
JNUC
2019
Title slide uses the “Reveal” slide transition

To update, double-click to edit master

JIGSAW24

Mark Lamont

Professional Services Engineer
Jigsaw24

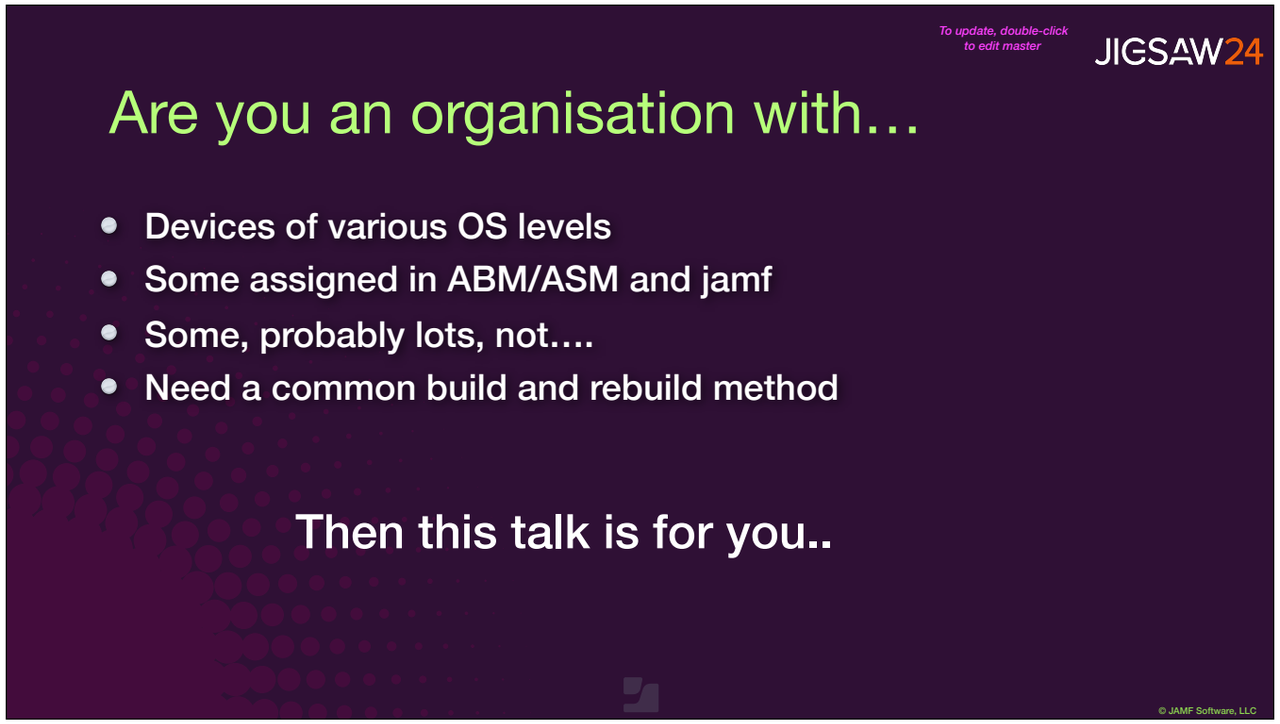
Jack Hollister

Apple Enterprise Solutions Architect
Jigsaw24

In session recording, Picture-in-Picture of you presenting will be placed here.

Please don't put anything especially important in this area.

Going to cover what an automatic build looks like
How to perform the same build with DEP assigned device and also non DEP assigned devices
How to get back to the start again using policies and smart groups and a bit of magic



Organisations often have different OS states
Hardware from many vendors acquired over many years so some is assigned to ABM/ASM and some not.
Maybe some could be assigned later, we'll cover that in this method later

But now for organisational requirements mean you need to catch up and standardise on common OS and keep up to date

To update, double-click to edit master

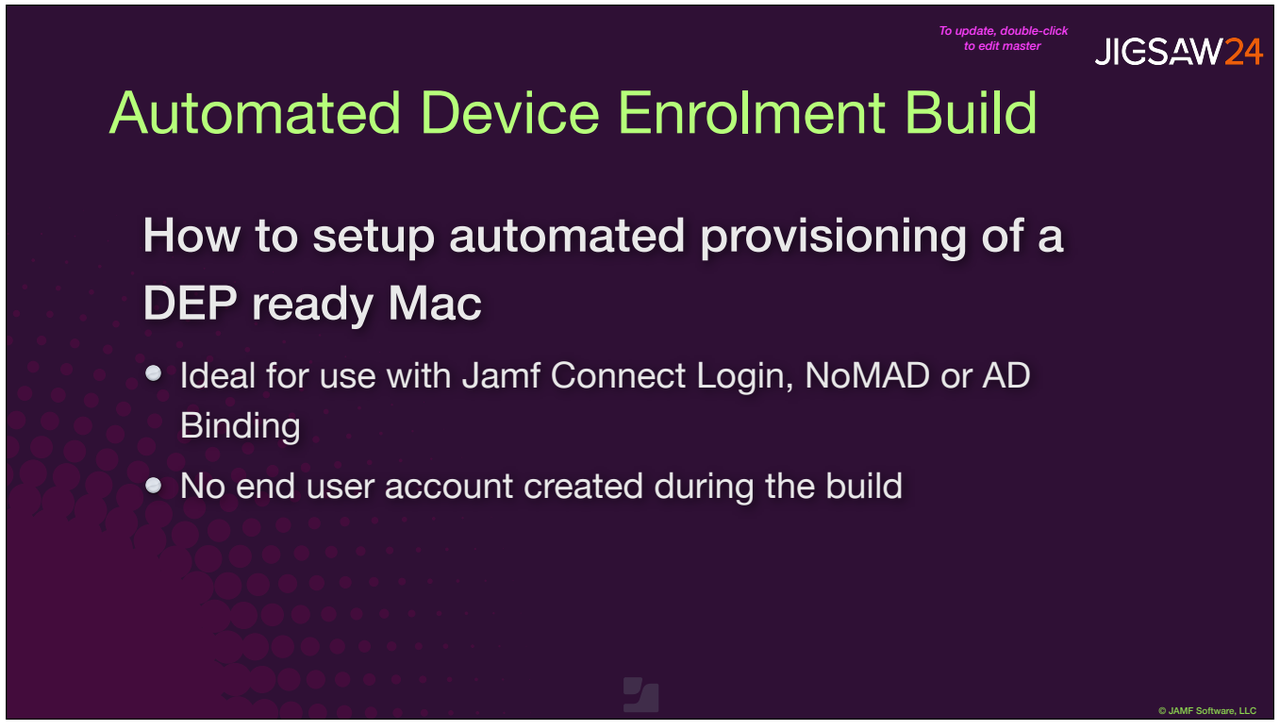
JIGSAW24

First Up

Out of the box DEP build….

Information only…

Click on



Going to show a minimal click **no user** account build process ending with Jamf connect azure login
You have probably seen or created variations on this already
Apologies if you have…..
This build prompts for a device type choice

JIGSAW24

Automated Device Enrolment

Firstly a custom package is needed….

Information only…

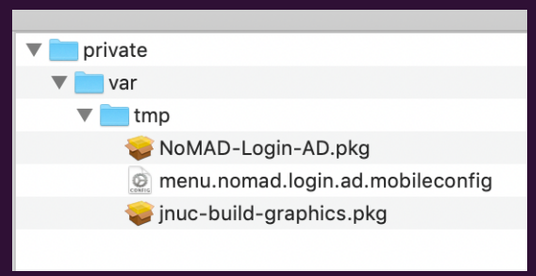
Click on

JIGSAW24

Automated DE Build - Package

Required components

- NoMAD Login
- NoMAD settings profile
- Branding graphics



To make this work we need a package.
Contains NoMAD login, images required during the build screens and a profile to configure NoMAD

Using NoMAD because we can use the UserInput logon screen method with this. Jamf connect did not have this at time of writing.

JIGSAW24

Automated DE Build - Package

Package Post Install Script

```
# perform Installs
log "installing profile"
/usr/bin/profiles -I -F "/var/tmp/menu.nomad.login.ad.mobileconfig"
log "installing graphics"
installer -pkg "/var/tmp/jnuc-build-graphics.pkg" -target /
log "installing nomad"
installer -pkg "/var/tmp/NoMAD-Login-AD.pkg" -target /



# set initial notify values
log "running Authchanger"
/usr/local/bin/authchanger -reset -preLogin NoMADLoginAD:UserInput NoMADLoginAD:Notify

/bin/echo "Command: MainTitle: Setting things up…"  >> /var/tmp/depnotify.log
/bin/echo "Command: MainText: Starting the device build, progress screens will display shortly." >> /var/tmp/depnotify.log
/bin/echo "Command: Image: "/Library/Management/jigsaw24/logo.png"" >> /var/tmp/depnotify.log
/bin/echo "Status: Please wait..." >> /var/tmp/depnotify.log
```

The package needs a post install script:

Installs the profile to configure NoMAD
Installs the graphics required by NoMAD
Installs NoMAD
Runs AUTHCHANGER to modify logon window process to configure NoMAD to use the USER INPUT method followed by NOTIFY method

USER INPUT method allows data entry at login window
NOTIFY method allows text and graphics over the login window

Sets initial text values for the NOTIFY screen

JIGSAW24

Enrollment Prestage

A prestage is needed….

Information only…

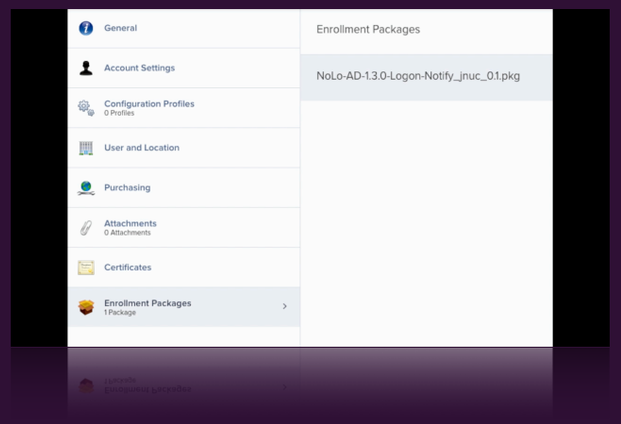
Click on

JIGSAW24

Automated DE Build - Prestage

Add the package to

Enrollment packages



Device enrolment allows for a package to be deployed at enrolment so deploy our package that way

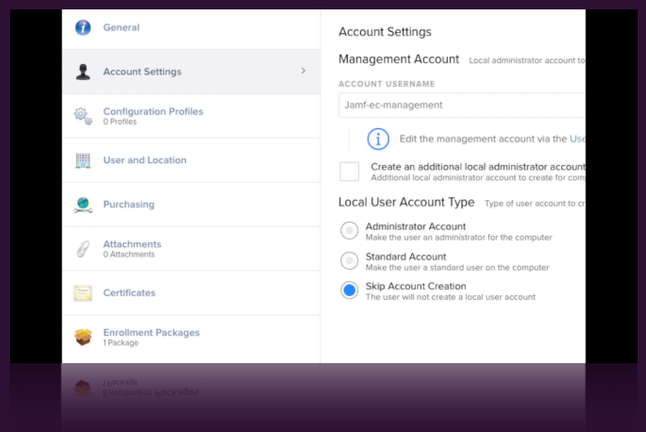
Add the package to the prestage
This gets downloaded and installed once the remote management agreement screen is agreed

JIGSAW24

Automated DE Build - Prestage

Set local User account creation to

Skip Account Creation



We don’t need any local user for this build as it will end up using another logon method.
If you want another local admin you can select this option

To update, double-click to edit master

JIGSAW24

Post Enrollment Policy

A build policy is needed….

Information only…

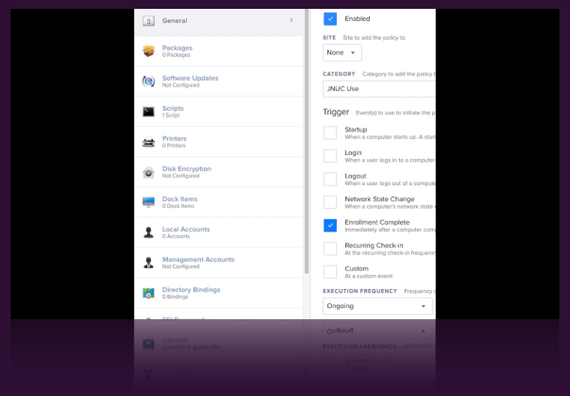
Click on

JIGSAW24

Automated DE Build - Policy

One policy required to run the build

- run on Enrollment Complete
- ongoing frequency
- Runs the build script



Policy for the build
runs after enrolment complete
Set to ongoing for reuse
One common policy for both DEP and NON DEP builds for simplicity
Only requires one script……..

JIGSAW24

Automated DE Build - Script

What does it do?

- waits until any required input is made
- Runs policies using custom triggers
- Displays status updates
- Sets login window type

script has several key sections and uses some repeated code templates

1. Waits for required input from the USER_INPUT method
2. Runs any build policies with custom triggers
3. Display progress using the NOTIFY method
4. Sets require login window type

Can’t see it? Don’t worry… we’ll break it down —>

JIGSAW24

Automated DE Build - Master Script

Let’s break it down

script has several key sections and uses some repeated code templates

1. Waits for required input from the USER_INPUT method
2. Runs any build policies with custom triggers
3. Display progress using the NOTIFY method

Can’t see it? Don’t worry… we’ll break it down —>

JIGSAW24

Automated DE Build - Script

Wait until user input completed

```
# Wait for the user data to be submitted...
while [[ ! -f /var/tmp/userinputoutput.txt ]]; do
    log "Waiting for user data..."
    /bin/sleep 2
done
```


First snippet: prevent the script from progressing until the initial required selection is made.
Reason is the enrolment policy can run before the selection is made so need to wait.
In our demo prompting for a device role.
The path to read data from is set in the NoMAD profile

JIGSAW24

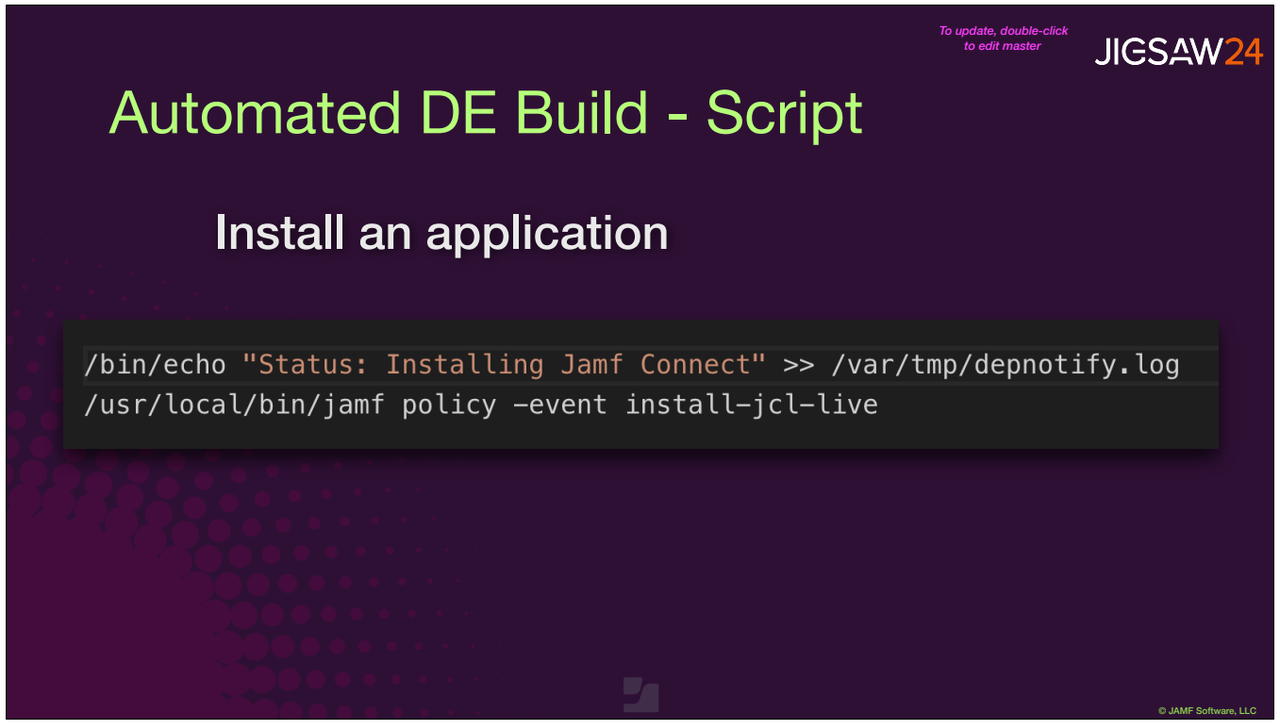
Automated DE Build - Script

Write to Notify screen

```
# Change DEPNotify title and text...
/bin/echo "Command: MainTitle: Setting things up..."  >> /var/tmp/depnotify.log
/bin/echo "Command: MainText: Please wait while we set this Mac up with the software and settings it
needs.\n We'll restart automatically when we're finished. \n \n Role: "$computerRole" Mac \n Serial
Number: "$serial" \n macOS Version: "$osversion""  >> /var/tmp/depnotify.log
```


Second snippet: Write out to the notify screen showing the build has started moving.
Standard **DEP notify** commands are used in the form of :
echo " Command: ……" These dynamically update the NoMAD notify screen.

Will Not be covering how to use the commands today



Third snippet: display progress on the notify screen and send a trigger to run a policy.
Could run one or many policies

Can be more informative by also changing main text and icons
Not doing that for this demo

JIGSAW24

Automated DE Build - Script

Reset Logon window

```
# Reset login window to Jamf Connect
log "Resetting Login Window..."
/usr/local/bin/authchanger -reset -OIDC
```


Forth snippet: Reset the logon window to remove NoMAD.
In this demo set to use JCL with OIDC for an Azure connection
If bound to AD use the **-reset** to use native Mac logon

JIGSAW24

Automated DE Build - Demo Time

Points: **build selection** – this is what holds the build script up
initial screen text setup in the enrolment package

following screen messages are driven by build script

A simple build, ideal for central building and shipping..

JIGSAW24

Part 2…. Non Device Enrollment Build

Teaching (your) grandmother to suck eggs is an English language saying that refers to a person giving advice to another person in a subject with which the other person is already familiar (and probably more so than the first person)

You all probably know most of that or similar workflows
Finished teaching how to suck eggs as we say in the UK

JIGSAW24

NON DE Build - Uses

What's this used for?

For devices **NOT** DEP capable but you to…

- Want a common OS level
- Erase with a common method
- Build with same build as the DEP devices

Non DEP build agenda

For your devices that aren't in ABM or ASM (They may be added later…We'll cover that!)
You want a common wipe and build method to the DEP capable devices

To update, double-click to edit master

JIGSAW24

NON DE Build - Overview

Stages:

1. Wipe and install clean macOS
2. Install extra workflow packages
3. Enroll in Jamf
4. Apply build

It’s OK, it’s mostly automated!

1. Wipe the old OS and put a clean one on- APFS 10.13.4 and up
   will use the macOS erase install switch and options
2. Install the extra packages required to make the build work
3. Enroll device in Jamf
4. perform the build as per the previous build

Don’t worry…. Mainly automated

JIGSAW24

Let’s get started

Back to basics…..

Information only…

Click on

JIGSAW24

NON DE Build - Your starter for 10

How do we get this process started?

Mac Deploy Stick

https://twocanoes.com/products/mac/mac-deploy-stick/

How do we get this started?
Mac deploy stick…
Use this to get the old OS wiped and
the clean OS installed with the extra bits needed
MDS will install the clean OS and the extra components required

To update, double-click to edit master

JIGSAW24

The Extras

A couple of packages…..

Information only…

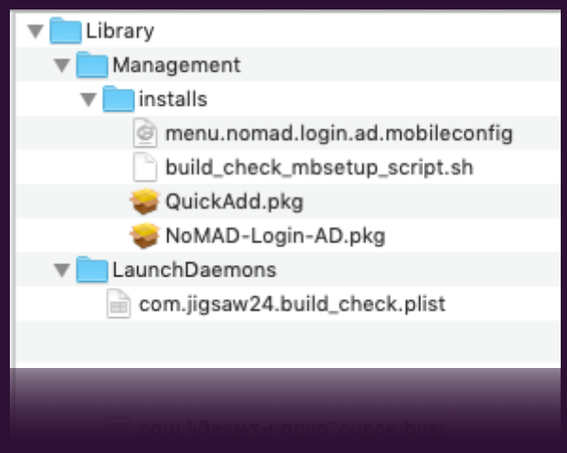
Click on

JIGSAW24

NON DE Build - Package #1

Components:

- NoMAD Login
- NoMAD settings profile
- QuickAdd.pkg
- Control script
- LaunchDaemon for script



To make this work we need a package.
1. This must contain NoMAD …… Same as DEP package
2. profile to configure NoMAD ….. Same as DEP package
3. QuickAdd package for your Jamf
4. The control script
5. launchdaemon to run the script

JIGSAW24

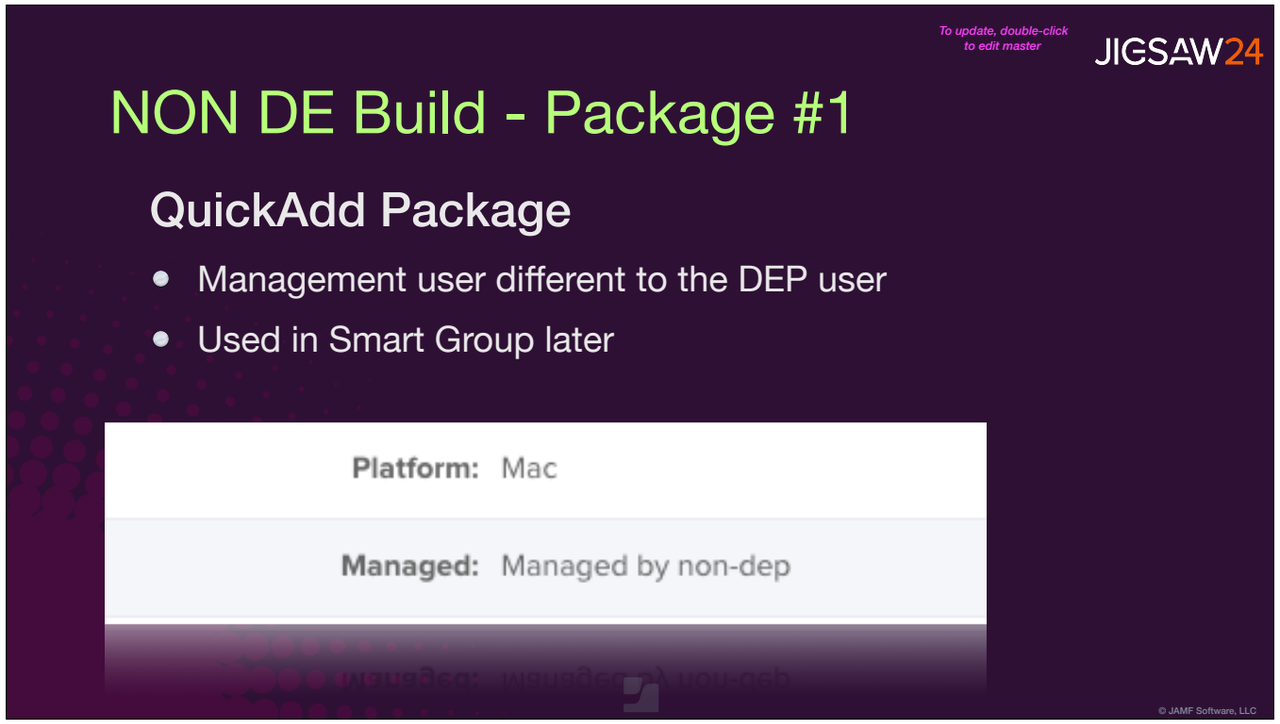
NON DE Build - Package #1

Postinstall script

- LaunchDaemon has to be loaded during package install
- Post install script does this

```
launchctl load /Library/LaunchDaemons/com.jigsaw24.build_check.plist
```


The package copies all the components into place.
This all happens **after** the OS is installed to the disk and before setup starts
There are no more restarts….
So to make the launchedaemon run it has to be loaded at install time



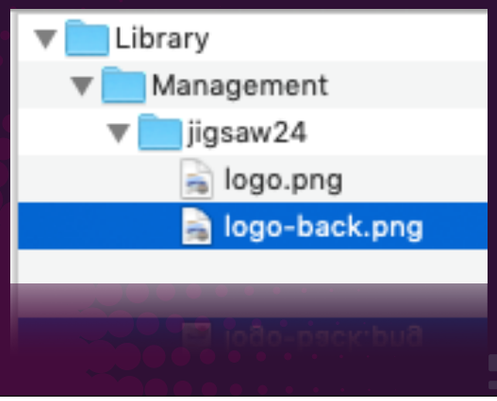
The main package contains a QuickAdd, made with Jamf recon..
1. Different Jamf management user name
2. used in a smart group later….
Allows another way to report on or scope builds later

JIGSAW24

NON DE Build - Package #2

Graphics

Common graphics package



Separate graphics package
Why not in main package?
Couldn’t get the graphics to work from the main package!

Worked fine in the DEP package

JIGSAW24

The Extras

The control script…..

Mentioned earlier the package has a control script
This is the heart of the process
Break the script down now

JIGSAW24

NON DE Build - Control Script

Check _mbsetupuser is active

```
# Wait for MBSetup User
mbSetupLoggedIn=0

until [ $mbSetupLoggedIn -gt 0 ]; do
    secho "Waiting for User"
    if [ "$loggedInUser" == "_mbsetupuser" ]; then
        mbSetupLoggedIn=1
    else
        sleep 1
        loggedInUser=$(/usr/bin/python -c 'from SystemConfiguration
    fi
done
```


First: Need to Wait for mbsetupuser to be active
Shows that Setup screens now active

Script can now move on….

JIGSAW24

NON DE Build - Control Script

Setup NoMAD

```
setupNoMAD () {
# Install NoMAD Profile
/usr/bin/profiles -I -F "$installersPath"menu.nomad.login.ad.mobileconfig

# Install NoLo
/usr/sbin/installer -pkg "$installersPath"NoMAD-Login-AD.pkg -target /

/usr/local/bin/authchanger -reset -preLogin NoMADLoginAD:UserInput NoMADLoginAD:Notify

/bin/echo "Command: MainTitle: Setting things up - NON DEP"  >> /var/tmp/depnotify.log
/bin/echo "Command: MainText: Starting mac build, progress screens will display shortly.\n \n This is
/bin/echo "Command: Image: "/Library/Management/jigsaw24/logo.png"" >> /var/tmp/depnotify.log
/bin/echo "Status: Please wait..." >> /var/tmp/depnotify.log

}
```


Need to setup nomad
1. Install the NoMAD profile
2. Install NoMAD
3. Run authchanger - make UserInput and notify set for prelogin
4. Set the initial text for notify screen

JIGSAW24

NON DE Build - Control Script

Check Jamf contactable

```
internetLive=0
until [ "$internetLive" == "200" ]; do

    internetLive=$(curl -s -k https://[redacted]jamfcloud.com/healthCheck.html --write-out %{http_code} -o /dev/null)
    sleep 1
done
```


Need Jamf to be available for the quickadd to work

Note:

If the device needs to connect to wifi then the standard setup screens
Will display
If already connected to network, ethernet or vm, they get bypassed

JIGSAW24

NON DE Build - Control Script

Switch Login Window to NoMAD Login

- Create .AppleSetupDone
- Kill loginwindow

```
# Create AppleSetupDone File
touch /var/db/.AppleSetupDone
# Restart loginwindow
killall loginwindow
# Enrol the device
startBuild
```


Creating applesetupdone means the setup won’t restart
Killing login window means NoMAD starts up

Then start the quickadd install…..

JIGSAW24

NON DE Build - Control Script

Install the QuickAdd package

- Enrolls device in Jamf
  - Sends enrollment trigger
  - Starts the build process

```
startBuild () {
/usr/sbin/installer -pkg "$installersPath"QuickAdd.pkg -target /
}
```


QuickAdd enrolls in Jamf
This sends the enrolment trigger to start the build process

JIGSAW24

User Logon post build

It’s built, now what happens?

User logs in using:

- Jamf connect
- NoMAD
- AD Bound

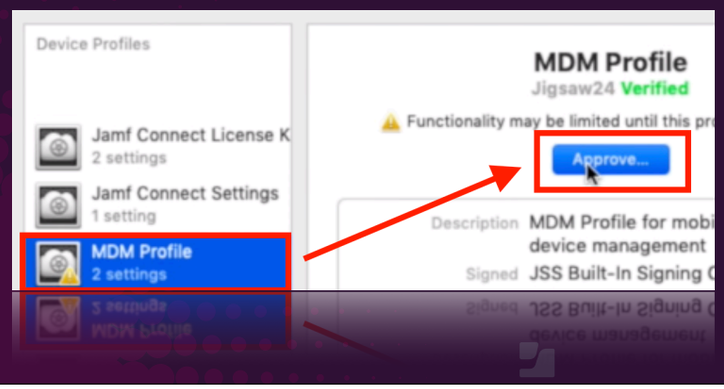
The build has finished
Users can login
This build style can used for Jamf Connect, NoMAD or AD binding

JIGSAW24

MDM Profile Approval

MDM Profile needs approving

Full MDM functionality not available without this



With NON DEP enrolment the MDM profile must be approved for full MDM functionality –
PPPC
KEXT

JIGSAW24

MDM Profile Approval

So how can we force approval?

Can’t force approval but…

- Can guide
- Repeatedly until they give up…

No way to force approval
But we can guide them and keep guiding if they ignore

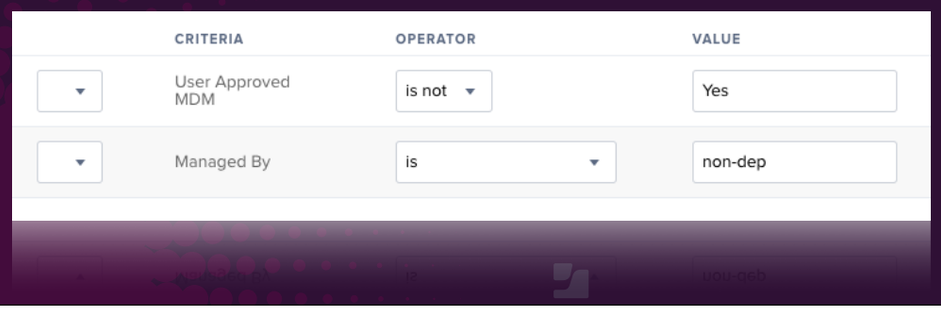
To update, double-click to edit master

JIGSAW24

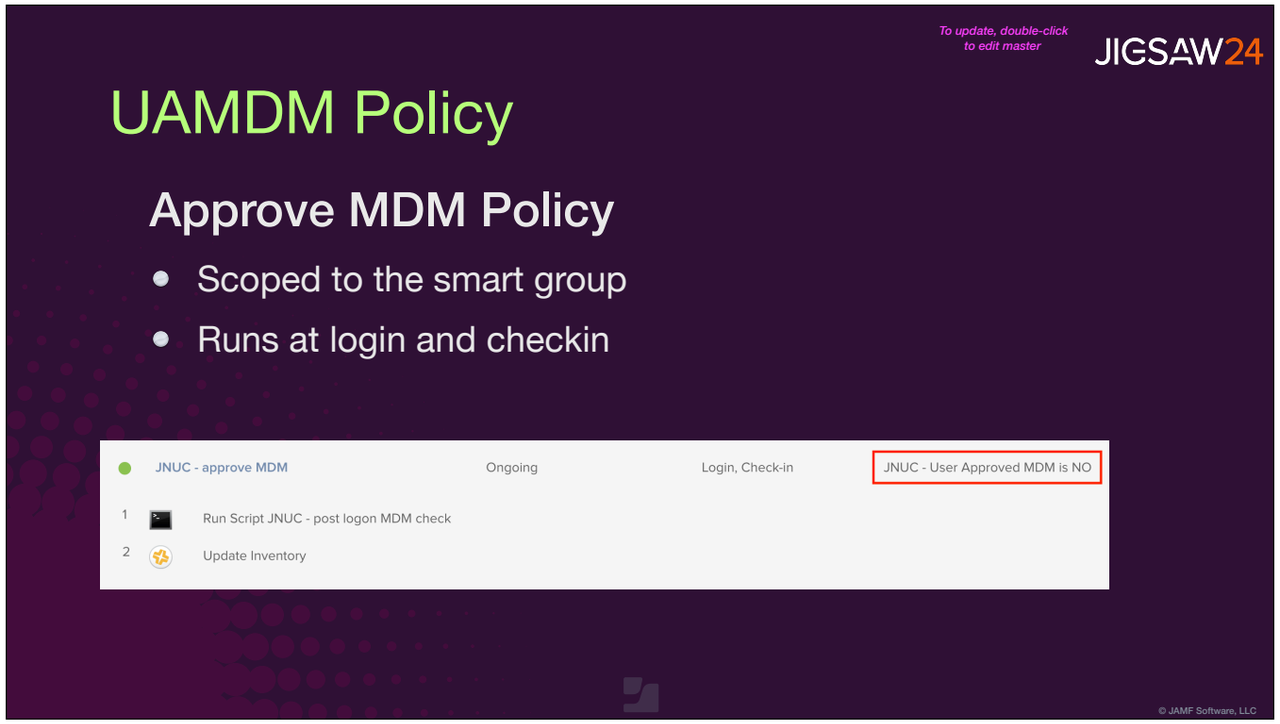
UAMDM Smart Group

Smart Group to scope approve MDM policy

- User Approved MDM not **Yes**
- Managed by **non-dep management user**



First need a smart group to control the approval policy
Picks up non user approved MDM for devices that have the non-dep management user
Used to scope the approval workflow policy after user login



Simple policy to run a script
Scoped to the smart group just mentioned
Runs at login and checkin
ensures it runs and keeps on running until approved

JIGSAW24

UAMDM Script

What does the script do?

- Checks if the MDM profile is approved
- Runs the user facing approval screens

This policy and script will keep running until the user approves
Always check if it is approved in case user has done it themselves

JIGSAW24

UAMDM Script

Check MDM Profile approved

```
isMDMEnrolled=$(profiles status -type enrollment | grep "MDM" |
awk -F":" '{ print $2}' | sed 's/ //' | grep -o Yes)

isMDMUserApproved=$(profiles status -type enrollment | grep "MDM" |
 awk -F":" '{ print $2}' | sed 's/ //' | grep -o "User Approved")
```

```
if [ "$isMDMEnrolled" = "Yes" ]; then
MDMStatus="enrolled"
    if [ "$isMDMUserApproved" = "User Approved" ]; then
    MDMStatus="approved"
    fi
else
    MDMStatus="none"
fi
```


Here use profiles command to find the User Approved status.
Then set MDMstatus to use in a case statement…

JIGSAW24

UAMDM Script

Run the approval process

- If approved - do nothing
- If not - Start user process

```
case $MDMStatus in
"approved" )
secho "Finished **********"
exit 0
;;
"enrolled" )
notifyUserToApprove
exit 0
;;
esac
```


If approved exit, inventory will take out of smart group
If not approved launch the user screens

JIGSAW24

UAMDM Script

The process

- Launch Jamf helper
  - Alternatively use DEP Notify
- Open Profiles pane
- Approve!

Good news, no more code shots!
Launch a notification screen – Jamf helper, DEP Notify etc, then launch self service or preferences directly.
I used self service for the demo but also show DEP notify as a bonus feature

JIGSAW24

Non DE Build - Finally its Demo time

OVERVIEW….

Points: **build selection** – this is what holds the build script up

initial screen message setup in the enrolment package

JIGSAW24

Part 3…… Rinse and Repeat

And now the good bit…

JIGSAW24

That was amazing…What now?

Now the devices are in Jamf

- Deploy policy driven erase and rebuild
  - Self Service or remote execution policy
- Conditionally deploy the NON DEP workflow packages
  - Automatically switch between Non DEP and DEP

Now devices in Jamf can deploy install erase mechanism
Deploy the Non DEP workflow packages
Remove the non DEP packages if device added to DEP later on

JIGSAW24

Policy Driven Device Reprovisioning

What's required?

- Graham Pugh's Erase-install script
  - https://github.com/grahampugh/erase-install
- A couple of Extension Attributes
- and of course smart groups and policies!

This uses Graham Pugh's excellent erase-install script.

Mark worked with Graham to include the functionality to find and install extra packages dynamically, so useful for more than just this workflow.

Created a couple of EA's… explained in a bit

JIGSAW24

Erase-Install script

What does it do?

- Downloads the appropriate OS install package
  - Can be locked to a specific version if required
- Will perform the erase and install
  - Displays user messaging when starting

It has other functionality but not talking about them today

Several download options can be set
In this demo locked to 10.14
Uses jamf helper for messaging when erasing
This script is being used to download the OS, maintain the downloaded OS and perform the erase.
Works like MDS but no recovery partition boot required.

JIGSAW24

Required Extension Attributes

Two extension attributes required….

Information only…

Click on

JIGSAW24

Extension Attribute #1

Installer valid EA

To record if OS installer is:

1. Downloaded
2. Valid version
   - Equal to or higher than installed OS version

This is enhancement to the erase install script
Use this, with smart groups, to make the download policy run.
Not downloaded… run
Downloaded but version lower than OS … run

Always maintain version parity even if OS upgraded
Not going to put that code in here ….. too big!

To update, double-click to edit master

JIGSAW24

Extension Attribute #2

Is the device DEP capable?

EA that records if device has DEP record in Apple

- Profiles show -type enrollment

```
# set default answer to "no". only want to set to "yes" if it's true
result="no"

# use profiles to see if the device is in Apple. if it is 1 is returned, if not 0
configURL=$(profiles show -type enrollment | grep ConfigurationURL | grep -c http)

if [ "$configURL" = "1" ]; then
result="yes"
fi

echo "<result>$result</result>"

```


For this to work need to know if the device is DEP ready, that is apple has a record for it pointing to an MDM server.

The EA records yes or no.
Allows scoping using smart groups…

JIGSAW24

Required Package

One package required….

Information only…

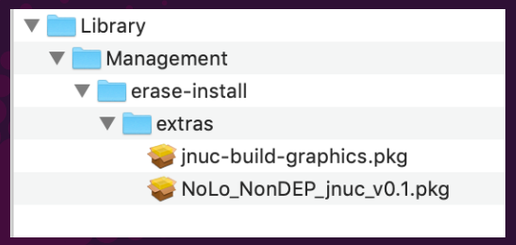
Click on

JIGSAW24

Package Required

Deploy Extras package

- Need to deploy the non DEP packages
  - Deployed as a single package



For the non DEP build need to deploy the extras used on the MDS build earlier Only deploying them, NOT installing them…
Bundle them into one package for deployment
The folder erase-install/extras is the default location for the script

To update, double-click to edit master

JIGSAW24

Required Smart Groups

Four Smart Groups required….

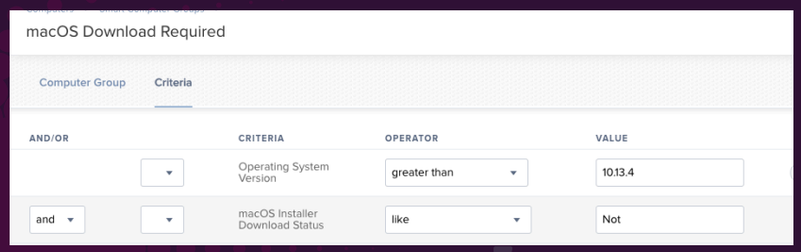
Information only…
Click on

JIGSAW24

Smart Group #1

Installer download required

- OS > 10.13.4
  - Add APFS requirement if HFS devices in estate
- Installer **not** downloaded or **not** valid



Is the installer downloaded?
OS greater than 10.13.4 for erase install to work
If you have devices that aren't APFS then scope these out
***** Group Common to both workflows ******

To update, double-click to edit master

JIGSAW24

Smart Group #2

Non DEP Extra packages required

- Have valid OS installer
- Not DEP capable

JNUC - NON DEP Extras required

Computer Group | Criteria

AND/OR		CRITERIA	OPERATOR	VALUE
	▾	JNUC - macOS installer valid	matches regex ▾	^[v]
and ▾	▾	JNUC - DEP Build Capable	is ▾	no

Need to know which devices are NON DEP and have the installer downloaded so the extra packages can be deployed

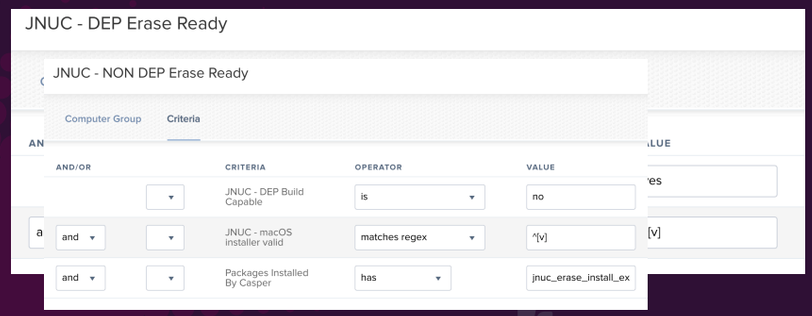
JIGSAW24

Smart Group #3

Erase Ready Groups

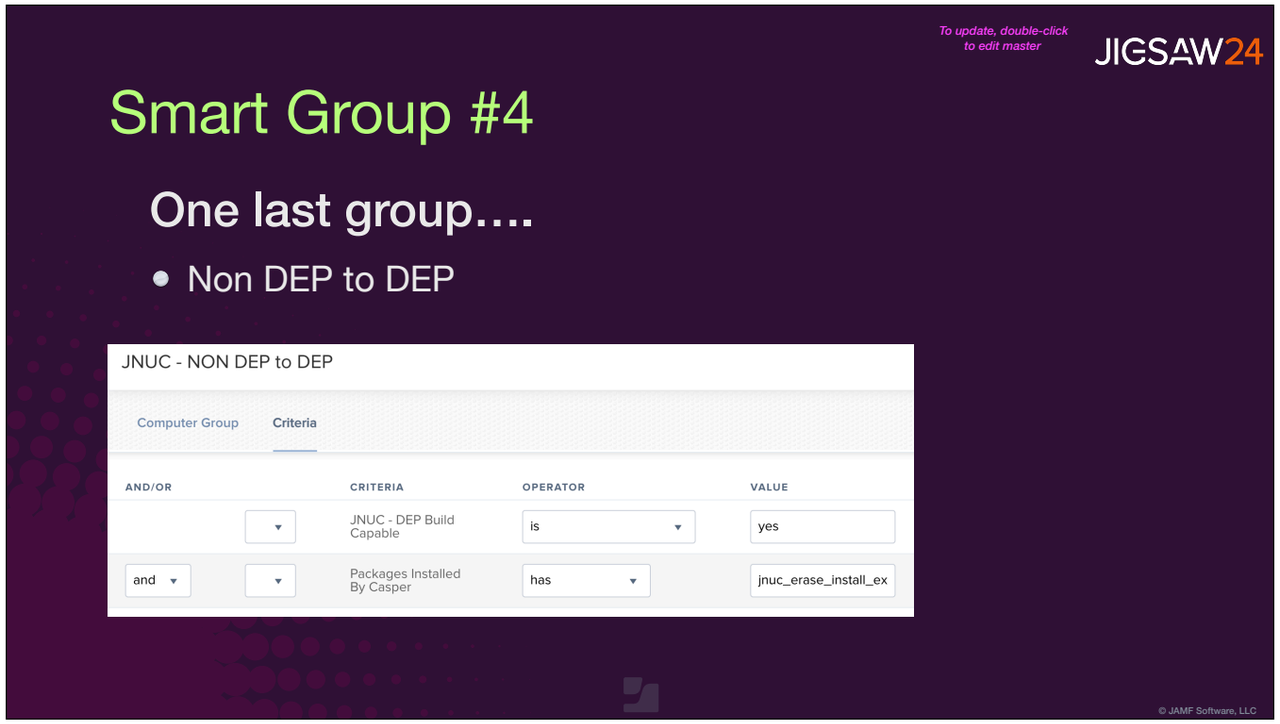
Two separate groups for erase ready

- One for DEP devices



Two groups for erase ready
DEP just needs a valid OS installer downloaded
Non DEP needs the OS installer and the extra packages

Makes them eligible for wipe mechanism



And finally for smart groups….
You get some of your non DEP devices added into DEP by the reseller
But they're already set up for the non DEP build. Need to find this and remove the extra packages.

Check for DEP ready is yes AND has the extras package deployed

To update, double-click to edit master

JIGSAW24

Required Policies

Four policies are required….

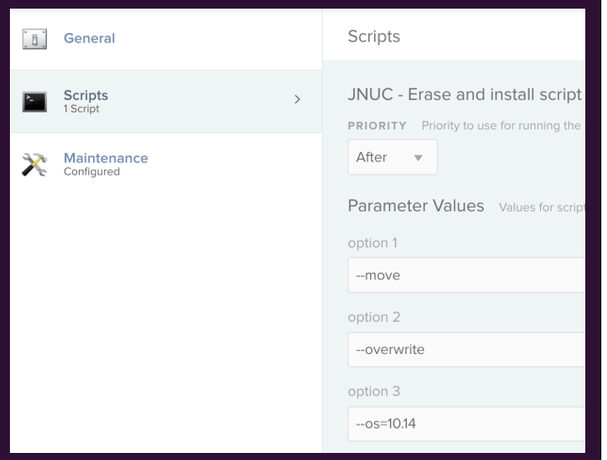
Information only…

JIGSAW24

Policy #1

OS Installer download

- Downloads the OS installer
  - Overwrites existing
- Scoped to download required
- Version options set
- Runs inventory when complete



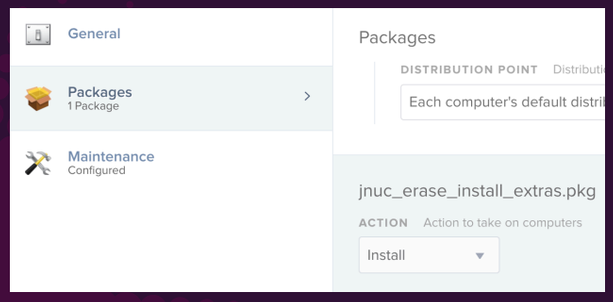
Core to process is download the OS installer
Set options required
Scoped to download required Smart Group keeps it updated if becomes invalid
This example overwrites any existing installer and locks version at 10.14
Inventory at completion

JIGSAW24

Policy #2

Deploy Extras package

- Deploy the extras packages
- Scoped to extras required smart group



For the non DEP build need to deploy the extras used on the MDS build earlier

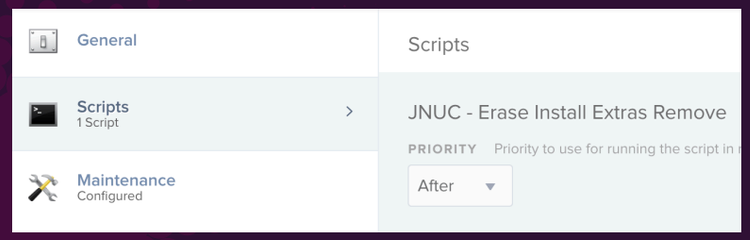
To update, double-click to edit master

JIGSAW24

Policy #3

Remove Extras package

- When devices added to DEP
- Delete the extras package
  - Build method will now be DEP



For when devices get added into DEP need to remove the extras packageSimple script …. Next slide

JIGSAW24

Policy #3

Remove Extras package

- Removes the packages
- Removes Jamf receipt

```
rm -Rf /Library/Management/erase-install/extras
rm -f "/Library/Application Support/JAMF/Receipts/jnuc_erase_install_extras.pkg"
```

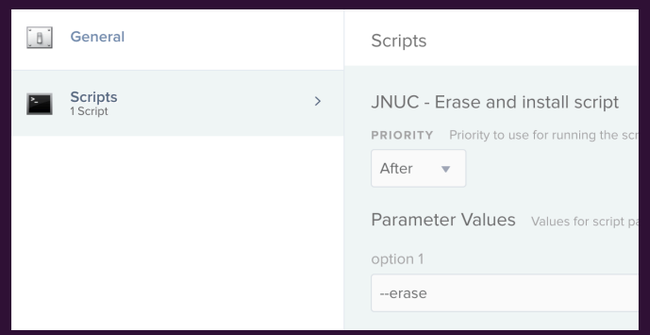

Ensures the device is standard DEP build
Simple script removes the package and the Jamf receipt

JIGSAW24

Policy #4

The big one!

- Erase and install
  - Can be Self Service
  - Can be force deployed
  - Can be event trigger
- Scoped to the Erase ready groups



The final policy, the big one is the erase policy
Runs the script with the —erase command
The script will check the extras folder and install the NON DEP extra packages if there.
Deploy through self service or scope to devices and run.
Put extra display / warnings / info round self service for better experience

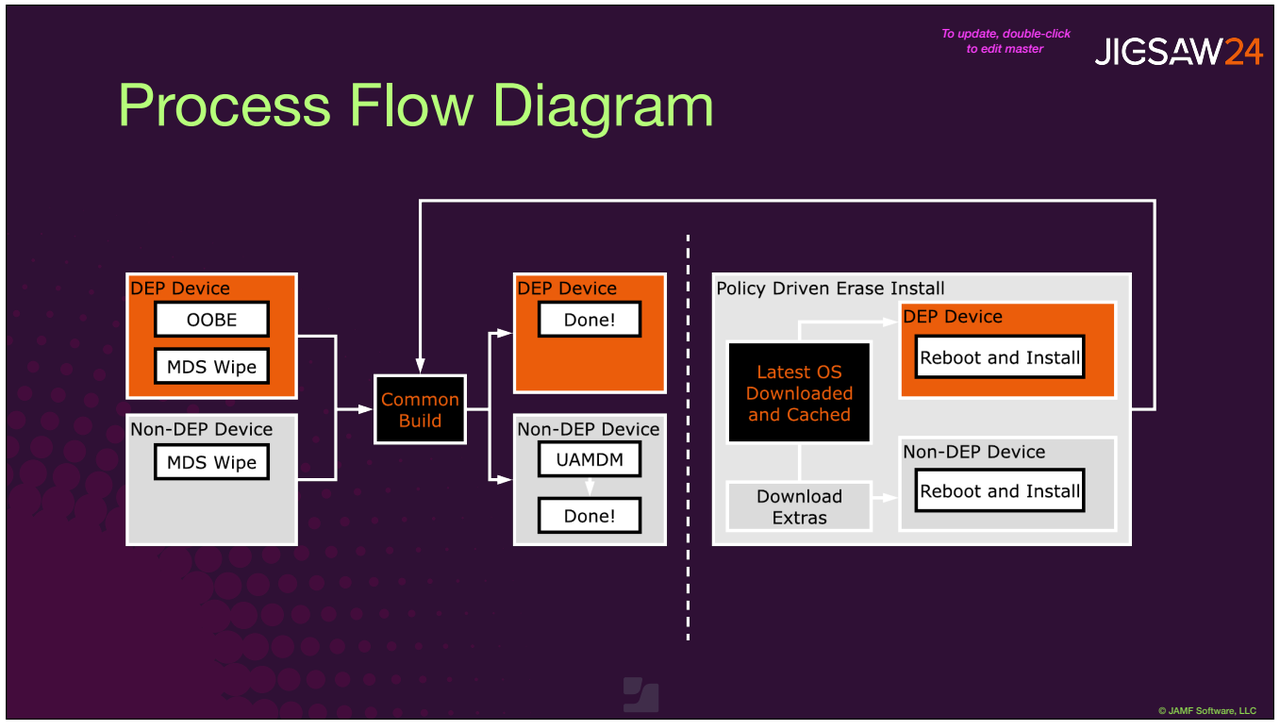
JIGSAW24

Wrap up

Nearly there ….

Information only…

Click on



And here’s how it all joins together
Start with new OOB or MDS wipe
Common build
Automatically download the installer and any extras
Wipe and go again
Don’t need to use the build sticks in normal operations once into this workflow

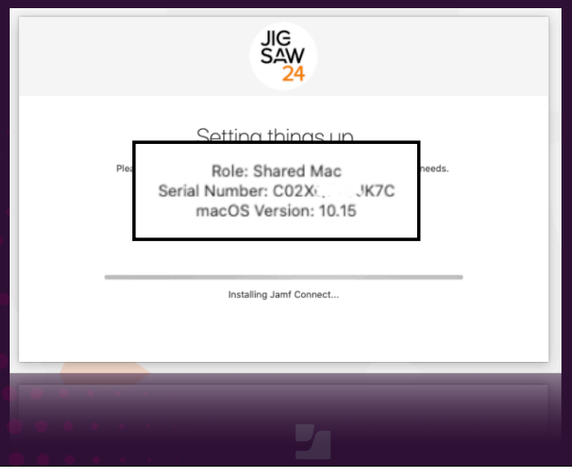
To update, double-click to edit master

JIGSAW24

Before you ask..

Does this work with 10.15?

YES!



Click x 2

Does this work with 10.15?
Tested it on release and it did.

JIGSAW24

And finally some links

1. Neil Martin's MacADUK 2019 Presentation
2. twocanoes.com mac-deploy-stick
3. Graham Pugh's erase-install
4. montysmacmusings blog