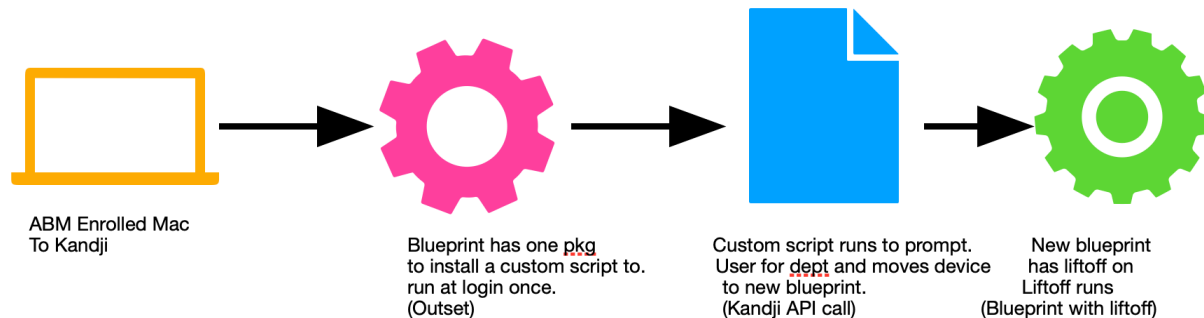


## Kandji and DEPNotify

This guide will walk you through how to achieve this workflow outlined below.

(Note Outset was not used instead this was all done via a custom PKG)



### Required Resources:

1. DEPNotify PKG ([From Here](#))
2. DEPNotify Start Script ([From Here](#))
3. Kandji Instance ([Get on here](#))
4. Kandji API account created ([See how to create here](#))
5. Blueprint IDs for each Dept. ([So how to find here](#))

Step 1. Edit the DEPNotify Script with your values. Note the BANNER\_IMAGE\_PATH would be where to plan to package the logo in your custom PKG.

```
# Banner image can be 600px wide by 100px high. Images will be scaled to fit
# If this variable is left blank, the generic image will appear. If using custom Self
# Service branding, please see the Customized Self Service Branding area below
BANNER_IMAGE_PATH="/Library/Resources/Logo.png"

# Update the variable below replacing "Organization" with the actual name of your organization. Example "ACME
ORG_NAME="Ericson Tech"

# Main heading that will be displayed under the image
# If this variable is left blank, the generic banner will appear
BANNER_TITLE="Welcome to $ORG_NAME"

# Update the variable below replacing "email helpdesk@company.com" with the actual plaintext instructions for
SUPPORT_CONTACT_DETAILS="email helpdesk@company.com"
```

Step 2. Remove all the example Polices in this Array

```
#####
# Policy Variable to Modify
#####
# The policy array must be formatted "Progress Bar text,customTrigger". These will be
# run in order as they appear below.
POLICY_ARRAY=(

)
#####
```

## Kandji and DEPNotify

### Step 3. Enable the Registration Screen (Set to true)

```
#####
# Registration Variables to Modify
#####
# Registration window configuration
  REGISTRATION_ENABLED=true # Set variable to true or false
```

Step 4. Disable all the other text input and drop down menu items by removing the text in the quotes. Leave one for the Popup for your departments.

Example:

REG\_TEXT\_LABEL\_1=""

REG\_TEXT\_LABEL\_2=""

REG\_POPUP\_LABEL\_1=""

```
# First Text Field
#####
# Text Field Label
  REG_TEXT_LABEL_1=""

# Place Holder Text
  REG_TEXT_LABEL_1_PLACEHOLDER="macBook0123"

# Optional flag for making the field an optional input for end user
  REG_TEXT_LABEL_1_OPTIONAL="false" # Set variable to true or false

# Help Bubble for Input. If title left blank, this will not appear
  REG_TEXT_LABEL_1_HELP_TITLE="Computer Name Field"
  REG_TEXT_LABEL_1_HELP_TEXT="This field is sets the name of your new Mac to what is in the Computer Name"

# Logic below was put in this section rather than in core code as folks may
# want to change what the field does. This is a function that gets called
# when needed later on. BE VERY CAREFUL IN CHANGING THE FUNCTION!
  REG_TEXT_LABEL_1_LOGIC (){
    REG_TEXT_LABEL_1_VALUE=$(/usr/bin/defaults read "$DEP_NOTIFY_USER_INPUT_PLIST" "$REG_TEXT_LABEL_1")
    if [ "$REG_TEXT_LABEL_1_OPTIONAL" = true ] && [ "$REG_TEXT_LABEL_1_VALUE" = "" ]; then
      echo "Status: $REG_TEXT_LABEL_1 was left empty. Skipping..." >> "$DEP_NOTIFY_LOG"
      echo "$(date +%a %h %d %H:%M:%S): $REG_TEXT_LABEL_1 was set to optional and was left empty. Skipping..." >> "$DEP_NOTIFY_LOG"
      sleep 5
    else
      echo "Status: $REGISTRATION_BEGIN_WORD $REG_TEXT_LABEL_1 $REGISTRATION_MIDDLE_WORD $REG_TEXT_LABEL_1"
      if [ "$TESTING_MODE" = true ]; then
        sleep 10
      else
        "$JAMF_BINARY" setComputerName -name "$REG_TEXT_LABEL_1_VALUE"
        sleep 5
      fi
    fi
  }

# Second Text Field
#####
# Text Field Label
  REG_TEXT_LABEL_2=""

# Place Holder Text
  REG_TEXT_LABEL_2_PLACEHOLDER="81926392"

# Optional flag for making the field an optional input for end user
  REG_TEXT_LABEL_2_OPTIONAL="true" # Set variable to true or false

# Help Bubble for Input. If title left blank, this will not appear
  REG_TEXT_LABEL_2_HELP_TITLE="Asset Tag Field"
  REG_TEXT_LABEL_2_HELP_TEXT="This field is used to give an updated asset tag to our asset management system"

# Logic below was put in this section rather than in core code as folks may
# want to change what the field does. This is a function that gets called
# when needed later on. BE VERY CAREFUL IN CHANGING THE FUNCTION!
  REG_TEXT_LABEL_2_LOGIC (){
    REG_TEXT_LABEL_2_VALUE=$(/usr/bin/defaults read "$DEP_NOTIFY_USER_INPUT_PLIST" "$REG_TEXT_LABEL_2")
    if [ "$REG_TEXT_LABEL_2_OPTIONAL" = true ] && [ "$REG_TEXT_LABEL_2_VALUE" = "" ]; then
      echo "Status: $REG_TEXT_LABEL_2 was left empty. Skipping..." >> "$DEP_NOTIFY_LOG"
      echo "$(date +%a %h %d %H:%M:%S): $REG_TEXT_LABEL_2 was set to optional and was left empty. Skipping..." >> "$DEP_NOTIFY_LOG"
      sleep 5
    else
      echo "Status: $REGISTRATION_BEGIN_WORD $REG_TEXT_LABEL_2 $REGISTRATION_MIDDLE_WORD $REG_TEXT_LABEL_2"
      if [ "$TESTING_MODE" = true ]; then
        sleep 10
      else
        "$JAMF_BINARY" recon -assetTag "$REG_TEXT_LABEL_2_VALUE"
      fi
    fi
  }
```

## Kandji and DEPNotify

Step 5. Add this logic to the popup #2:

**Note the BID is the Blueprint ID you want to move devices to.**

Example is dept. HR is selected then the device will be moved to that Blueprint.

**Note: Liftoff config will need to be enabled on the blueprints your plan to use this with.**

```
else

if [ $REG_POPUP_LABEL_2_VALUE = "HR" ]; then
echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"
1 BID="0a704d70-0dd5-4a9a-bff5-b785c76736af"
echo "Blueprint ID for YES $BID" >> "$DEP_NOTIFY_LOG"
fi

if [ $REG_POPUP_LABEL_2_VALUE = "IT" ]; then
echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"
2 BID="686e65ba-8dad-4f0c-a1a2-be3b8e06d043"
echo "Blueprint ID for NO $BID" >> "$DEP_NOTIFY_LOG"
fi

if [ $REG_POPUP_LABEL_2_VALUE = "ENG" ]; then
echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"
3 BID="686e65ba-9dad-4f3c-a1a2-be3b5e06d443"
echo "Blueprint ID for NO $BID" >> "$DEP_NOTIFY_LOG"
fi

if [ $REG_POPUP_LABEL_2_VALUE = "ARG" ]; then
echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"
4 BID="686e65ba-5dad-4f1c-a1a2-be3b8e06d449"
echo "Blueprint ID for NO $BID" >> "$DEP_NOTIFY_LOG"
fi

5 echo "Blueprint ID Final is $BID" >> "$DEP_NOTIFY_LOG"
ID=$(defaults read /Library/Preferences/io.kandji.Kandji.plist ComputerURL)
ID2=$(echo ${ID##*/})
echo "Computer URL is $ID and Device ID is $ID2" >> "$DEP_NOTIFY_LOG"
string="https://yourorg.clients.us-1.kandji.io/api/v1/devices/"
string="${string}${ID2}/"
echo "String is $string" >> "$DEP_NOTIFY_LOG"
echo "String fixed is $string" >> "$DEP_NOTIFY_LOG"
curl --location --request PATCH $string \
--header 'Content-Type: application/json' \
6 --header 'Authorization: Bearer 704b1b-428b-428b-428b-428b428b428b' \
--data-raw '{
  "blueprint_id": ""$BID""
}'

open -a /Library/Kandji/Kandji\ Agent.app/Contents/Resources/Helpers/Kandji\ Liftoff.app
```

1-4: Your Blueprint IDS.

5: Your Kandji url

6: Your Bearer token created in Kandji console

The Bearer token in the code below is an example not a real token

**Example CODE:**

**# Popup 2**

```
#####  
#####
```

```
# Label for the popup  
REG_POPUP_LABEL_2="Department"
```

```
# Array of options for the user to select  
REG_POPUP_LABEL_2_OPTIONS=(  
    "HR"  
    "IT"  
    "ENG"  
    "ARG"
```

```
)
```

```
# Help Bubble for Input. If title left blank, this will not appear  
REG_POPUP_LABEL_2_HELP_TITLE="Department"  
REG_POPUP_LABEL_2_HELP_TEXT="Pick your department you work in"
```

```
# Logic below was put in this section rather than in core code as folks may  
# want to change what the field does. This is a function that gets called  
# when needed later on. BE VERY CAREFUL IN CHANGING THE FUNCTION!
```

```
REG_POPUP_LABEL_2_LOGIC (){  
    REG_POPUP_LABEL_2_VALUE=$(/usr/bin/defaults read  
"$DEP_NOTIFY_USER_INPUT_PLIST" "$REG_POPUP_LABEL_2")  
    echo "Status: $REGISTRATION_BEGIN_WORD $REG_POPUP_LABEL_2  
$REGISTRATION_MIDDLE_WORD $REG_POPUP_LABEL_2_VALUE" >>  
"$DEP_NOTIFY_LOG"  
    if [ "$TESTING_MODE" = true ]; then  
        sleep 10  
    else
```

```
        if [ $REG_POPUP_LABEL_2_VALUE = "HR" ]; then  
            echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"  
            BID="0a704d70-0dd5-4a9a-bff5-b785c76736af"  
            echo "Blueprint ID for YES $BID" >> "$DEP_NOTIFY_LOG"  
        fi
```

```
        if [ $REG_POPUP_LABEL_2_VALUE = "IT" ]; then  
            echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"  
            BID="686e65ba-8dad-4f0c-a1a2-be3b8e06d043"  
            echo "Blueprint ID for NO $BID" >> "$DEP_NOTIFY_LOG"
```

## Kandji and DEPNotify

```
fi

if [ $REG_POPUP_LABEL_2_VALUE = "ENG" ]; then
    echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"
    BID="686e65ba-9dad-4f3c-a1a2-be3b5e06d443"
    echo "Blueprint ID for NO $BID" >> "$DEP_NOTIFY_LOG"
fi

if [ $REG_POPUP_LABEL_2_VALUE = "ARG" ]; then
    echo $REG_POPUP_LABEL_2_VALUE >> "$DEP_NOTIFY_LOG"
    BID="686e65ba-5dad-4f1c-a1a2-be3b8e06d449"
    echo "Blueprint ID for NO $BID" >> "$DEP_NOTIFY_LOG"
fi

echo "Blueprint ID Final is $BID" >> "$DEP_NOTIFY_LOG"
ID=$(defaults read /Library/Preferences/io.kandji.Kandji.plist
ComputerURL)
ID2=$(echo ${ID##*/})
echo "Computer URL is $ID and Device ID is $ID2" >>
"$DEP_NOTIFY_LOG"
string="https://yourorg.clients.us-1.kandji.io/api/v1/devices/"
string="${string}${ID2}/"
echo "String is $string" >> "$DEP_NOTIFY_LOG"
echo "String fixed is $string" >> "$DEP_NOTIFY_LOG"
curl --location --request PATCH $string \
--header 'Content-Type: application/json' \
--header 'Authorization: Bearer 9999a9a9-03e9-4106-b31a-8a992a4e988d' \
--data-raw '{
    "blueprint_id": ""$BID""
}'
open -a /Library/Kandji/Kandji\ Agent.app/Contents/Resources/Helpers/
Kandji\ Liftoff.app
sleep 5
killall DEPNotify

fi

}
```

Step 6. Now that your script is ready you can package up and add this to an intake blueprint that all devices will first enroll to.

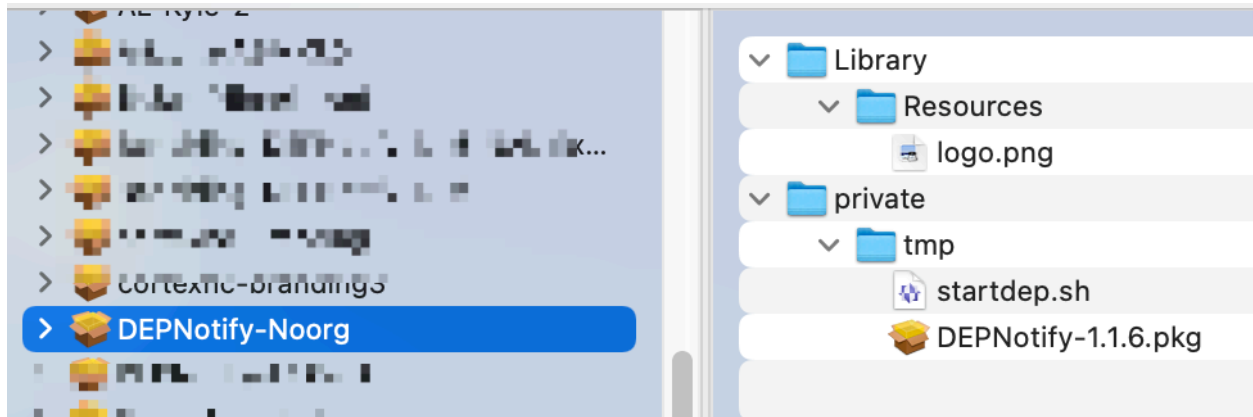
**Note: This Blueprint will not have liftoff enabled in it**

Using Composer or some other packaging tool to add these files to your custom PKG>

Step 7. Add this Custom Package to your new blueprint for all devices so that all devices enrolled with get this.

**See a video of this in action [here](#)**

## Kandji and DEPNotify



## Done.

If you have any questions reach out:

Email:

[kyle@ericson-tech.com](mailto:kyle@ericson-tech.com)

Slack Username:

Kyle Ericson

LinkedIn:

<https://www.linkedin.com/in/kylejericson>

If you would like me to set this up in your Kandji environment let me know we can get a contract going.