Scripts de Apple con Jamf 201:

Una guía intermedia para automatizar tareas comunes

Si usted es un administrador de Apple que desea llevar las habilidades básicas de scripting que aprendió en Scripting 101 al siguiente nivel, ¡esta guía es para usted!

Puede automatizar tres tareas comunes de IT para la administración de dispositivos Apple con tres herramientas simples y poderosas para la creación de scripts:

Escribir texto en un archivo y recuperarlo para su uso posterior

Usar los parámetros de los scripts en Jamf Pro

Crear diálogos interactivos usando jamfHelper y osascript

Escribir y leer un archivo

A veces, necesitas dejar información en una Mac para poder acceder a ella más tarde. Por ejemplo, es posible que desee dejar un recibo de aprovisionamiento que le permita saber cuándo se preparó e implementó una Mac y quién la construyó.

Aquí tiene algunas formas de hacerlo:

A través de la Terminal

¿Recuerda el comando "echo" de nuestro webinar Scripting 101?
Este comando hace eco, o imprime, lo que introduzca. Si introduce:

```
echo '¡Hola, mundo!'
```

...y pulsa retorno, sale:

```
¡Hola, mundo!
```

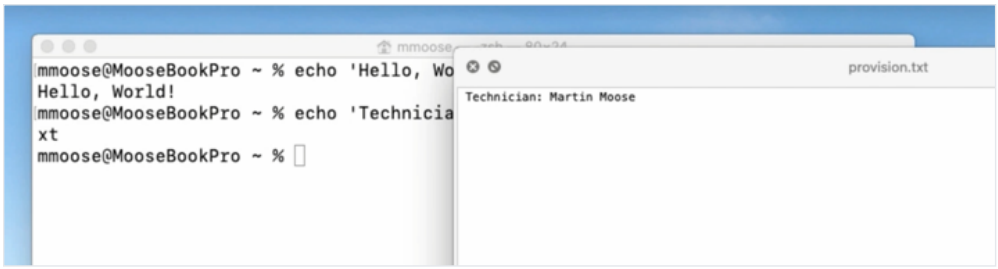
Guardar en un archivo de texto con el comando echo

Utilizando el signo >>, que se llama símbolo de "redirección", puede guardar en un archivo. Si el archivo no existe, el símbolo de redirección lo creará. Si ya existe, añadirá al final del archivo lo que haya hecho eco. (Si utiliza una sola redirección, >, se sobrescribe el contenido del archivo).

```
echo Técnico: Martin Moose >> ~/Desktop/provision.txt
echo "Fecha: $( /bin/date +%y-%m-%d )">>!$
echo Departamento: Gráficos >> !$
```

Cuando pulsa la tecla de retorno, el archivo aparecerá en su Escritorio.

Puede seleccionar el archivo y tocar la barra espaciadora para verlo en QuickLook.



Agregar la fecha actual a un archivo

Para automatizar la adición de la fecha actual a un archivo, utilice el comando:

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

Los términos "%y-%m-%d" significan año, mes y fecha.

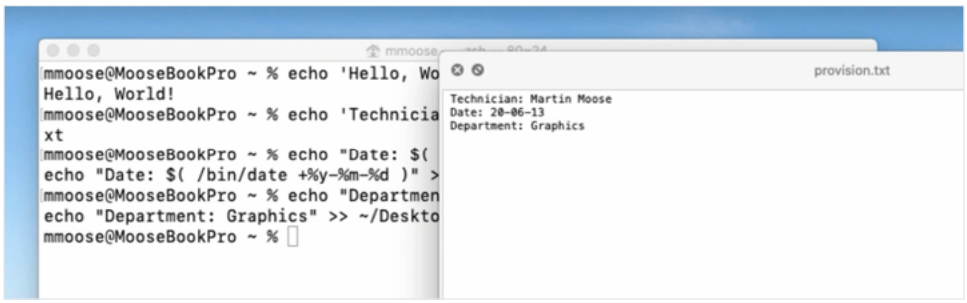
Para ahorrarse un par de teclas, utilice un atajo que repita el último argumento del comando anterior para redirigir el texto al archivo:

```
>> !$
```

Y luego escriba un nombre de departamento en el archivo:

```
echo Departamento: Gráficos >> !$
```

Cuando ahora aplicamos QuickLook al archivo, todo nuestro texto está ahí en el orden en que lo añadimos:



Para volver a leerlo, use el comando "cat" seguido de la ruta del archivo:

```
/bin/cat ~/Desktop/provision.txt
```

Entonces mostrará el contenido de ese archivo directamente en el Terminal:

```
Técnico: Martin Moose
Fecha: 20-06-13
Departamento: Gráficos
```

Revisión

`echo`	Imprime lo que introduzca en la ventana de Terminal
`>>`	Redirige la salida a un archivo/añade un archivo existente
`>`	Redirige la salida a un archivo/sobreescribe el archivo existente
`$( command )`	Ejecuta un comando
`!$`	Repite el último argumento
`cat`	Lee un archivo

Comando por defecto

El comando "cat" le permite leer todo el archivo, pero ¿qué pasa si solo quiere parte de su información?

El comando "valores predeterminados" es el mismo que utilizaría para leer los plists de su carpeta de preferencias, y también puede utilizarse para leer información de vuelta.

También podemos utilizarlo para escribir nuestros propios plists.

Para iniciar un nuevo archivo, utilice "escritura por defecto" y dígale dónde quiere el archivo. A continuación, proporcione una descripción de una sola palabra, "Técnico", para indicar el técnico que construyó esta computadora, y siga con el nombre del técnico.

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

Oprima retorno.

El archivo aparece en el escritorio, y cuando utilice QuickLook para verlo, será muy diferente del primero:

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

El comando "por defecto" añade mucho formato, pero si se fijas bien, toda la información está ahí.

Agreguemos la fecha y el departamento y miremos de nuevo el archivo:

```
/usr/bin/defaults write ~/Desktop/provision.plist Técnico 'Martin Moose'
/usr/bin/defaults write ~/Desktop/provision.plist Fecha $ ( /bin/date '+%y-%m-%d' )
/usr/bin/defaults write ~/Desktop/provision.plist Departamento 'Gráficos'
```

Ahora, el archivo muestra los tres datos:

```
<?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="1.0">
<dict>
    <key>Fecha</key>
    <string>20-06-13</string>
    <key>Departamento</key>
    <string>Gráficos</string>
    <key>Técnico</key>
    <string>Martin Moose</string>
</dict>
</plist>
```

Como se trata de un plist, cada tipo de información —Técnico, Fecha y Departamento— aparece como una "clave" y el valor de ese tipo de información aparece debajo de cada clave. (El comando por defecto alfabetiza automáticamente las claves. No importa el orden en que los añada, aparecerán en orden alfabético).

El plist parece mucho más complejo que nuestro primer archivo de texto, pero aquí es donde brilla:

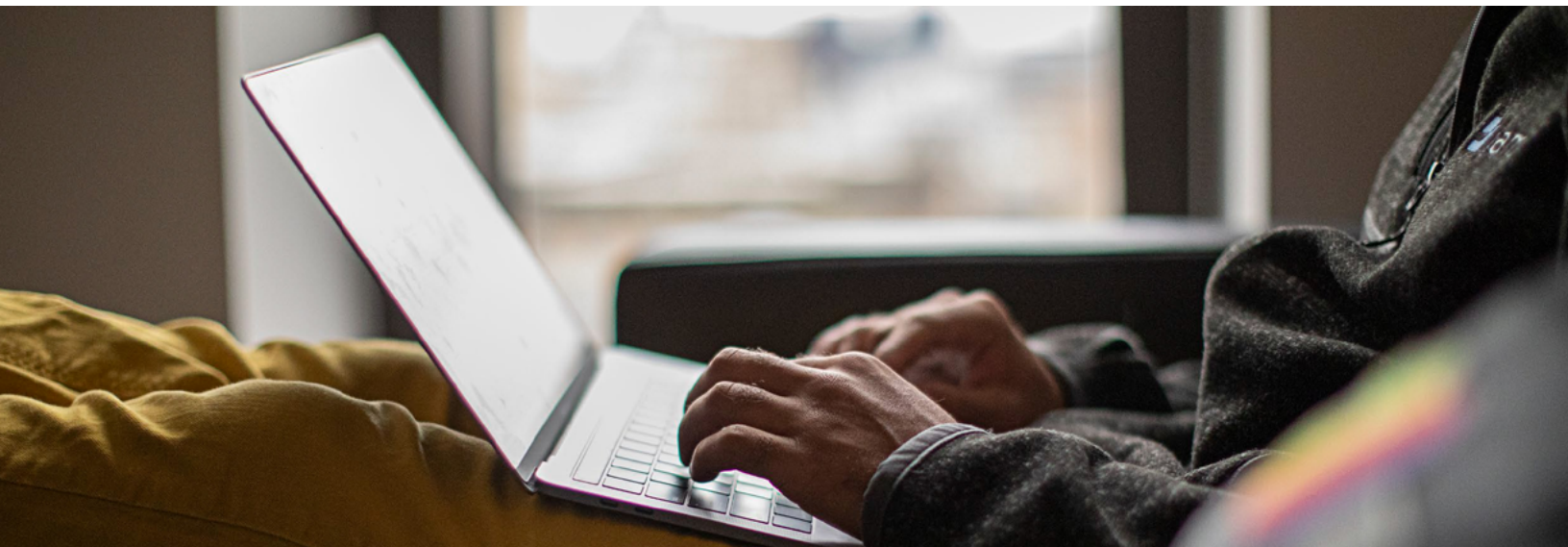
Utiliza "lectura por defecto" para leer el archivo, y luego especifica "Técnico", "Fecha" o "Departamento" para devolver el valor de un solo elemento. Esto es particularmente útil para algo como un Atributo de Extensión, donde un administrador podría usar este comando en un script corto para que por medio del echo enviar el resultado de vuelta a Jamf Pro.

Ejemplo:

```
/usr/bin/defaults read~/Desktop/provision.plist Fecha
```

El Terminal sólo mostrará la fecha:

```
20-06-13
```



Uso de los parámetros de los scripts en Jamf Pro

¿Qué son los parámetros de los scripts?

Vamos a demostrarlo con un script rápido.

```
#!/bin/zsh
```

El shebang es zsh o zee-shell, que suena como "síshel".

Añadiré una sentencia echo, que me dará una línea vacía para que sea más fácil de ver:

```
echo
```

Luego añadiré otra sentencia echo con las variables de parámetro "1" a "5".

```
echo $1 $2 $3 $4 $5
```

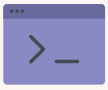
Recuerde que cuando algo en un script comienza con un "$" suele ser un indicador de que algo es una variable o un marcador de posición para algo.

El script completo:

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

Guarda el script en el escritorio como parameters.zsh.

Cada vez que usted crea un nuevo archivo de script, tiene que utilizar un comando en el Terminal llamado "chmod" para hacerlo ejecutable. De lo contrario, Terminal considera que se trata de un simple archivo de texto en lugar de un script que puede ejecutar.



Utilice su editor de código favorito o cualquier editor de texto simple como BBEdit o incluso TextEdit que viene con su Mac. Desactive todos los ajustes de autocorrección que transforman las comillas rectas en cursivas. Solo necesitamos comillas rectas ahora. (Un editor de scripts lo hace automáticamente).

Escriba siempre al principio de cada script algo llamado shebang: llamado hash + bang #! Seguido de /bin/zsh (o bash o el tipo de script que esté utilizando), para que cuando utilice sus scripts, Terminal sepa que debe utilizar seashell (o el tipo de script que esté utilizando) como intérprete.

Escriba la palabra "chmod", que significa "cambiar el modo", y luego añada +x, que significa hacerlo "ejecutable", y arrastre su script a la ventana.

Ahora, cuando ejecute el script seguido de "María tenía un corderito", te mostrará el echo de "María tenía un corderito:"

```
chmod +x ~/Desktop/parameters.zsh María tenía un corderito
María tenía un corderito
```

Esto no es muy emocionante, así que vamos a divertirnos un poco.

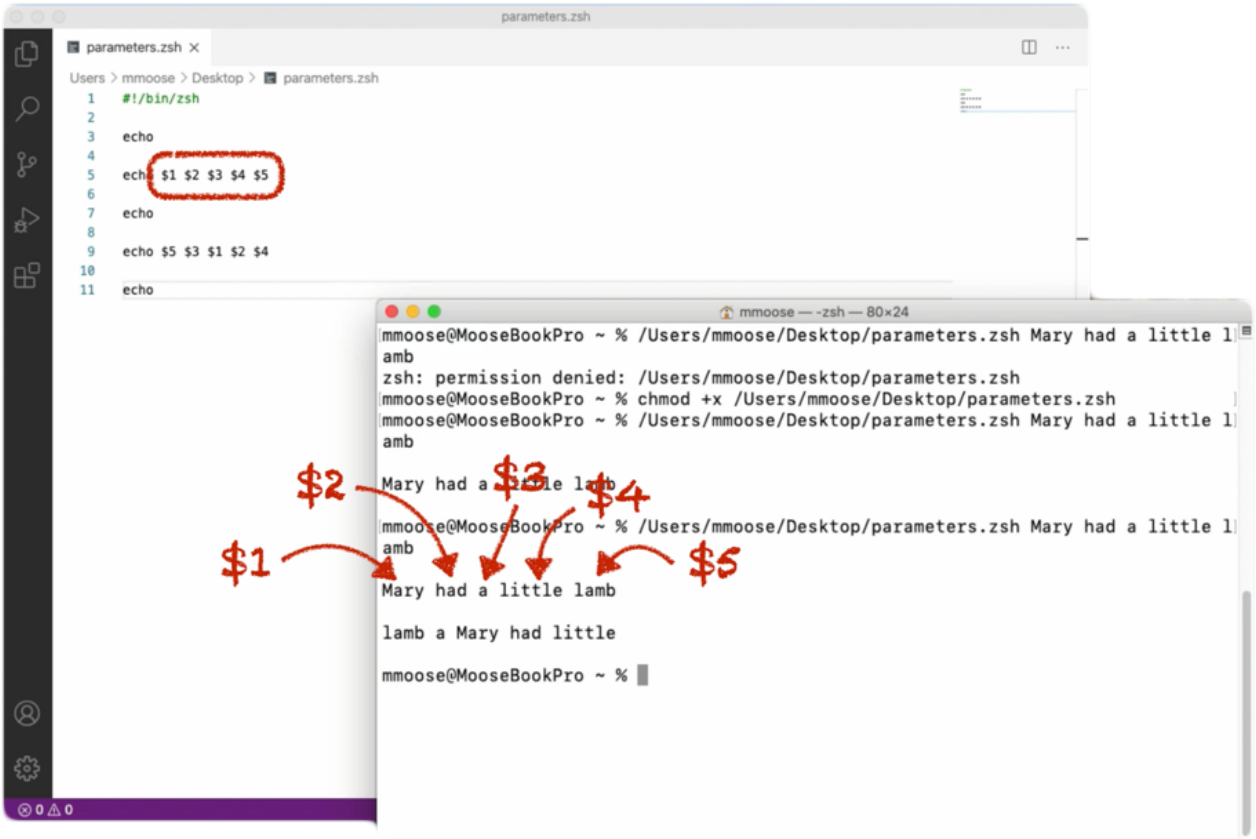
De nuevo en el script, añada otra orden echo. Pero esta vez, mezcle el orden de esas variables de script:

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

Ahora guarde y ejecute de nuevo el script en Terminal seguido de "María tenía un corderito".

```
~/Desktop/parameters.zsh María tenía un corderito
```

Esto es lo que verá:



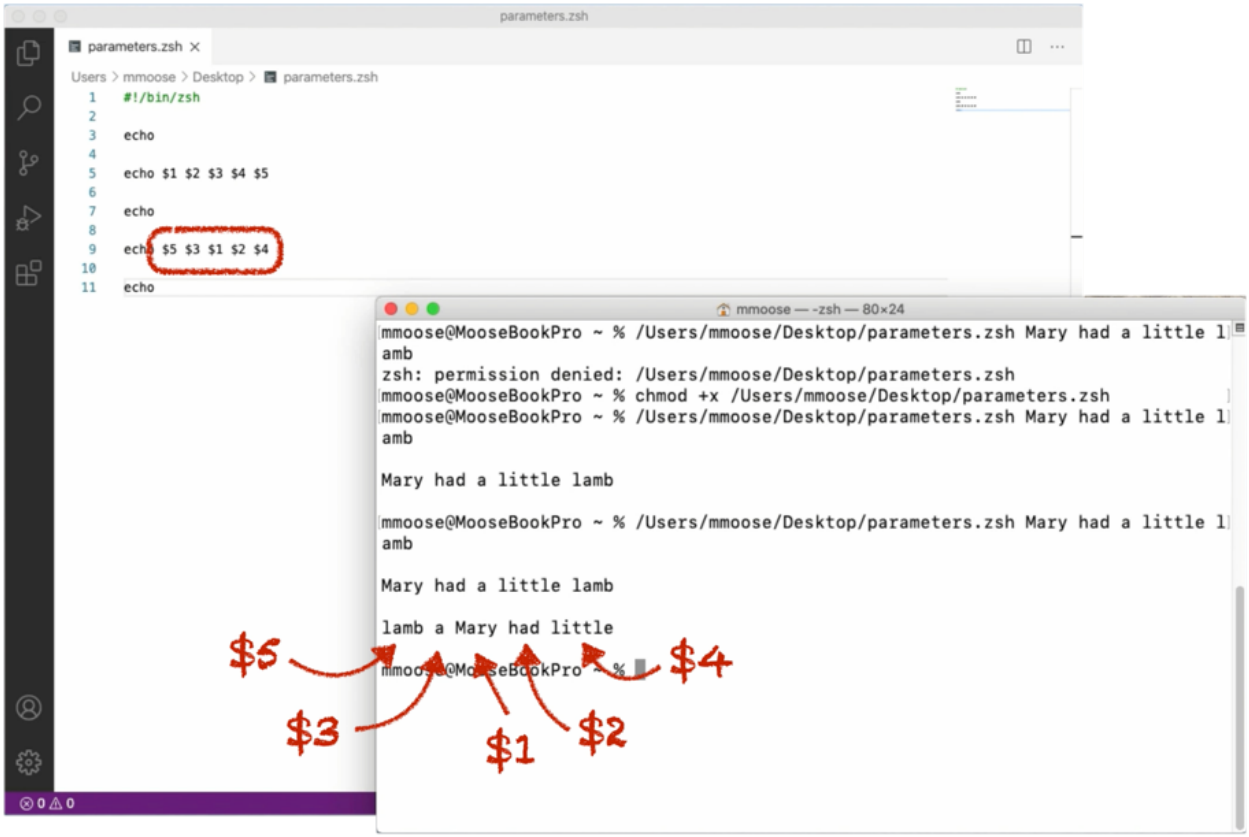
La primera línea parece correcta, pero ahora la segunda está desordenada. Lo que nos está diciendo es que cada una de esas variables numéricas del script —$1-2-3-4-5— se corresponde con el orden de los elementos que añadimos al final del script.

"María" es la primera palabra y es "$1".

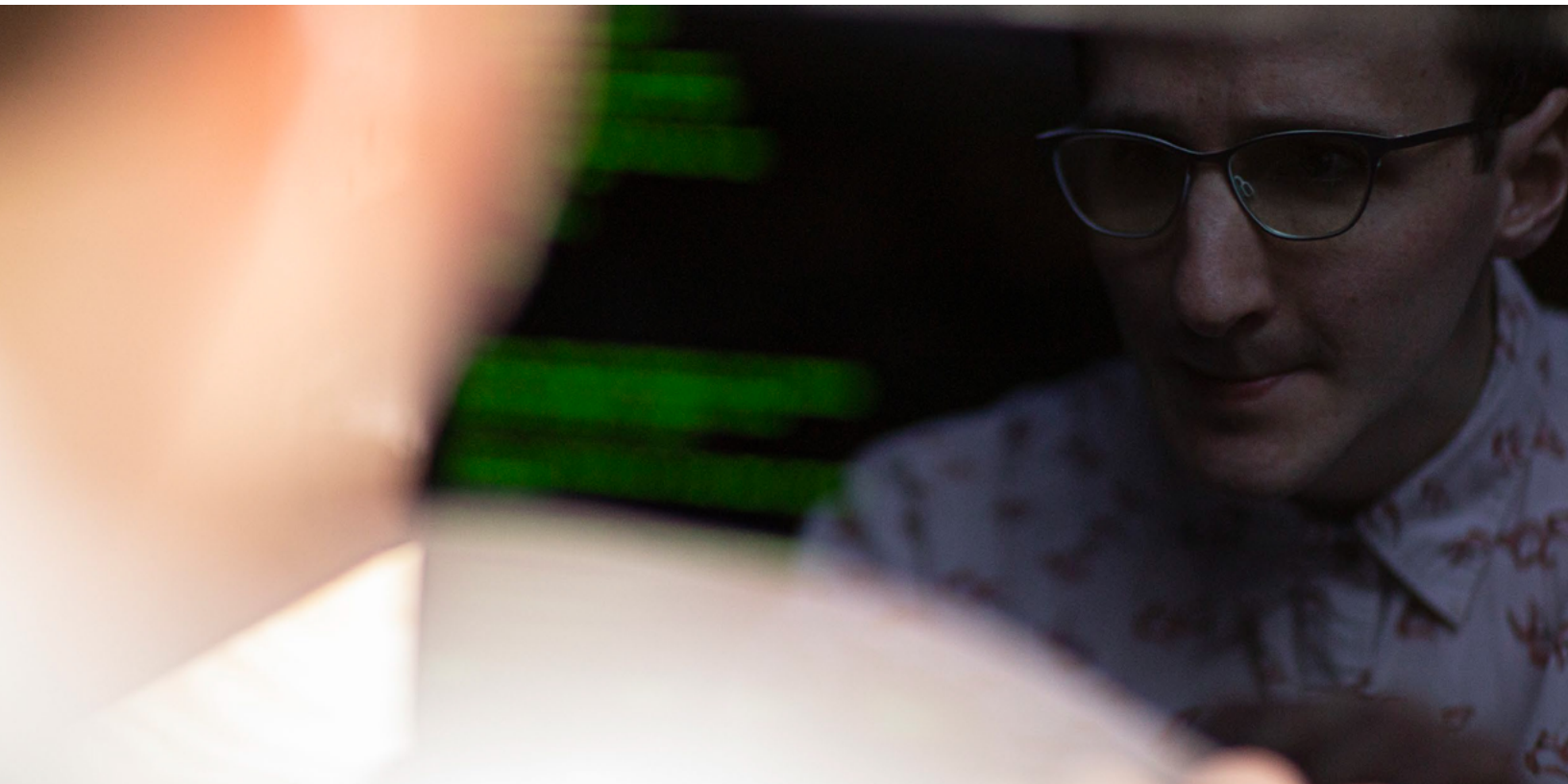
"tenía" es la segunda palabra y es "$2", etc.

Cada una de las palabras de "María tenía un corderito" es un "parámetro de script". Y podemos referenciar cada parámetro del script por su posición a continuación del script.

Si cambia el orden de las variables en el script, el orden de las palabras cambia cuando ejecute el script.



Ahora que sabemos que los parámetros de los scripts tienen que ver con el orden de los elementos que siguen a un script, ¿cómo utilizamos esto en Jamf Pro? Echemos un vistazo.



Encontrar scripts existentes

Puede encontrar todo tipo de scripts existentes que otros administradores de Jamf Pro han creado en jamfnation.com: recursos ➔ complementos de Jamf Pro ➔ Scripts.

Desde aquí, puede navegar o utilizar la función de búsqueda para lo que quiera hacer.

En este caso de ejemplo, utilizaremos un script de parámetros de "zona horaria". Esto es lo que hará:

- Descargue el script
- Abra el archivo
- Seleccione todo el contenido
- Copie

En tu instancia de Jamf Pro, navegue a:

Ajustes > Administración de computadoras > Scripts > Nuevo script

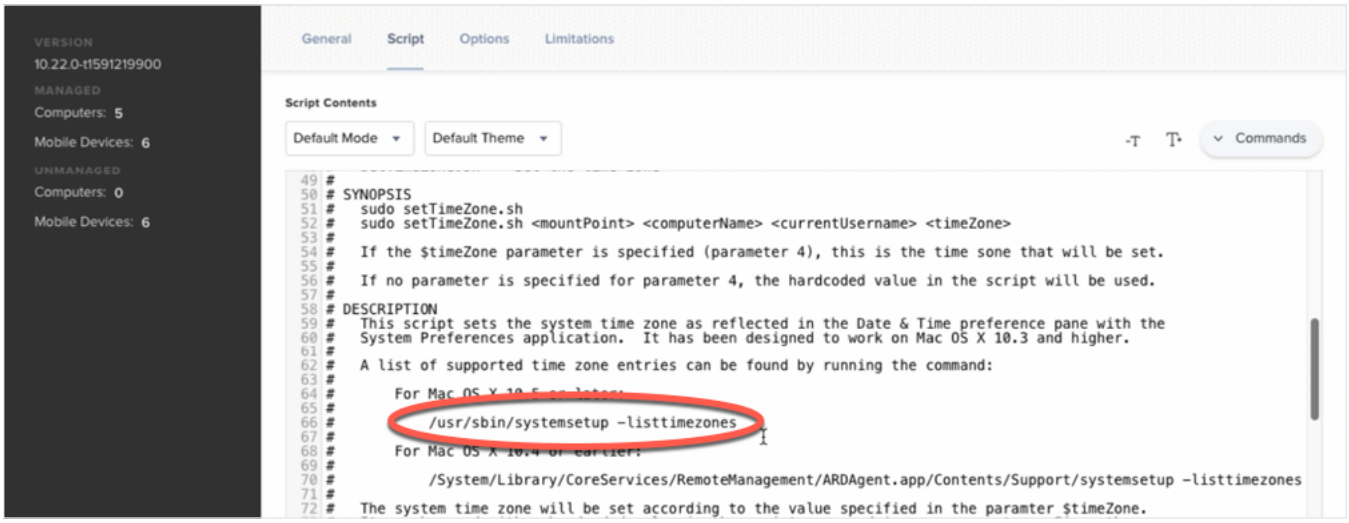
- Nombre el nuevo script "Establecer zona horaria".
- En la pestaña Scripts, pegue el script.
- En la pestaña "Opciones", haga clic en el campo "Parámetro 4" y establezca el nombre de la etiqueta en algo como "Zona horaria". Puede ponerle el nombre que quieraa, pero tiene sentido ponerle un nombre significativo. En este caso, cuando ejecute el script, va a establecer la zona horaria que quiera en el cuarto parámetro o "$4".
- Guarde.
- Vuelva a seleccionar la pestaña Scripts.
- Busque el comando para listar las zonas horarias que desee copiar.

Entonces, ¿por qué empiezan con el parámetro "4" y no con el "1"?

Si se fija bien en la letra pequeña que hay justo encima de las etiquetas de los parámetros, veráa un texto que dice "Los parámetros del 1 al 3 están predefinidos como punto de montaje, nombre de computadora y nombre de usuario".

Eso significa que Jamf Pro siempre enviará esa información como los tres primeros parámetros con cada script, ya sea que el script los utilice o no.

No tiene control sobre esos parámetros, pero sí sobre el resto. Tiene hasta 8 parámetros más que puede definir como quiera.



Cómo ejecutar el comando

- Abra Terminal
- Escriba "sudo", que significa ejecutar con privilegios elevados, y luego pegue el comando:

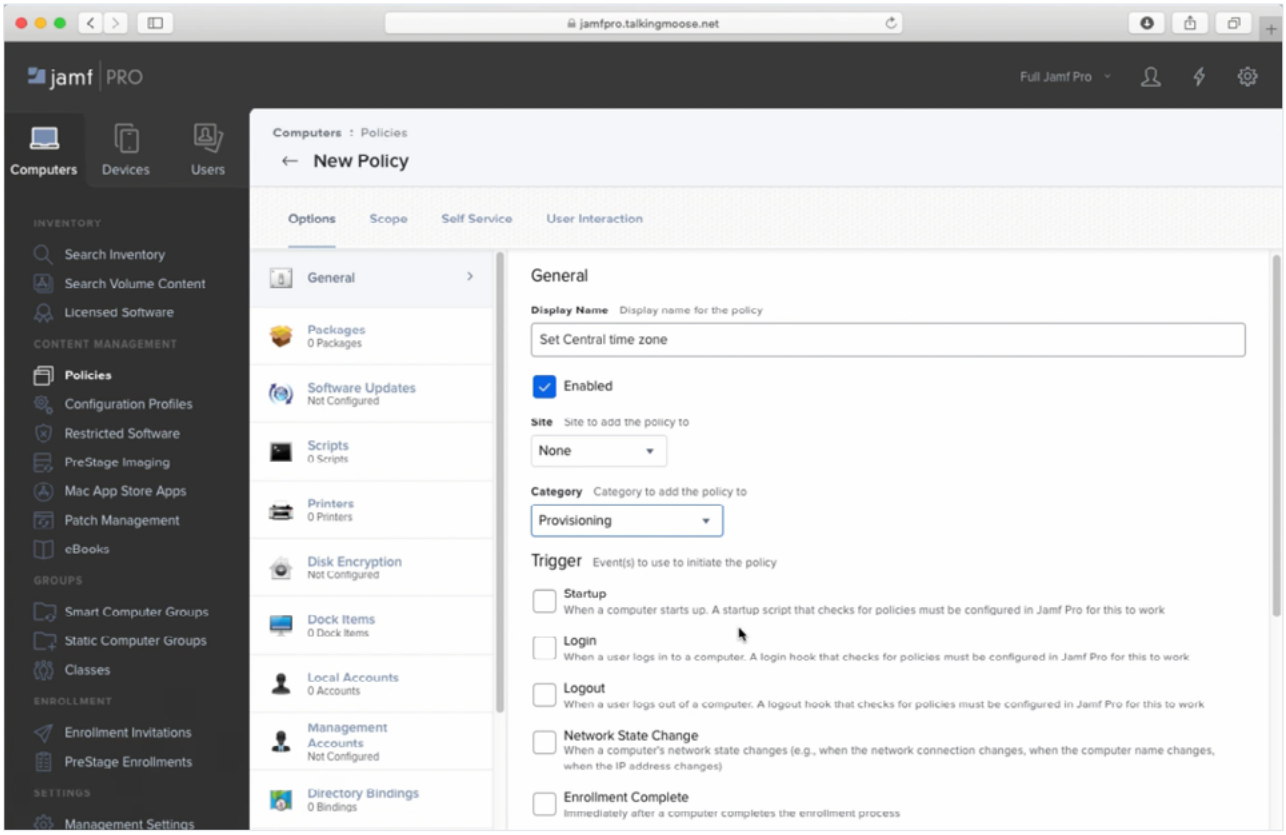
```
/usr/sbin/systemsetup -listtimezones
```

Esto mostrará una larga lista de zonas horarias disponibles en el formato correcto para el script. En este ejemplo, elegimos Chicago, que es representativa de la zona horaria central de Estados Unidos, y la copiamos.

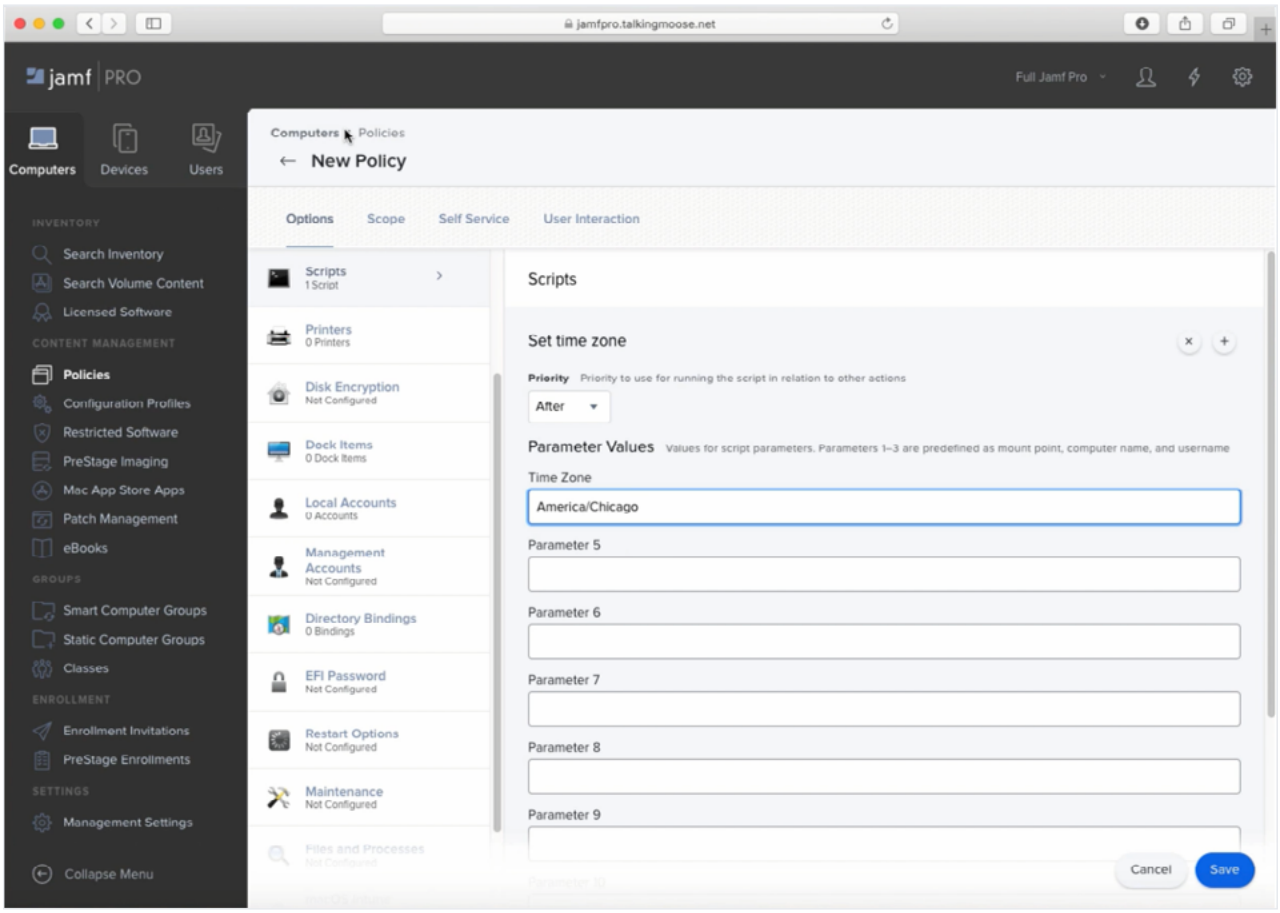
Crear una política

Ahora, vamos a crear una política para ejecutar nuestro script.

- En su instancia de Jamf, seleccione "Políticas" en el menú de la izquierda, y luego la categoría. En este ejemplo, lo añadiremos a la categoría "Aprovisionamiento".
- Póngale el nombre de "Establecer zona horaria central" y añádala a la categoría "Aprovisionamiento".



- Establezca un activador personalizado que me permita llamar a esta política por su nombre en un script posterior, como setcentraltimezone.
- Seleccione la opción Scripts y péguela en la zona horaria que haya copiado del comando Terminal.



Observe que este campo muestra la etiqueta que añadimos en la pestaña Opciones cuando pegamos el script. La etiqueta le dice lo que debe poner aquí.

Todo lo que tiene que hacer ahora es ampliar la política y guardarla.

Y ahora tiene una nueva política en su categoría de "Aprovisionamiento".

Los parámetros de los scripts nos permiten hacer algo realmente genial.

Como el script fue escrito para aceptar parámetros de script, puede reutilizarlo tantas veces como quiera para diferentes zonas horarias. Lo único que tiene que hacer es crear una nueva política para cada zona horaria, añadir el mismo script y rellenar el valor de la zona horaria.

Crear diálogos con jamfHelper y osascript

Los diálogos son útiles no sólo para mostrar un mensaje a sus usuarios finales, sino también para solicitarles información.

Aquí tiene dos formas de hacer diálogos, cada una con sus propias ventajas.

jamfHelper

jamfHelper es una herramienta de línea de comandos.

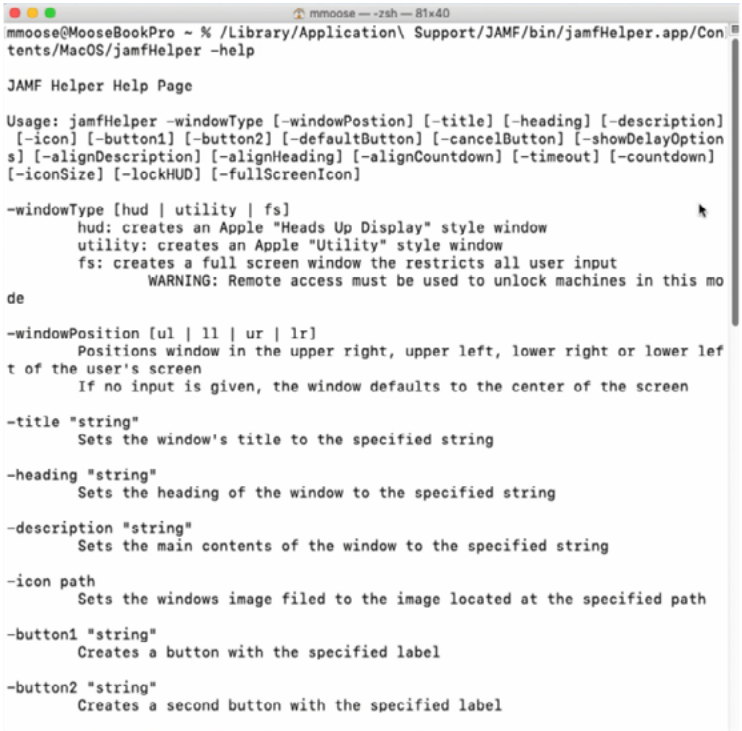
Abra Terminal. Después:

- Vaya a Biblioteca > Soporte de aplicaciones > JAMF > bin > jamfHelper
- Haga clic con el botón derecho del mouse en jamfHelper y elija "Mostrar contenido del paquete".
- Vaya a Contenido > MacOS, donde encontrará la herramienta de línea de comandos.
- Arrástrelo al Terminal.

Al final de la línea de comandos que se rellene, añada "-help" y oprima retorno:

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

Esto le mostrará todo lo que necesita saber para utilizar jamfHelper.



Como la ruta de acceso a jamfHelper es tan larga, puede ponerla en un nombre de variable más corto llamado simplemente jamfHelper con este script:

```
#!/bin/zsh
```

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

A partir de aquí, para llamar a jamfHelper, sólo tiene que poner un signo de pesos delante del nombre de su variable y colocarlo entre comillas dobles:

```
"$jamfHelper"
```

Aquí tiene un buen truco para obtener el nombre de la ruta correcta. Haga clic con el botón derecho del mouse en jamfHelper y señale "Copiar", pero luego oprima también la tecla Opción para copiar su ruta en su lugar. En este script, lo único que tiene que hacer es pegar la ruta entre las comillas.

¿Qué puede hacer jamfHelper?

Digamos que quiere añadir algunas opciones a jamfHelper.
La primera es definir un tipo de ventana: admite tres.
Para empezar, nos quedaremos con "Heads-Up Display".
Añada -windowType a su script, y el tipo de ventana: "hud".

```
"$jamfHelper" -windowType hud  \
```

Una diagonal invertida al final de una línea le permite envolver lo que normalmente sería un comando muy largo en varias líneas.

A continuación, añada un título. Debe ir entre comillas:

```
-heading "Preparando su computadora…" \
```

Luego, añada una descripción.

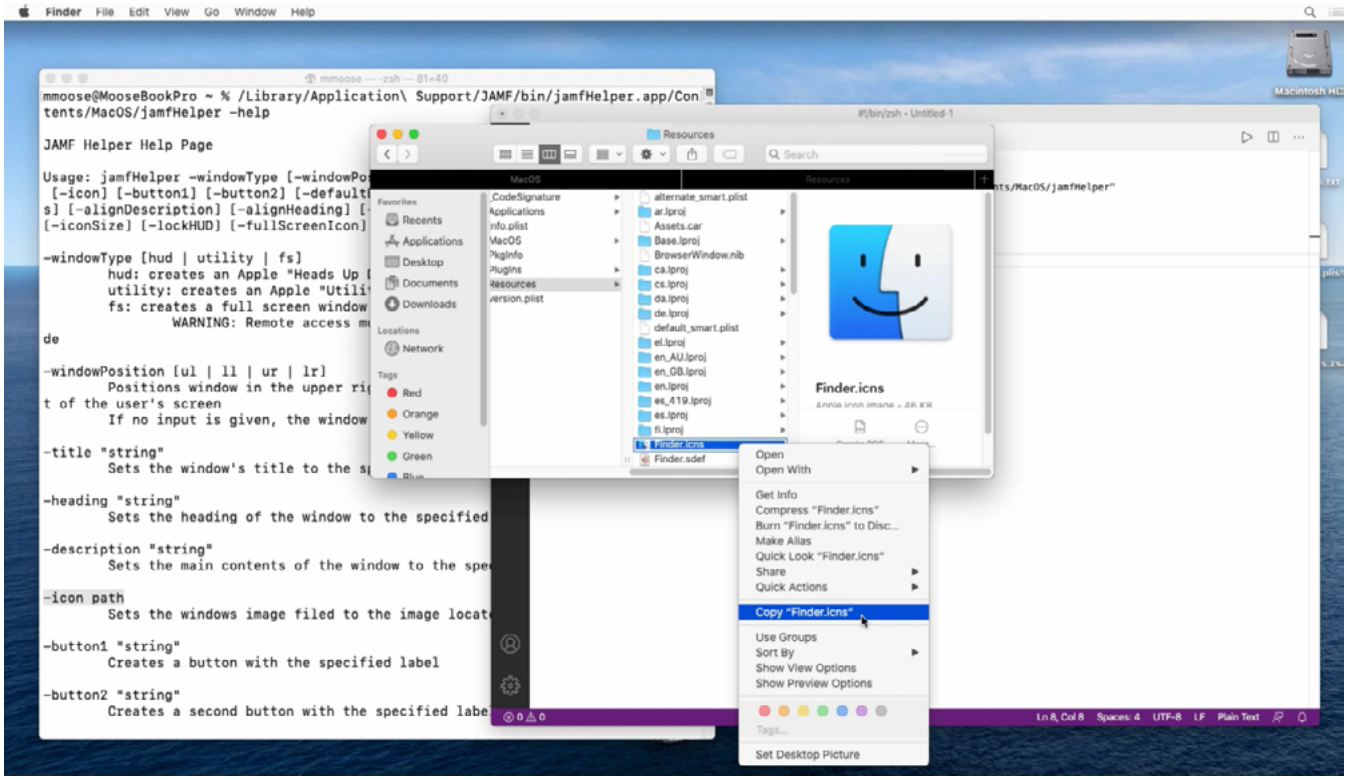
```
-description "instalando Microsoft Office 2019"\
```

Si desea añadir interés al cuadro de diálogo, añada un icono. Elija un icono y coloque su ruta aquí. Ponga la ruta entre comillas en caso de que contenga espacios.

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

Cómo encontrar iconos ya existentes en su Mac

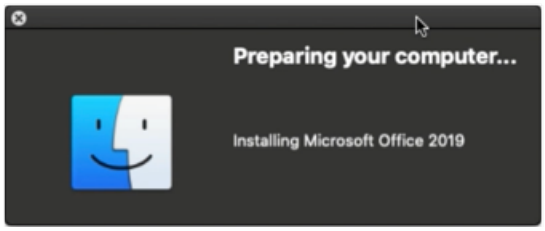
- En el Finder, vaya a Sistema > Biblioteca > CoreServices y localice la app Finder.
- Haga clic con el botón derecho del ratón en el Finder, elige "Mostrar contenido del paquete" y luego Contenido > Recursos
- Seleccione el icono del Finder.
- Haga clic con el botón derecho del mouse en el icono del Finder, mantenga oprimida la tecla Opción y copie la ruta.
- Luego, pega la ruta en tu script.



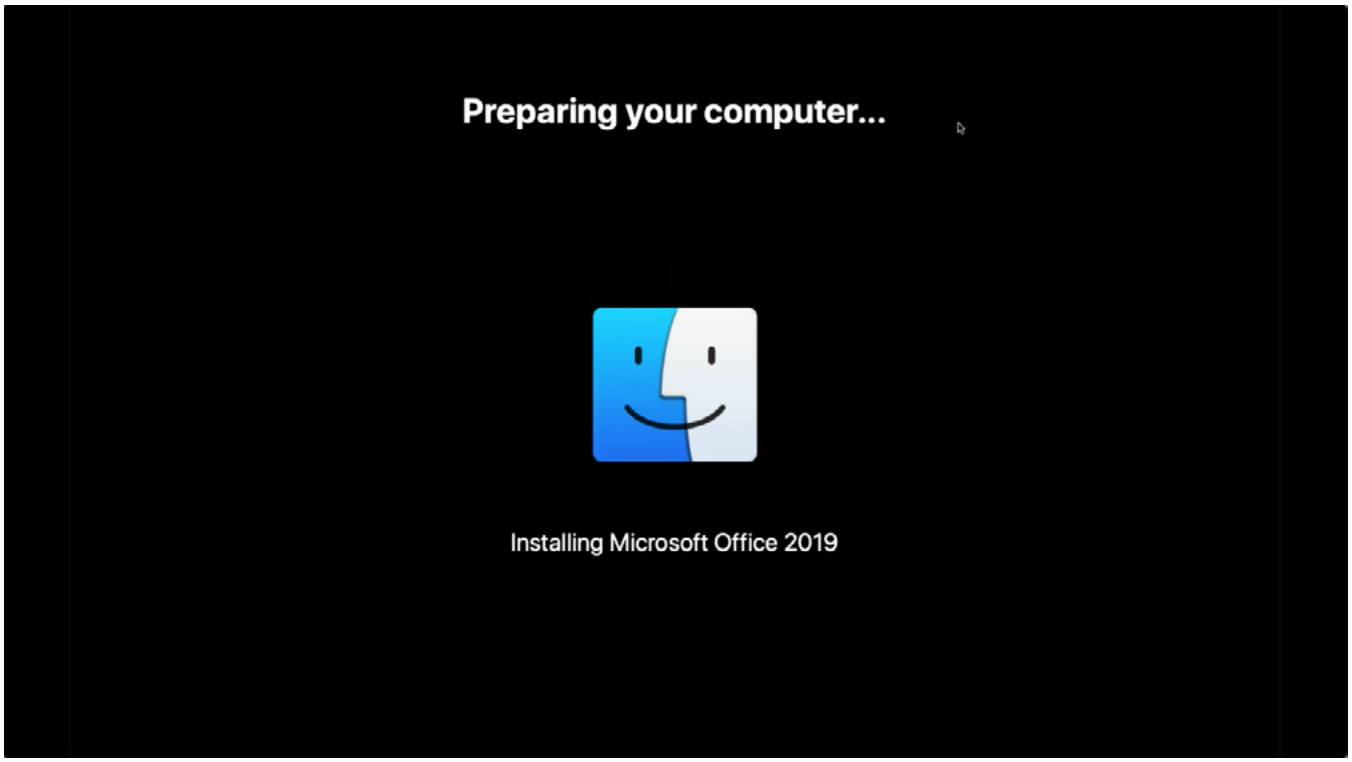
Así es como se ve todo junto:

```
#!/bin/zsh

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

"$jamfHelper" -windowType hud  \
-heading "Preparando su computadora…" \
-description "instalación de Microsoft Office 2019"\
-icon "/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns"
```

Ejecute su script directamente desde su editor de scripts.
Obtendrá un diálogo muy sencillo:



Para obtener un diálogo a pantalla completa, cambie el tipo de ventana de "hud" a "fs",
que significa "Pantalla completa", y ejecútelo de nuevo:



Un diálogo a pantalla completa como éste es una buena forma de evitar que la persona que está frente a la computadora lo utilice mientras se está aprovisionando. También puede actualizarse para informar al usuario de los progresos realizados mientras se ejecutan tareas como la instalación de software.

Si en algún momento necesita ver lo que ocurre detrás del diálogo, sólo tiene que pulsar Command + q para salir.

El comando osascript

También puede crear diálogos a través de AppleScript con el comando osascript desde el Terminal.

Puede hacer algo que jamfHelper no puede.

En primer lugar, cree un comando para un diálogo de "elección", incluyendo tres nombres de departamentos.

Debe utilizar comillas dobles para empezar y terminar este comando, y cuando también tenga comillas dobles como parte del comando, tendrá que añadir diagonales invertidas antes de ellas para tratarlas como comillas dobles literales, no las comillas dobles que rodean al comando:

```
asCommand="elige de la lista {\"Contabilidad\", \"Ventas\", \"Quiosco\"}
```

A continuación, añada un mensaje que se mostrará al usuario final, explicando lo que va a hacer la Mac:

```
con el aviso \"Hola, vamos a preparar su Mac.
Para empezar, elija su departamento a continuación...\"
```

La última parte del comando añade un título a la ventana de diálogo. Acuérdese de poner la comilla doble de cierre al final del comando:

```
con el título \"Prepare su Mac\""
```

Esta línea realmente le dice a osascript que ejecute el comando y luego ponga el resultado en la variable "departamento":

```
departmento=$( /usr/bin/osascript -e "$asCommand" )
```

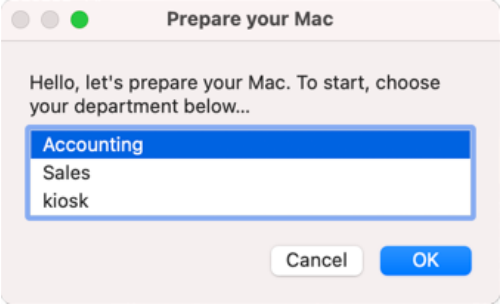
Por último, haga eco de la variable "departamento" para mostrar lo que el usuario ha seleccionado:

```
echo "$departamento"
```

Este es el aspecto del script completo:

```
#!/bin/zsh

asCommand="elija de la lista {\"Contabilidad\", \"Ventas\", \"Quiosco\"} con el mensaje \
"Hola, vamos a preparar su Mac. Para empezar, elija su departamento a continuación... \"
con el título \"Prepare su Mac\""

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

echo "$departamento"
```

Ahora, pruebe el script en su editor de scripts. Aquí el usuario eligió "Contabilidad",
y este es el resultado que verá:



AppleScript y osascript pueden hacer mucho más con los diálogos, pero esta es una buena manera de empezar.

Pongámoslo todo junto

Aquí tiene un script que puede añadir al Self Service como método para permitir que un nuevo empleado en iniciación aprovisione su Mac eligiendo su departamento.

En primer lugar, recoge el nombre completo y el nombre de usuario del usuario conectado. A continuación, utilizará el nombre completo en la petición osascript para elegir un departamento:

```
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" ) # por ejemplo "mmoose"

echo "El usuario de aprovisionamiento es $fullName ($currentUser)"
```

A continuación, pida al usuario que elija un departamento:

```
asCommand="elija de la lista {\"Contabilidad\", \"Ventas\", \"Quiosco\"} con el mensaje
\"¡Hola, $fullName! Vamos a preparar su Mac. Para empezar, elija su departamento
a continuación... \" con el título \"Prepare su Mac\""

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

echo "Aprovisionamiento para el departamento $departamento"
```

Otro osascript pide la etiqueta de activo de la computadora:

```
asCommand="texto devuelto de (mostrar el diálogo \"Introduzca la etiqueta de activos de esta Mac (ver parte inferior de la computadora)...\" respuesta preestablecida \"\" con el título \ "Prepare su Mac")"

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

echo "La etiqueta de activos del dispositivo es $assetTag"
```

El script ajustará la zona horaria de todas las computadoras:

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

Y luego evaluará el departamento elegido e instalará el software necesario:

```
if [[ "$departamento" = "Contabilidad" ]]; then
          echo"Aprovisionamiento de este Mac para contabilidad"

          /usr/local/bin/jamf policy -event installchrome
          /usr/local/bin/jamf policy -event installoffice

elif [[ "$departamento" = "Ventas" ]]; then

          echo "Aprovisionamiento de este Mac para ventas"

          /usr/local/bin/jamf policy -event installchrome
          /usr/local/bin/jamf policy -event installoffice
          /usr/local/bin/jamf policy -event installzoom

else

          echo "Aprovisionamiento de este Mac como quiosco"

          /usr/local/bin/jamf policy -event installchrome
          /usr/local/bin/jamf policy -event installzoom

fi
```

Una vez instalado el software, se ejecutará una actualización del inventario que incluirá la etiqueta del activo, el departamento y el nombre de usuario de la computadora:

```
"$jamfHelper" -windowType "fs" \
-heading "Preparando su Mac..." \
-description "Actualizando el inventario" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &

/usr/local/bin/jamf recon -assetTag "$assetTag" -department "$department"
-endUsername "$currentUser"
```

A continuación, creará una carpeta en la Biblioteca para las herramientas administrativas y la información:

```
"$jamfHelper" -windowType "fs" \
-heading "Preparando su Mac..." \
-description "Creando carpeta de admin" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &

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

Y deje un recibo de aprovisionamiento:

```
"$jamfHelper" -windowType "fs" \
-heading "Preparando su Mac..." \
-description "Escribiendo el recibo de aprovisionamiento" \
-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"
```

Finalmente, informará al usuario y luego reiniciará la computadora:

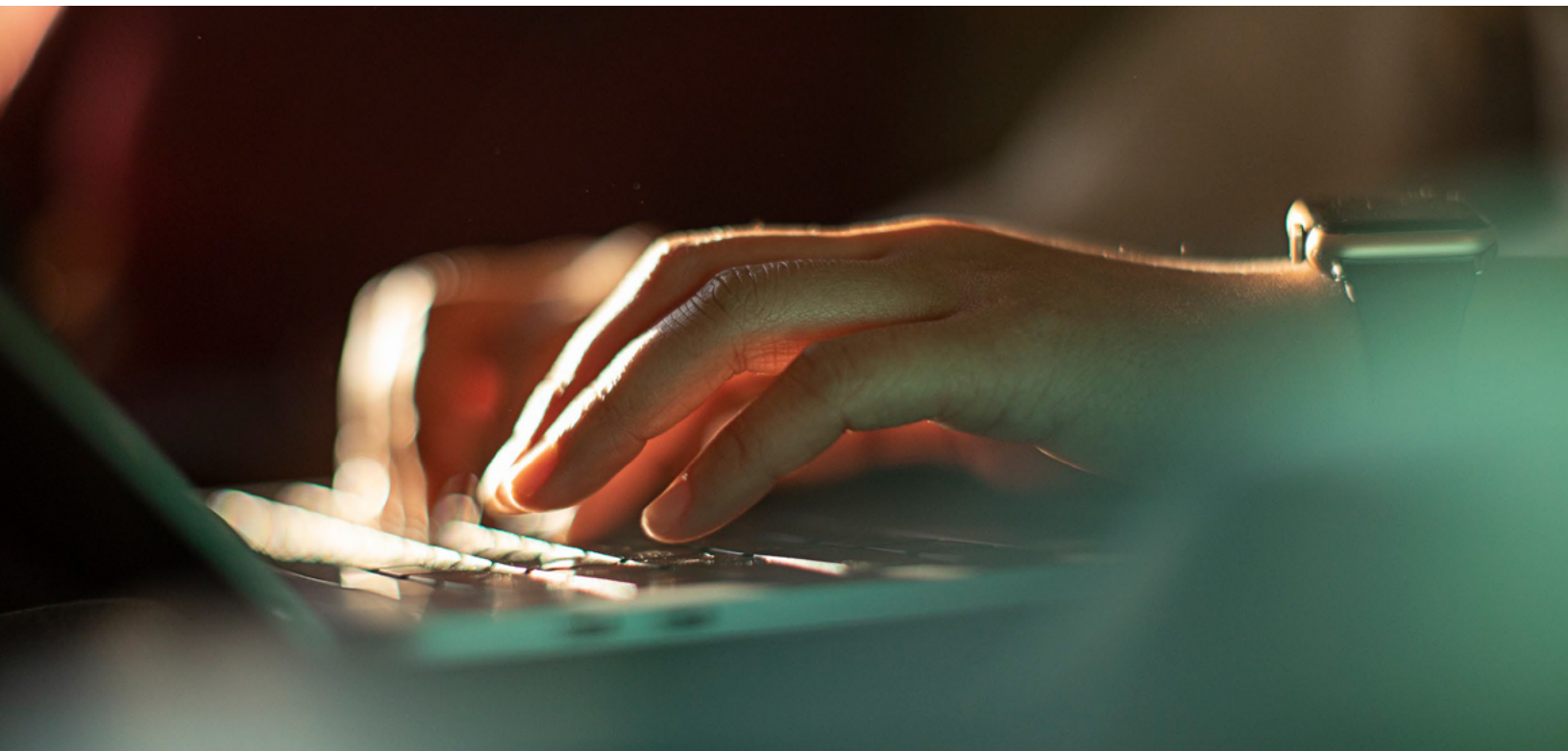
```
"$jamfHelper" -windowType "fs" \
-heading "Preparing your Mac..." \
-description "Reiniciando su Mac en un minuto" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &
/sbin/shutdown -r +1 &

exit 0
```

Script completo, con descriptores incrustados:

```
#!/bin/zsh

# asigna la ruta larga a JamfHelper a una variable más corta

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

# obtener información sobre el usuario actual

currentUser=$( /usr/bin/stat -f "%Su" /dev/console )
fullName=$( /usr/bin/id -F "$currentUser" ) # por ejemplo "Martin Moose"

echo "El usuario de aprovisionamiento es $fullName ($currentUser)"

# Comando AppleScript para pedir al usuario actualmente conectado que elija un departamento

asCommand="elige de la lista {\"Contabilidad\", \"Ventas\", \"Quiosco\"} con el mensaje
\"¡Hola, $fullName! Vamos a preparar su Mac. Para empezar, elija su departamento a
continuación... \" con el título \"Prepara tu Mac\""

# ejecuta el comando

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

echo "Aprovisionamiento para el departamento $departamento"

# Comando AppleScript para pedir al usuario actualmente conectado que introduzca una
etiqueta de activo

asCommand="text returned of (display dialog \"Introduzca la etiqueta de activo de esta Mac
(vea la parte inferior de la computadora)...\" default answer \"\" with title \"Prepare su
Mac\")"

# ejecuta el comando

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

echo "La etiqueta de activos del dispositivo es $assetTag"
```

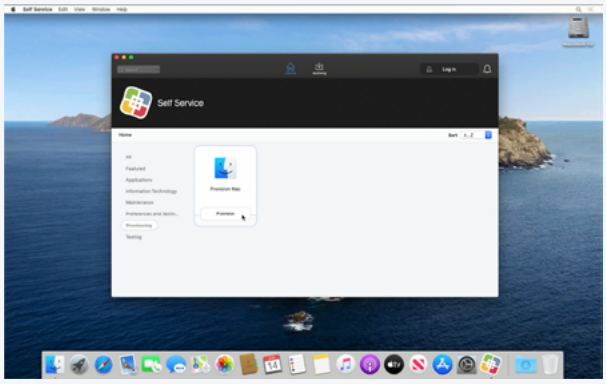
```
# prepara esta Mac para el departamento seleccionado

# ajustes globales

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

si [[ "$departamento" = "Contabilidad" ]]; entonces

          echo "Aprovisionamiento de este Mac para Contabilidad"

          /usr/local/bin/jamf policy -event installchrome
          /usr/local/bin/jamf policy -event installoffice

elif [[ "$departamento" = "Ventas" ]]; then

          echo "Aprovisionamiento de este Mac para Ventas"

          /usr/local/bin/jamf policy -event installchrome
          /usr/local/bin/jamf policy -event installoffice
          /usr/local/bin/jamf policy -event installzoom

else

          echo "Aprovisionamiento de este Mac como quiosco"

          /usr/local/bin/jamf policy -event installchrome
          /usr/local/bin/jamf policy -event installzoom

fi

# actualizar el inventario de Jamf Pro

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

/usr/local/bin/jamf recon -assetTag "$assetTag" -department "$department" -
endUsername "$currentUser"

# crear carpeta admin

"$jamfHelper" -windowType "fs" \
-heading "Preparando su Mac..." \
-description "Creando carpeta de admin" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &

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

# crear recibo de aprovisionamiento
```

```
"$jamfHelper" -windowType "fs" \
-heading "Preparando su Mac..." \
-description "Escribiendo el recibo de aprovisionamiento" \
-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 "Preparando su Mac..." \
-description "Se reiniciará su Mac en un minuto" \
-icon
"/System/Library/CoreServices/Finder.app/Contents/Resources/Finder.icns" &

# reinicia el Mac
/sbin/shutdown -r +1 &

exit 0
```

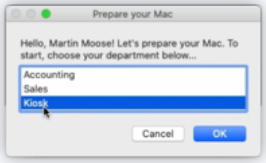


¡Vamos a ver cómo queda todo!

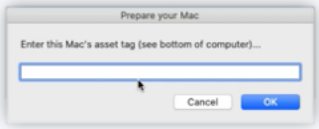
Con el Self Service abierto, inicie el proceso haciendo clic en el botón de aprovisionamiento.



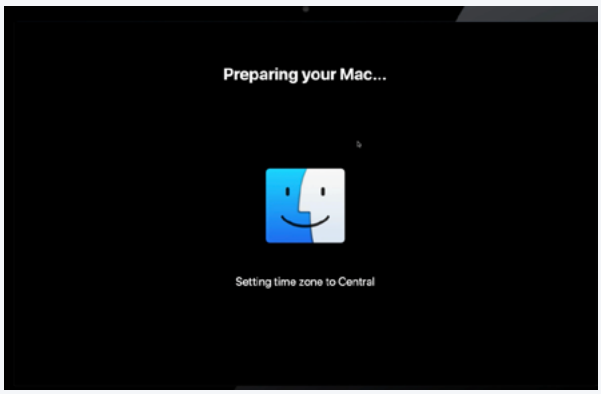
PASO 1.



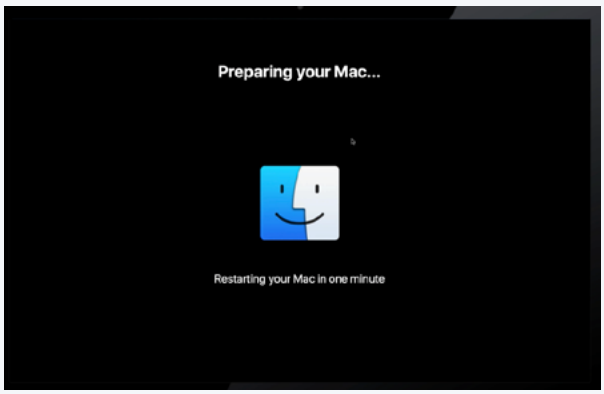
PASO 2.



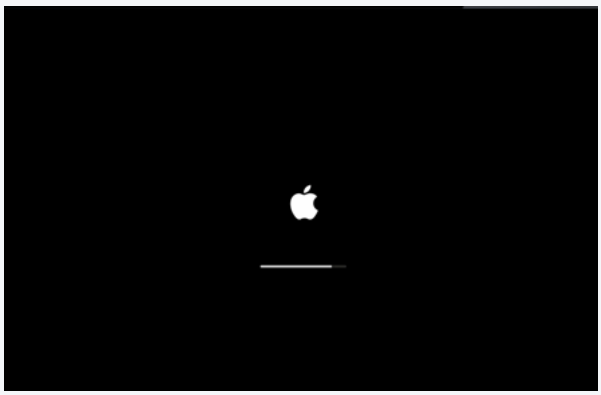
PASO 3.



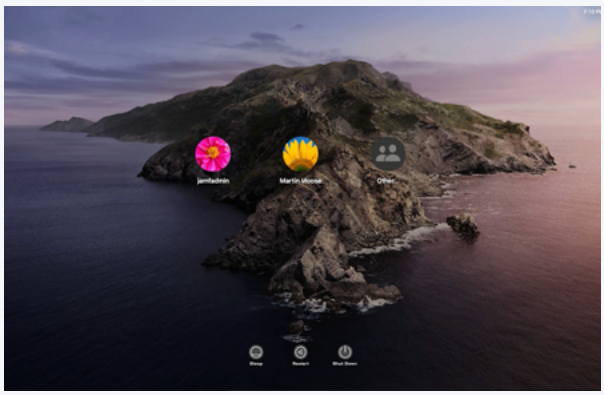
PASO 4.



PASO 5.



PASO 6.

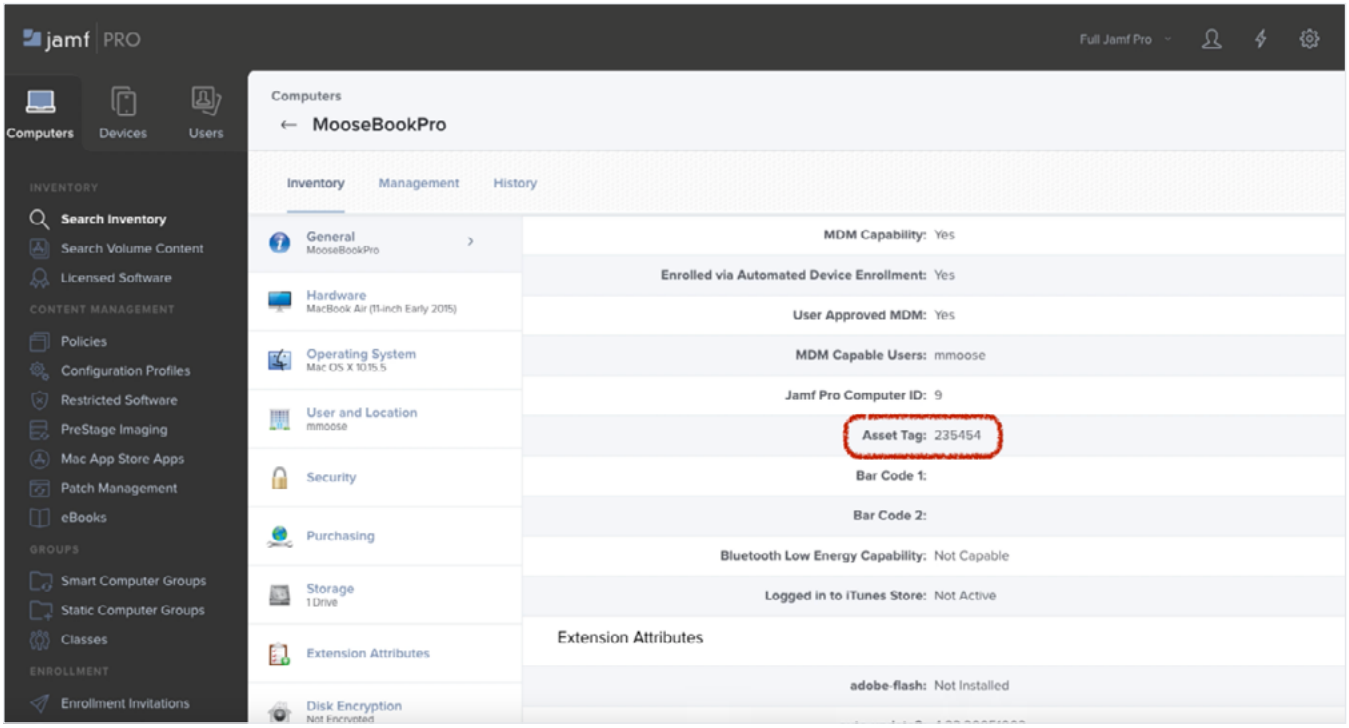


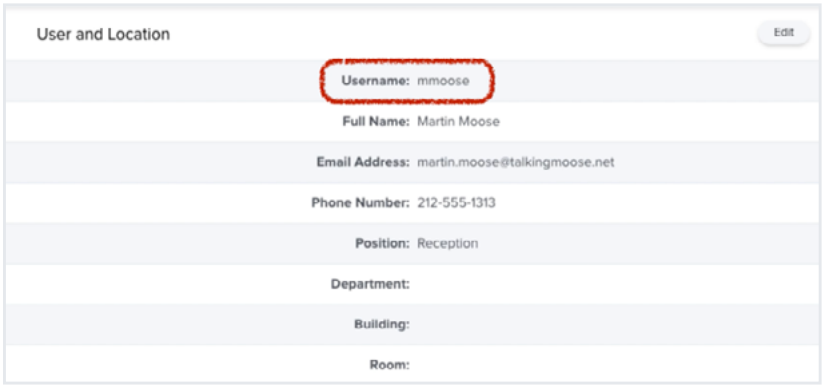
PASO 7.

Cuando termina el aprovisionamiento, se reinicia la computadora y la Mac vuelve a mostrar la ventana de inicio de sesión, lista para funcionar.

Después de que el usuario se conecte, su software se instala y puede ver su recibo de aprovisionamiento en la carpeta Biblioteca.

Cuando el administrador de Mac ve el registro de la comptadora en Jamf Pro, la etiqueta de activos se rellena desde el diálogo osascript, el campo de nombre de usuario se rellena, lo que permite al administrador hacer una búsqueda LDAP y extraer información adicional como la dirección de correo electrónico, el número de teléfono, el nombre de usuario principal y otros detalles de Active Directory.





¡Y eso es todo!

Otros recursos y formas de aprender

Catálogo de capacitación de Jamf

Mientras siga haciendo scripts, no olvide que todos los clientes de Jamf tienen acceso a nuestro catálogo de capacitación completo, con más de 15 horas de videos cortos sobre cómo hacerlo, que lo cubren todo, desde el servicio de asistencia hasta el ingeniero y el administrador: trainingcatalog.jamf.com

Eche un vistazo a la serie de scripts con 15 módulos y lecciones en video que se completan en menos de 30 minutos. Esto le ayudará a seguir aprendiendo a hacer scripts.

Github

Explore la comunidad de código abierto en GitHub.com. El software de código abierto es de uso gratuito, y encontrarás cientos de scripts y proyectos en la página GitHub de Jamf: github.com/jamf

Recursos del autor Bill Smith

Si desea ver muchos ejemplos de scripts cortos y largos para una variedad de necesidades, consulte el repositorio gist de Bill Smith en GitHub. Aquí es donde encontrará el script de aprovisionamiento de muestra de este documento técnico y algunos de los otros códigos que utilizó aquí también: https://gist.github.com/talkingmoose

Esperamos que esta guía le haya ayudado a llevar sus habilidades de scripts al siguiente nivel.

Los scripts combinados con un buen sistema de administración empresarial pueden marcar realmente la diferencia entre horas y horas de trabajo y el clic de un botón.

Solicitar prueba

Nos encantaría sugerirle Jamf Pro.