JAMF Helper Help Page	
Usage: jamfHelper -windowType [-icon] [-button1] [-button2 s] [-alignDescription] [-alig [-iconSize] [-lockHUD] [-full	[-windowPostion] [-title] [-heading] [-description] ] [-defaultButton] [-cancelButton] [-showDelayOption nHeading] [-alignCountdown] [-timeout] [-countdown] ScreenIcon]
-windowType [hud   utility   hud: creates an Apple utility: creates an A fs: creates a full sc WARNING: Remo de	fs] "Heads Up Display" style window pple "Utility" style window reen window the restricts all user input te access must be used to unlock machines in this mo
-windowPosition [ul   ll   ur Positions window in t t of the user's screen If no input is given,	lr] he upper right, upper left, lower right or lower lef the window defaults to the center of the screen
-title "string" Sets the window's tit	le to the specified string
-heading "string" Sets the heading of t	he window to the specified string
-description "string" Sets the main content	s of the window to the specified string
-icon path Sets the windows imag	e filed to the image located at the specified path
-button1 "string" Creates a button with	the specified label
-button2 "string" Creates a second butt	on with the specified label

jamfHelperのパスは長いので、以下のスクリプトでシンプルなjamfHelperという変数に変換することをお勧めします。

#### #!/bin/zsh

jamfHelper="/Library/Application
Support/JAMF/bin/jamfHelper.app/Contents/MacOS/
jamfHelper"

あとは、変数の前にドル記号(\$)を付けて二重引用符で括るだけで、 jamfHelperを呼び出すことができます。 パス名を間違えないた めのヒントをご紹介し ます。jamfHelperを右 クリックして「Copy」にポインタを 置いた状態でOptionキーを押す と、パスをコピーすることができま す。今回のスクリプトの場合、引用 符の間にコピーしたパスを貼り付 けるだけでOKです。

"\$jamfHelper"

## jamfHelperでできること

jamfHelperにいくつかのオプションを追加したい場合を例にとってみましょう。 まずはウィンドウの種類を定義してみます(3つまで定義できます)。 ひとつめは「ヘッドアップディスプレイ」(HUD)です。 スクリプトに「-windowType」を追加し、ウィンドウの種類として「hud」を指定 します。

"\$jamfHelper" -windowType hud \

次に見出しを付けます。必ず引用符で囲んでください。

-heading "Preparing your computer..."  $\$ 

次に説明文を追加します。

-description "installing Microsoft Office 2019"  $\$ 

ダイアログボックスをもう少し魅力的に見せるには、アイコンを追加すると 良いでしょう。アイコンを選び、そのパスを追加します。パスに空白が含まれ ている場合もあるので、引用符で囲んでください。

-icon "/System/Library/CoreServices/Finder.app/ Contents/Resources/Finder.icns"





# Mac上にすでに存在するアイコンを見つける方法

- Finderで、[システム] → [ライブラリ] → [CoreServices]と進み、Finderを見つけます。
- Finderを右クリックして「パッケージの内容を表示」を選択し、次に[Contents] → [Resources]を選択します。
- Finderアイコンを選択します。
- Finderアイコンを右クリックし、Optionキーを押しながらパスをコピーします。
- そのパスをスクリプトに貼り付けます。

<b>Finder</b> File Edit View Go Window Help			Q iii
●●●	zsh 81×40	a company of the second	
<pre>mmoose@MooseBookPro ~ % /Library/Applica tents/MacOS/jamfHelper -help</pre>	tion\ Support/JAMF/bin/jamfHelper.ap	p/Con	Macintosh H
JAMF Helper Help Page		× ⊕ Q Search	
Usage: jamfHelper -windowType [-windowPc [-icon] [-button1] [-button2] [-default s] [-alignDescription] [-alignHeading] [ [-iconSize] [-lockHUD] [-fullScreenIcon] -windowType [hud   utility   fs] hud: creates an Apple "Heads Up utility: creates an Apple "Utili fs: creates a full screen window WARNING: Remote access m de -windowPosition [ul   ll   ur   lr] Positions window in the upper ri t of the user's screen If no input is given, the window -title "string" Sets the window's title to the s	Vacob       Percentes     CodeSignature       Applications     Applications       Desktop     Pugns       Downloads     Papores       Ownloads     Person plist       Ownloads     Person plist       Orange     Yellow       Orange     Yellow       Due     Use	Alternate smart plist Iternate smart plist Iternate smart plist Sase Sorol Sase Sorol Sase Sorol Salproj Salproj Salproj Salproj Salproj Finder.icns Anota From Sana & dit Kit Iternation Sana Sorol Sana Sorol	
-neading "string" Sets the heading of the window t -description "string" Sets the main contents of the wi	b the specified	Compress "Finder.Icns" Burn "Finder.Icns" to Disc Make Allas Quick Look "Finder.Icns" Share Quick Actions	
-icon path Sets the windows image filed to	the image locate	Copy "Finder.icns"	
-button1 "string" Creates a button with the specif	ied label	Use Groups Sort By Show View Options Show Preview Options	
Creates a second button with the	specified labe ⊙o∆o	Col 8 Spaces	4 UTF-8 LF Plain Text 🔎 🗘
Spalle graph	Alexandre and	Set Desktop Picture	

上記の内容をまとめるとこのような形になります。

```
#!/bin/zsh
jamfHelper="/Library/Application
Support/JAMF/bin/jamfHelper.app/Contents/MacOS/jamfHelper"
"$jamfHelper" -windowType hud \
-heading "Preparing your computer..." \
-description "installing Microsoft Office 2019"\
-icon "/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns"
```

このスクリプトをスクリプトエディタから直接実行してみてください。 とてもシンプルなダイアログが表示されるはずです。



次はウィンドウの種類を「hud」から「fs」(フルスクリーンの意味)に変更し、もう一度スクリプトを実行してみてください。 同じダイアログボックスが全画面表示されるはずです。



このようにダイアログを全画面表示させることで、プロビジョニング中にユーザがコンピュータを使用するのを防ぐことができます。また、ソフトウェアのインストールなどの作業が進んでいる間に、ダイアログをアップデートして進捗状況をユ ーザに知らせることもできます。

ダイアログボックスの後ろで起こっていることを確認したい場合は、Command + qを押すことで終了できます。

## Osascriptコマンド

ターミナルからosascriptコマンドを使い、AppleScriptでダイアログを作成することもできます。

この方法を使うことで、jamfHelperにできないことができるようになります。

まず、3つの部署名から選択することを可能にするダイアログのコマンドを作成しましょう。

コマンドは二重引用符で囲む必要があり、コマンドそのものに二重引用符が含まれている場合は、その直前にバックスラ ッシュを付けてそれがコマンドの一部であることを示さなければなりません。

asCommand="choose from list {\"Accounting\", \"Sales\", \"kiosk\"}

次に、Macがこれから行うことをエンドユーザーに向けて説明するメッセージを追加します。

with prompt \"Hello, let's prepare your Mac. To start, choose your department below...\"

コマンドの最後に、ダイアログウィンドウのタイトルを追加します。コマンドの最後に二重引用符を付けることをお忘れなく。

with title \"Prepare your Mac\""

この文字列がコマンドの実行をosascriptに指示し、その結果を部署を示す変数 (department) に返します。

department=\$( /usr/bin/osascript -e "\$asCommand" )

最後に、ユーザの選択を示すためにdepartmentの変数をエコーします。

echo "\$department"

スクリプトの全体像は以下のようになります。

```
#!/bin/zsh
asCommand="choose from list {\"Accounting\", \"Sales\", \"kiosk\"} with prompt \
"Hello, let's prepare your Mac. To start, choose your department below...\"
with title \"Prepare your Mac\""
department=$( /usr/bin/osascript -e "$asCommand" )
echo "$department"
```

では、このスクリプトをスクリプトエディタで実行してみましょう。ユーザが部署に「Accounting」を選択した場合、以下のように表示されます。

	Prepare your Mac
Hello, let's your depart	prepare your Mac. To start, choose tment below
Accountin	g
kiosk	
	Cancel

AppleScriptやosascriptを使ったダイアログでできることは数多くありますが、まずはこのような簡単なものから始めると良いでしょう。

# ここまでのまとめ

以下は、新しい社員に部署を選択させてMacのプロビジョニングを行うための方法として、Self Serviceに追加できるスクリプトの一例です。

まず、ログインしているユーザのフルネームとユーザ名を取得し、次に部署を選択させるためのosascriptのプロンプトで そのフルネームを使います。

jamfHelper="/Library/Application Support/JAMF/bin/jamfHelper.app/Contents/MacOS/ jamfHelper"

currentUser=\$( /usr/bin/stat -f "%Su" /dev/console )

fullName=\$( /usr/bin/id -F "\$currentUser" ) # e.g. "mmoose"

echo "Provisioning user is \$fullName (\$currentUser)"

次にユーザに部署の選択を求めます。

```
asCommand="choose from list {\"Accounting\", \"Sales\", \"Kiosk\"} with prompt 
\"Hello, $fullName! Let's prepare your Mac. To start, choose your department 
below...\" with title \"Prepare your Mac\""
```

department=\$( /usr/bin/osascript -e "\$asCommand" )

echo "Provisioned for department \$department"

さらに、別のosascriptでコンピュータのアセットタグをリクエストします。

asCommand="text returned of (display dialog \"Enter this Mac's asset tag (see bottom of computer)...\" default answer \"\" with title \"Prepare your Mac\")"

assetTag=\$( /usr/bin/osascript -e "\$asCommand" )

echo "Device asset tag is \$assetTag"

その後、スクリプトがすべてのコンピュータに対してタイムゾーンを設定します。

/usr/local/bin/jamf policy -event settimezonechicago

さらに、選ばれた部署に応じて必要なソフトウェアをインストールします。

if [[ "\$department" = "Accounting" ]]; then echo "Provisioning this Mac for Accounting" /usr/local/bin/jamf policy -event installchrome /usr/local/bin/jamf policy -event installoffice elif [[ "\$department" = "Sales" ]]; then echo "Provisioning this Mac for Sales" /usr/local/bin/jamf policy -event installchrome /usr/local/bin/jamf policy -event installoffice /usr/local/bin/jamf policy -event installoffice /usr/local/bin/jamf policy -event installzoom else echo "Provisioning this Mac as a kiosk" /usr/local/bin/jamf policy -event installchrome /usr/l ソフトウェアがインストールされたら、アセットタグ、部署、ユーザ名の情報を追加するためにインベントリアップデートが 実行されます。

```
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Updating inventory" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/usr/local/bin/jamf recon -assetTag "$assetTag" -department "$department"
-endUsername "$currentUser"
```

次に、管理ツールや情報を格納するためのフォルダがライブラリに作成されます。

```
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Creating admin folder" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
```

/bin/mkdir -p "/Library/Talking Moose Industries"

さらに、プロビジョニングレシートが残されます。

```
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Writing provisioning receipt" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
provisiondate -date $( /bin/date "+%Y-%m-%d" )
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
provisioner -string "$currentUser"
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
department -string "$department"
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
department -string "$department"
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
```

最後に、ユーザに通知してからコンピュータの再起動を行います。

```
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Restarting your Mac in one minute" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/sbin/shutdown -r +1 &
exit 0
```

以下は、ディスクリプタが埋め込まれた状態のスクリプトの全体像です。

```
#!/bin/zsh
# assign long path to JamfHelper to a shorter variable
jamfHelper="/Library/Application
Support/JAMF/bin/jamfHelper.app/Contents/MacOS/jamfHelper"
# get information about the current user
currentUser=$( /usr/bin/stat -f "%Su" /dev/console )
fullName=$( /usr/bin/id -F "$currentUser" ) # e.g. "Martin Moose"
echo "Provisioning user is $fullName ($currentUser)"
# AppleScript command to ask the currently logged in user to choose a department
asCommand="choose from list {\"Accounting\", \"Sales\", \"Kiosk\"} with prompt
\"Hello, $fullName! Let's prepare your Mac. To start, choose your department
below...\" with title \"Prepare your Mac\""
# run the command
department=$( /usr/bin/osascript -e "$asCommand" )
echo "Provisioned for department $department"
# AppleScript command to ask the currently logged in user to enter an asset tag
asCommand="text returned of (display dialog \"Enter this Mac's asset tag (see
bottom of computer)... \" default answer \"\" with title \"Prepare your Mac\")"
# run the command
assetTag=$( /usr/bin/osascript -e "$asCommand" )
echo "Device asset tag is $assetTag"
```

```
# build this Mac for the selected department
# global settings
/usr/local/bin/jamf policy -event settimezonechicago
if [[ "$department" = "Accounting" ]]; then
         echo "Provisioning this Mac for Accounting"
         /usr/local/bin/jamf policy -event installchrome
         /usr/local/bin/jamf policy -event installoffice
elif [[ "$department" = "Sales" ]]; then
         echo "Provisioning this Mac for Sales"
         /usr/local/bin/jamf policy -event installchrome
         /usr/local/bin/jamf policy -event installoffice
         /usr/local/bin/jamf policy -event installzoom
else
         echo "Provisioning this Mac as a kiosk"
         /usr/local/bin/jamf policy -event installchrome
         /usr/local/bin/jamf policy -event installzoom
fi
# update Jamf Pro inventory
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Updating inventory" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/usr/local/bin/jamf recon -assetTag "$assetTag" -department "$department" -
endUsername "$currentUser"
# create admin folder
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Creating admin folder" \setminus
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/bin/mkdir -p "/Library/Talking Moose Industries"
```

```
# create provisioning receipt
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Writing provisioning receipt" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
provisiondate -date $( /bin/date "+%Y-%m-%d" )
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
provisioner -string "$currentUser"
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
department -string "$department"
/usr/bin/defaults write "/Library/Talking Moose Industries/receipt.plist"
assettag -string "$assetTag"
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Restarting your Mac in one minute" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
# restart the Mac
/sbin/shutdown -r +1 &
exit 0
```

それでは、これが実際にどのように表示されるのかを見てみましょう。

最初に、Self Serviceを開いて「Provision」ボタンをクリックします。



ステップ1



Hello, Mart	in Moosel Let's prepare your Mac. To	
start, choo	se your department below	
Accountin		
Sales		
Kiosk		
	Cancel	
=,,,, <i>−</i> ,	)	
テップ	2	
テップ	2 Prepare your Mac	
テップ	2 Prepare your Mac	
テップ: inter this N	2 Prepare your Mac lac's asset tag (see bottom of computer)	-
テップ: inter this N	2 Prepare your Mac ac's asset tag (see bottom of computer)	1
テップ: inter this N	2 Prepare your Mac ac's asset tag (see bottom of computer)	
テップ: inter this N	Prepare your Mac ac's asset tag (see bottom of computer)	]



ステップ4



ステップ6



ステップ7

プロビジョニングが終わるとコンピュータが再起動し、再度ログインウィンドウが表示されます。

ユーザがログインすると、必要なソフトウェアがインストールされ、ライブラリフォルダにはプロビジョニングレシート が入っているはずです。

Jamf Proでコンピュータの記録を確認すると、osascriptダイアログ経由でアセットタグやユーザ名が入力されています。これにより、管理者はLDAP検索を行い、メールアドレス、電話番号、UserPrincipalNameなどの追加情報をActive Directoryから取得することができます。

🗖 jamf 🛛 PRO		Full Jamf Pro - <u>Q</u> 🗳 🚳
Computers Devices Users	Computers ← MooseBookPro	
Search Inventory		мл <b>у</b>
Search Volume Content	General >	MDM Capability: Yes
Licensed Software	- Hardware	Enrolled via Automated Device Enrollment: Yes
CONTENT MANAGEMENT	MacBook Air (11-inch Early 2015)	User Approved MDM: Yes
Policies	Operating System	MDM Capable Users: mmoose
Configuration Profiles     Restricted Software	Mac US A 10.15.5	Jamf Pro Computer ID: 9
PreStage Imaging	User and Location mmoose	Asset Tag: 235454
Mac App Store Apps	0	Per Code 1
Patch Management	Security	bai cove i.
eBooks	. Purchasing	Bar Code 2:
GROUPS	~	Bluetooth Low Energy Capability: Not Capable
Static Computer Groups	Storage 1 Drive	Logged in to iTunes Store: Not Active
Classes	Extension Attributes	Extension Attributes
ENROLLMENT		adobe-flash: Not Installed
Enrollment Invitations	Disk Encryption	

User and Location	Edit
Username: mmoose	
Full Name: Martin Moose	
Email Address: martin.moose@talkingmo	pose.net
Phone Number: 212-555-1313	
Position: Reception	
Department:	
Building:	
Room:	

これですべて完了です!

# ためになる学習リソース

## Jamfのトレーニングカタログ

今後スクリプトを使用していきたいと考えるJamfカスタマーは、ぜひJamfの トレーニングカタログをご活用ください。ヘルプデスクから管理者、エンジ ニアまで、すべての人にご利用いただける数々の短いハウツー動画(合計15 時間以上)が揃っています:trainingcatalog.jamf.com

15個のモジュールとビデオレッスンから構成されるスクリプトシリーズをぜ ひご覧ください。各レッスンは30分以内で完了できるので、気軽にスクリプ トの学習を続けることができます。

### Github

<u>GitHub.com</u>のオープンソースコミュニティを活用しましょう。無料で使用で きるオープンソースのソフトウェアが揃っているほか、JamfのGitHubページ (<u>github.com/jamf</u>)では、何百種類ものスクリプトやプロジェクトをご利用 いただけます。

# このホワイトペーパーの著者が提供しているリソース

多様なニーズに対応した様々な長さのスクリプトに興味のある方は、この ホワイトペーパーの著者であるBill Smithの<u>Gistリポジトリ</u>をご覧ください。 このホワイトペーパーでサンプルとして取り上げたプロビジョニング用のス クリプトや、その他のコードも掲載されています。<u>https://gist.github.com/</u> talkingmoose

このガイドが、あなたのスクリプトを次なるレベルに引き上げるための助けとなることを願っています。

スクリプトと優れたデバイス管理システムを組み合わせて、本来なら何時間 もかかる作業を数クリックで済ませることに興味のある方は、この機会にぜ ひJamf Proをお試しください。

無料トライアルに申し込む

この機会にぜひJamf Proをお試しください。