

Managing iPads from the Command Line & Scripting it too...



Jesse C. Smillie
Gateway School District

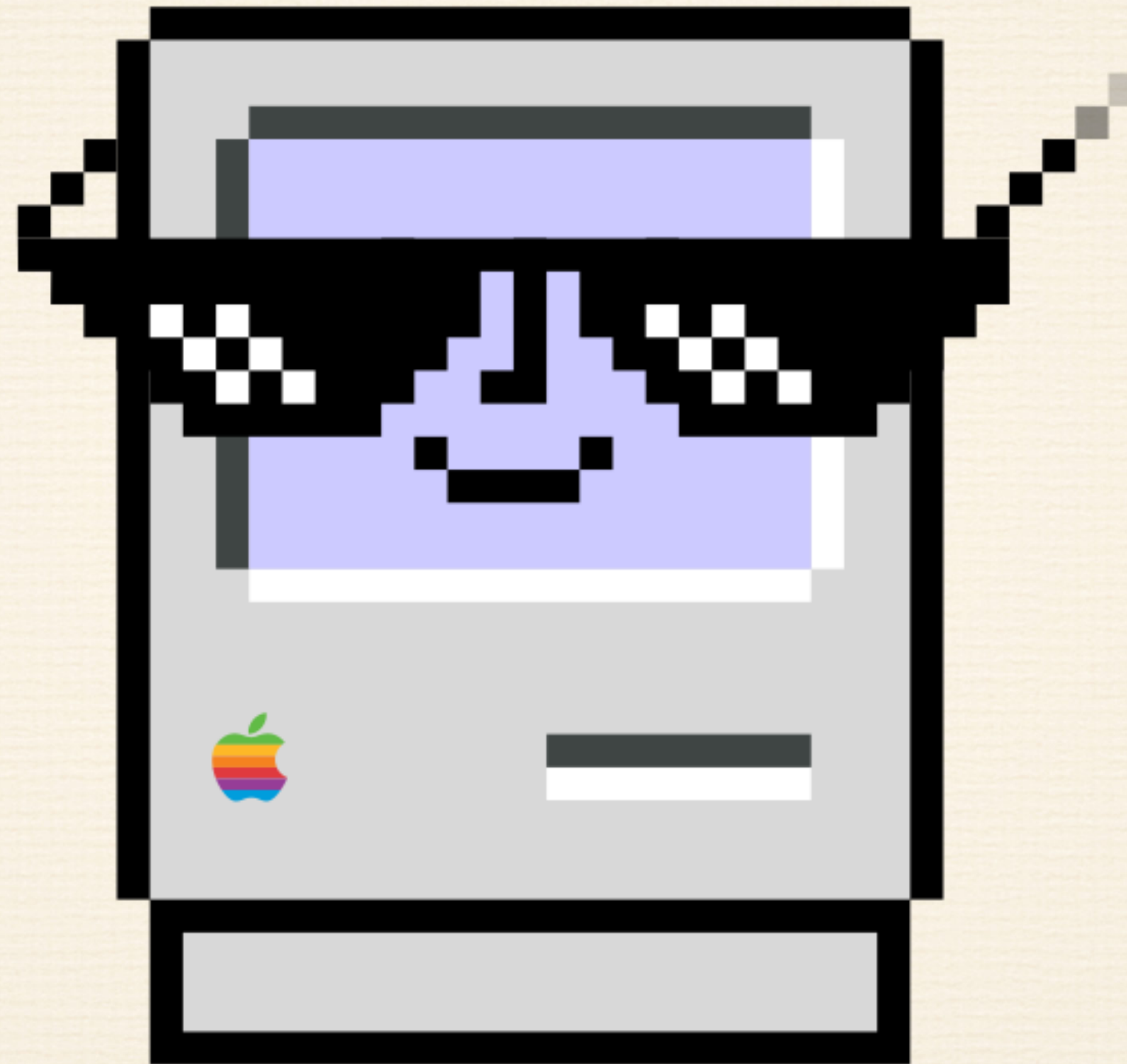


This Presentation and any Scripts mentioned will be available on GitHub:

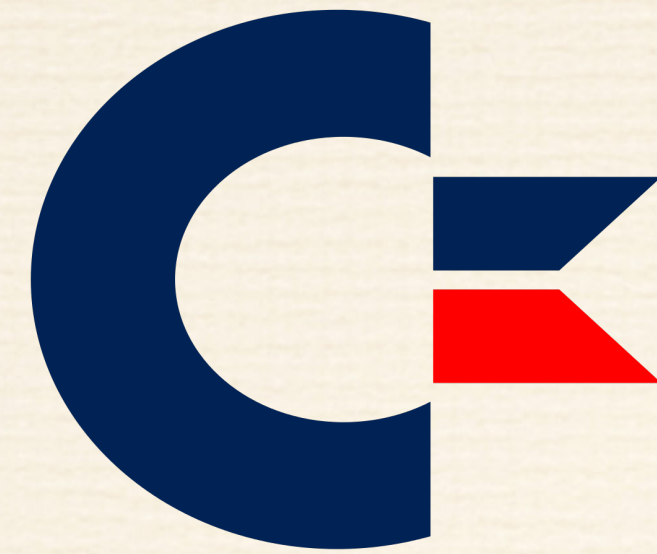


bit.ly/JesseSmiPSUMacPrez

Before I start lets get an idea of what you all have going on.

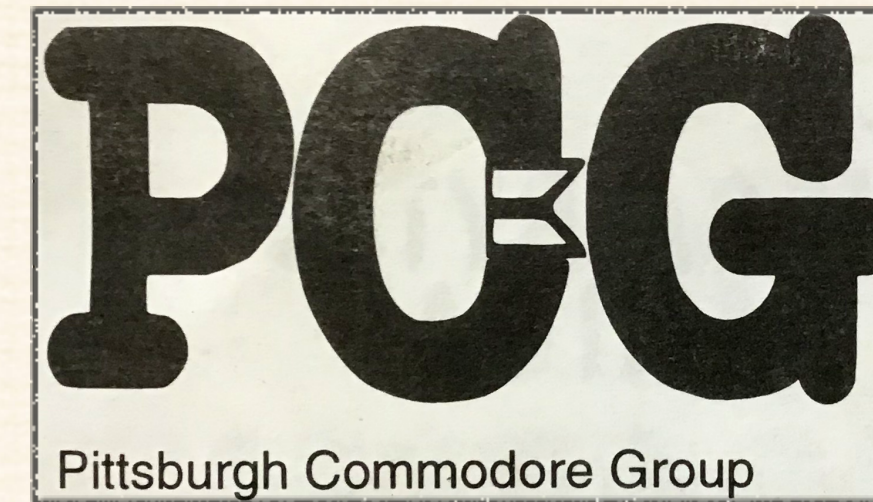


Me



My Love for Computers
starts here

Got Encouraged Here →



What was a Hobby
turns into work in 99'

And GSD introduced
me to Apple on work day 1



Jesse C. Smillie

Mac & Linux Operations Engineer

- With GSD since 1999
- ACMT since 2005
- Solo Apple Tech until Sept 2021
- Presenter at PSU MacAdmins 2019
- “Ask Apple: Mosyle Spotlight” Presenter April 2021
- Co-Presenter for March 22’ Apple IT Roundtable @ GSD
- Co-Presenter for April 22’ Apple Virtual Tech Event
“Onboarding iPads to MDM Using Apple Configurator”
- Co-Presenter at PSU MacAdmins 2022 (The Campfire years)



*Drinker of the Apple Kool Aide and
future Mosyle Cult member*

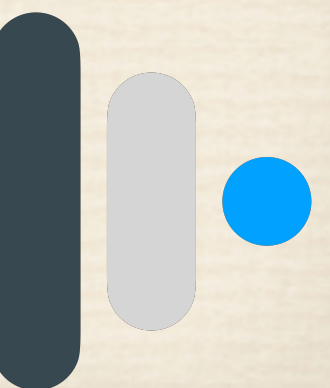
About Gateway

- ❖ Located in Monroeville, Pennsylvania (just outside Pittsburgh)
- ❖ 3,200 across 4 elementary schools, 1 middle school, and 1 high school
- ❖ 600 Staff (over half issued Laptops)
- ❖ MacBook Air (2020) for all teachers: Mac OS (100%)
- ❖ iPad 8th gen or better for Teachers who want them

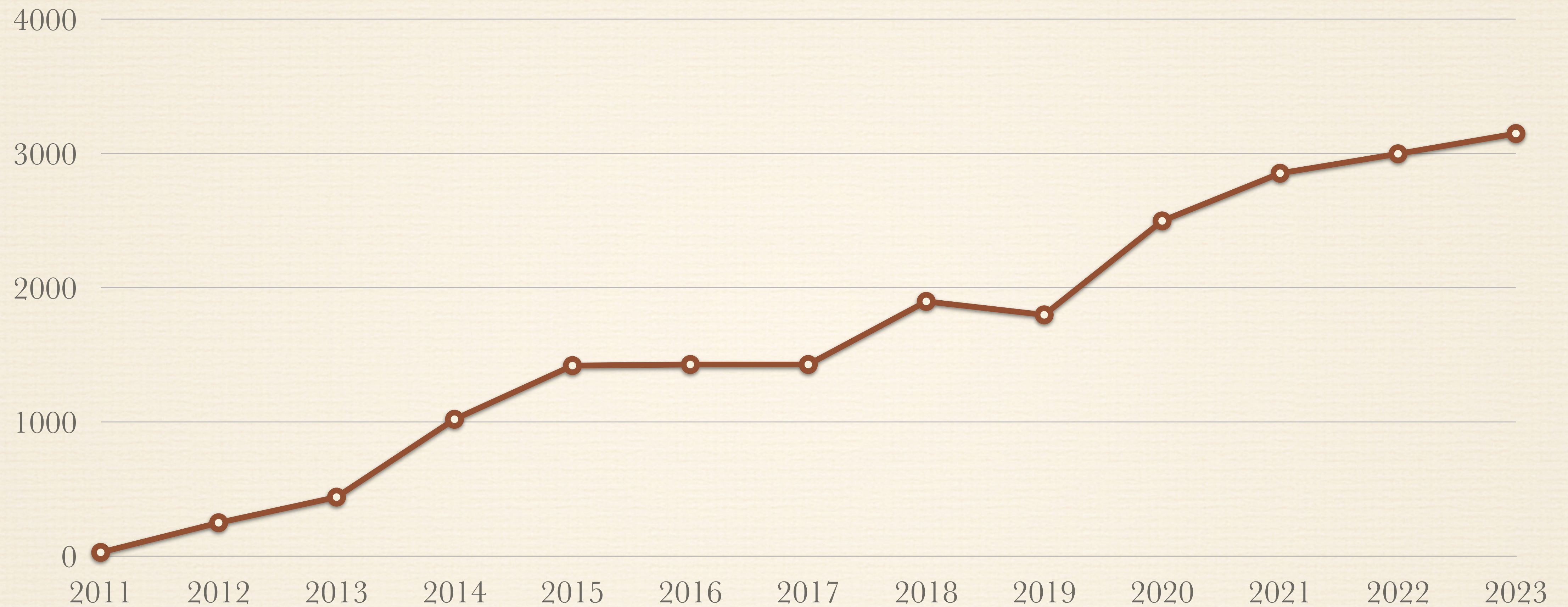


Technology Deployed at a Glance

- ❖ 4 Technicians and 1 Director
 - ❖ Middle School Student Led Help Desk
 - ❖ High School Student Led Help Desk
- ❖ Elementary (K-4) 1500 iPads 1:1
- ❖ Moss Side Middle (5-8) 1200 iPads 1:1
- ❖ Gateway High School: About 1,200 1:1 Chromebooks and a cart of Shared Mode iPads (30)
- ❖ District Wide Staff 250 iPads in hand
- ❖ All Apple hardware enrolled in Mosyle OneK12 April 2020 to current.



iPad Growth in the District

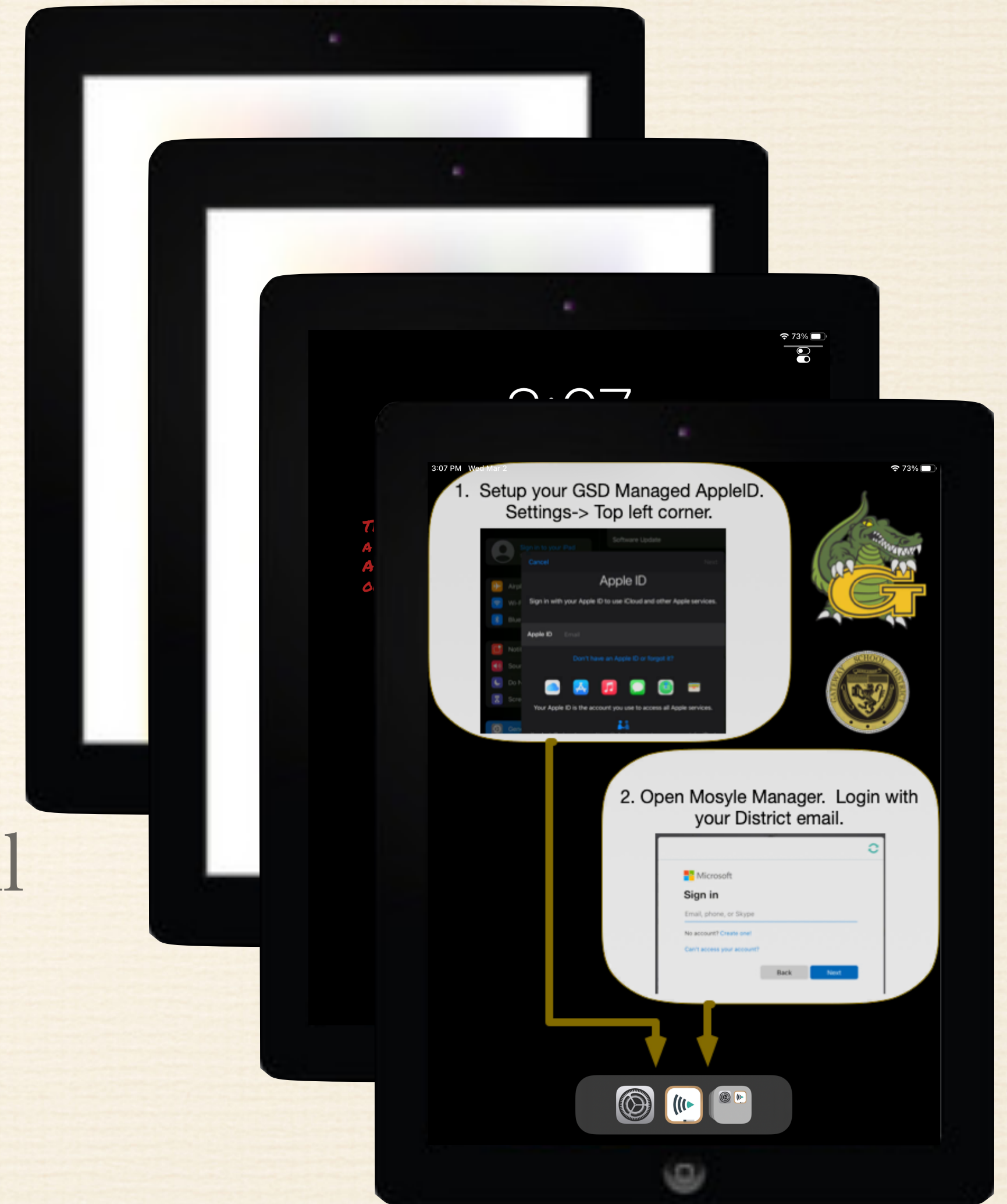


Things to Understand

- ❖ We are a Mosyle Shop. Everything I show you works with Mosyle OneK12 for sure. Most of this is universal though and will note as we go along.
- ❖ What isn't universal can easily be adapted to work with your MDM, ticket system, inventory, etc you just have to figure out what their API calls are.

In GSD All 1:1 Roads start with Limbo

- ❖ After Hello screens iPad goes right to Limbo.
- ❖ Apps deemed as must have for all are tied to Limbo. This way those apps are there from the beginning.
- ❖ Limbo iPads are only “Allowed” Manager and Settings. This forces setup. Login to Manager and iPad is now usable. Apps for all appear. Additional apps by grade install.
- ❖ Custom Wallpaper tells you to login.



Why do it this way?

- ❖ Speeds up iPad deployment (from my hands to their hands)
- ❖ Limbo iPads are inventoried, on charge, and automatically updated. Always ready to be handed out. In charging/storage carts until then.
- ❖ Avoids big wifi draw during hand outs. Post login by student/staff they are ready to use. Apps are there.
- ❖ Bad certificates, bad time on iPad, MDM server “hiccups” play less of a role as they don’t happen when the student/staff is standing here waiting to pickup.

iPad Storage Carts





CFGUTIL

Think Apple Configurator2 but from the Command Line

CFGUTIL

<https://support.apple.com/guide/apple-configurator-2/use-the-command-line-tool-cad856a8ea58/mac>

- ❖ Installable from Apple Configurator 2 (Automation Tools)
- ❖ `/usr/local/bin/cfgutil`
- ❖ Can work on devices which have TRUST between the Mac and iPad
- ❖ Scriptable `/usr/local/bin/cfgutil exec -a <YourScriptHere>`

My Sweet Terminal>_ My Personal Cell Phone

```
cfgutil list
```

```
Type: iPhone13,2 ECID: 0x11XXXXXA81001E UDID: 00008101-00114849XXXXXXXXXE  
Location: 0x1100000 Name: DaDarkCarnival
```

```
cfgutil --ecid 0x11XXXXXA81001E get buildVersion
```

20F75

iOS 16.5 Build Version

```
cfgutil --ecid 0x11XXXXXA81001E get firmwareVersion
```

16.5

**NOTE: PAIRING AND/OR TRUST ARE NOT REQUIRED TO
GET THIS INFO**

So what can we do with this?

My Sweet Terminal >_Some random iPad in My Office

**MYECID REPRESENTS
THE ECID OF THE IPAD**

```
cfgutil --ecid MyECID restore
```

Restores latest compatible iPadOS

```
cfgutil --ecid MyECID install-profile /Users/Shared/MyWifiProfile.mobileconfig
```

*Installs a wifi profile with SSID, Username, & Password.
Should be a dedicated setup network. I use "Apple Store"*

```
cfgutil --ecid MyECID prepare --dep --skip-language --skip-region
```

*Tell the iPad to do DEP process, skip-language & Region.
Other "Hello" prompts will be skipped too per your enrollment
config in your MDM*

NOTE: PAIRING + TRUST ARE REQUIRED HERE UNLESS IN DFU MODE

Why not Tether?

- ❖ If your only doing a few iPads USB tethering is probably fine and you can skip the Wifi profile business...
- ❖ But what I'm going to show you in a few slides here was designed to do masses of iPads. USB connection to start the process and wifi to finish App downloads.



Biggest Pitfall of Configurator/CFGUTIL

Is Supervision Identity Trust

- ❖ If iPad and your configurator Mac have trust for same Supervision Identity you can easily do all sorts of things to an iPad like erase, update, etc.
- ❖ Without Supervision Identity you can't pair and can't do anything to the iPad (restore, update, etc.)
- ❖ Apple Configurator **MUST BE OPEN** for CFGUTIL to access KeyChain stores!!!!

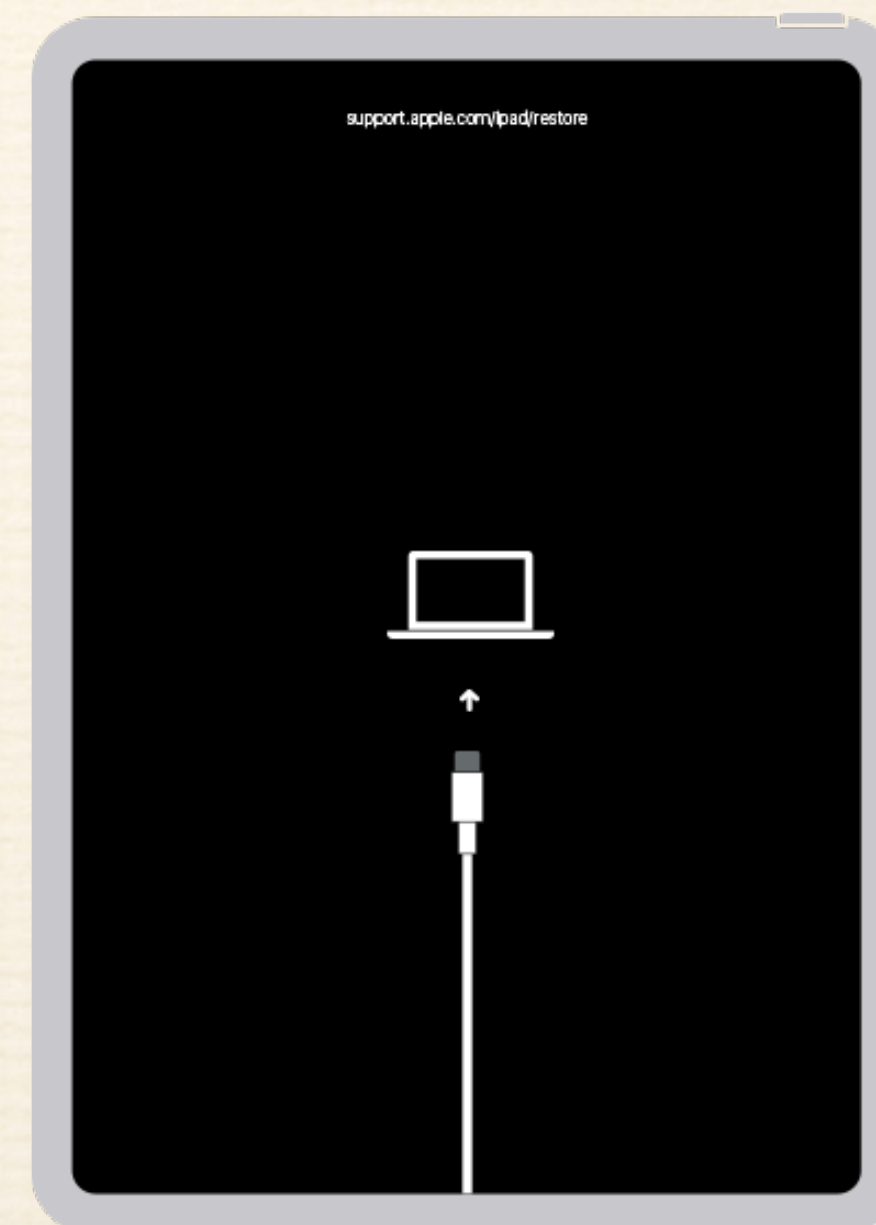
Ways around Trust Issues...

- ❖ Leverage MDM by using the ERASE command.

If iPad has checked in to MDM today
and its on wifi now easy way to go!

- ❖ Boot iPad to DFU mode. DFU mode doesn't care about Trust.

Tedious on the thumbs but
maybe your last option.



NOTE: cfgutil man page says about exec:

Run a custom command each time a device attaches or detaches.

There is no filtering on device attach or detach events; for example, restoring a device will cause it to detach and attach several times in a row. Complex "on attach" scripts should be written to exit if a previous attach script is still running for the device.

Script execution is not serialized. A detach script may start as an attach is still running and vice versa.

The `--ecid` and `--foreach` global options aren't respected by this command.

When the attach/detach scripts are running, these environment variables are set:

ECID	Target device's ECID.
PATH	The path is changed to include <code>cfgutil</code> .
UDID	Target device's UDID, if available.
deviceName	Target device's name, if available. ("iPhone 1")
deviceType	Target device's type, if available. ("iPhone7,2")
buildVersion	Installed iOS build number, if available. ("12B466")
firmwareVersion	Installed iOS version, if available. ("8.1.3")
locationID	Location ID of the device's USB port. ("0x00000001")

/usr/local/bin/cfgutil exec -a <my script>

- ❖ Executing this command will run your script every time an iPad is connected.
- ❖ Variables given to the script executed:

- ECID
 - UDID
 - PATH
 - deviceName
 - DeviceType
 - buildVersion
 - firmwareVersion
 - locationID
- Really Helpful Stuff**
-
- The diagram consists of four red arrows originating from the text 'Really Helpful Stuff' on the right. Two arrows point horizontally to the left towards the variables 'ECID' and 'UDID'. Two other arrows point diagonally down and to the left towards the variables 'buildVersion' and 'firmwareVersion'. The variables 'ECID', 'UDID', 'buildVersion', and 'firmwareVersion' are each enclosed in a red oval.

/usr/local/bin/cfgutil exec -a <my script>

```
# !/bin/zsh
#
# Quick Example of CFGUTil script. This would be launched from
# the command line as so-> /usr/local/bin/cfgutil exec -a /Users/Shared/blah/<thisscript>.sh
#
# NOTE THIS SCRIPT WILL KEEP RUNNING OVER AND OVER AGAIN UNTIL YOU PRESS CONTROL X.. This is because
# the starter line you used is effectively telling cfgutil to go into loop mode and execute the script
# everytime an iPad is connected.

echo "The iPad connected has:"
echo "-----"
echo "Type of iPad-> $deviceType"
echo "ECID-> $ECID"
echo "UDUD-> $UDID"
echo "Location-> $locationID <--This is the location of the device according to the USB tree."
echo "Name-> $deviceName <--Name of the device currently."
echo "Build Version of iPadOS-> $buildVersion"
echo "Firmware Version of iPad-> $firmwareVersion"

echo "All of the info in the variables is given weather you have trust or not..."

exit 0
```

In the GitHub—> `cfgutil_exec_Simple_output_Example.sh`


```
→ PSUMacAdmins2023_ScriptingConfigUtil git:(main) x  
→ PSUMacAdmins2023_ScriptingConfigUtil git:(main) x  
→ PSUMacAdmins2023_ScriptingConfigUtil git:(main) x
```

THIS VIDEO CAN BE FOUND LINKED IN THE GITHUB REPO



/usr/local/bin/cfgutil exec -a <my script>

Simple “clean out & Setup”

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

```
TEMPORARYWIFIPROFILE="/Users/Shared/ShakeNBake/Profiles/AppleStoreWifi7Days.mobileconfig"
```

```
#Restore device which is connected.
```

```
#####NOTE THIS WILL ONLY WORK ON A DEVICE YOUR MACHINE HAS TRUST WITH
```

```
bash -c "/usr/local/bin/cfgutil --ecid "$ECID" restore" 2>/dev/null
```

```
#Install Wifi Profile
```

```
#####NOTE THIS WILL ALSO ACTIVATE THE DEVICE
```

```
bash -c "/usr/local/bin/cfgutil --ecid "$ECID" install-profile $TEMPORARYWIFIPROFILE" 2>/dev/null
```

```
#Call up Apple and get the relay to your MDM. MDM takes over.
```

```
bash -c "/usr/local/bin/cfgutil --ecid "$ECID" prepare --dep --skip-language --skip-region" 2>/dev/null
```

```
#just making sure we have a clean break at the end of this script.
```

```
exit 0
```

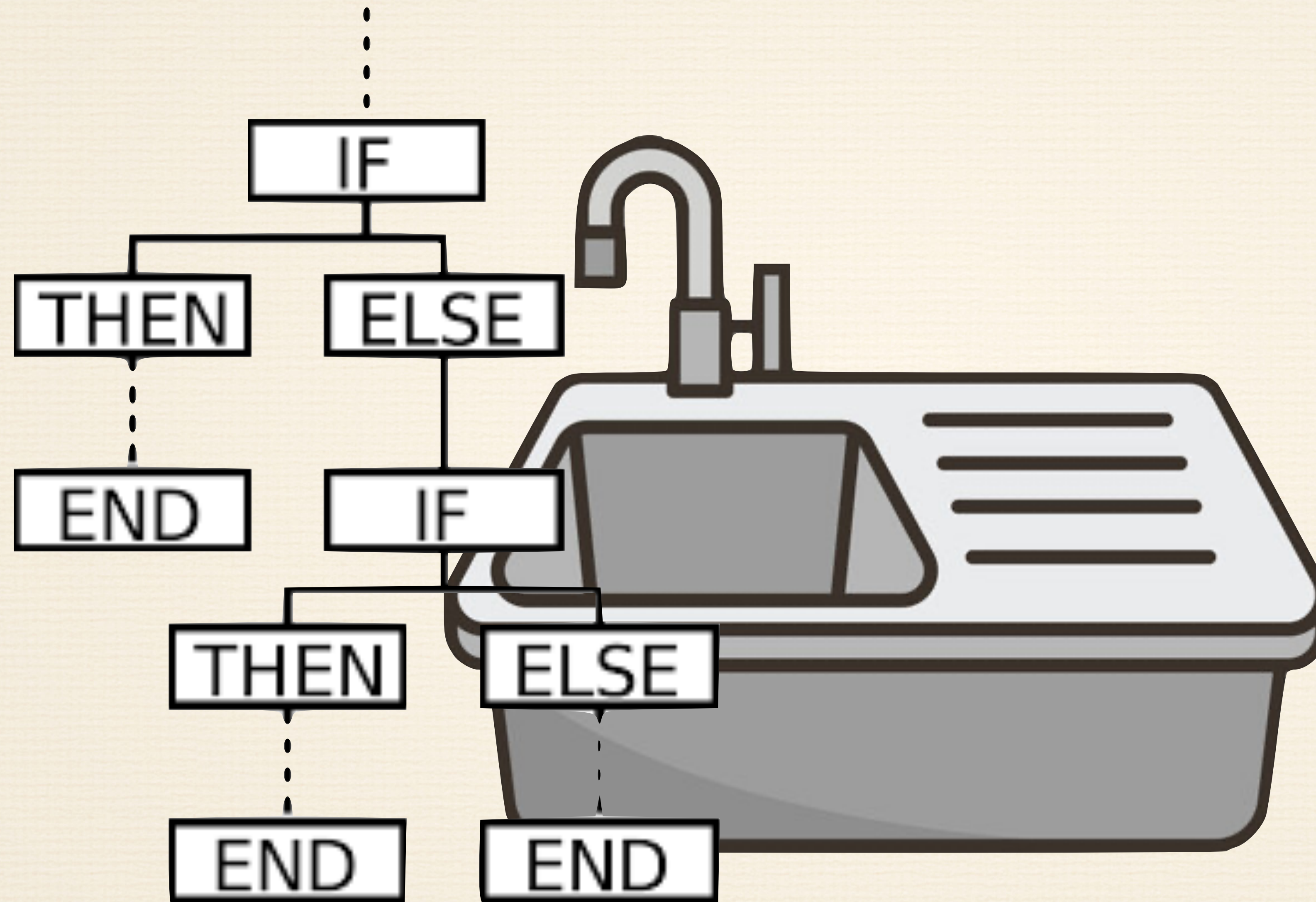
In the GitHub—> `cfgutil_exec_BasicWipe2Setup_Example.sh`

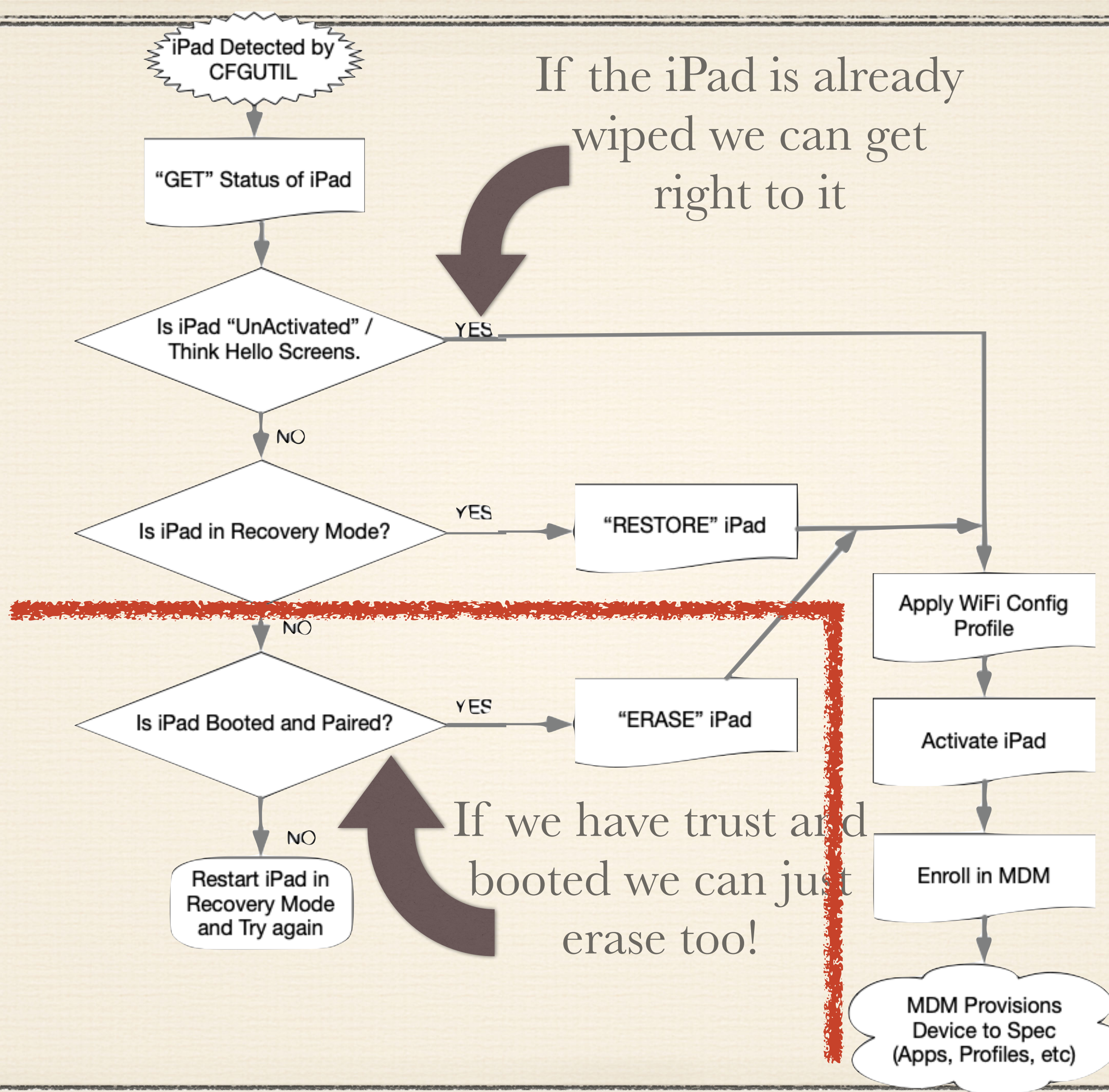
Running at 8x Speed

THIS VIDEO CAN BE FOUND LINKED IN THE GITHUB REPO



What else do we need?





→ PSUMacAdmins2023_ScriptingConfigUtil git:(main) x

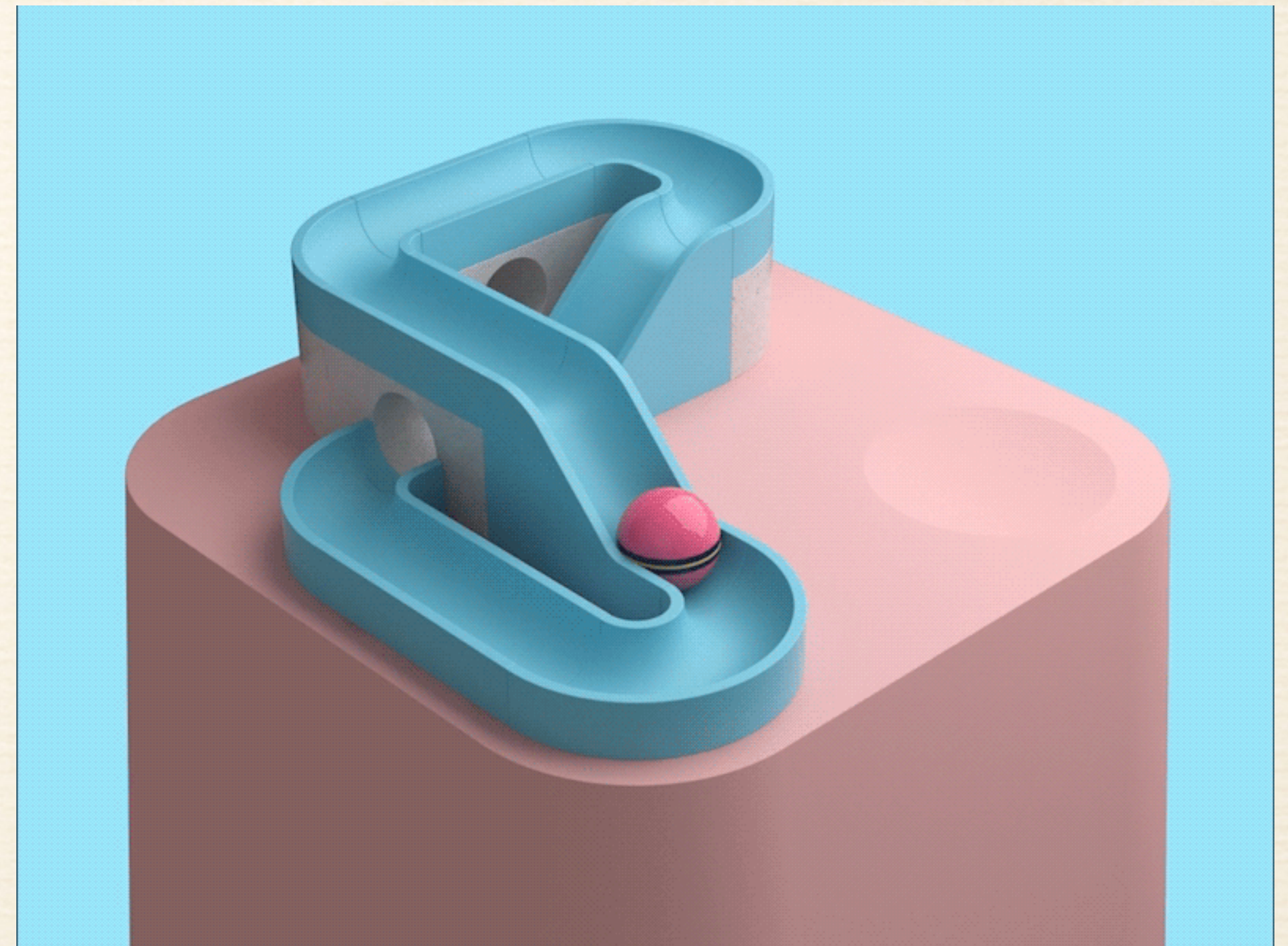
Running at 4x Speed

THIS VIDEO CAN BE FOUND LINKED IN THE GITHUB REPO



A warning about running CFGUTIL in a Loop

- ❖ Some chargers/Sync stations disconnect the iPad from USB connectivity momentarily to switch from charge mode to no charge mode. When this happens, if you are running CFGUTIL exec, it will detect that drop/connect as a new connection and wipe it again.
- ❖ Think infinity loop of iPad wiping...



Known to work:

- ❖ Lock N' Charge U16SCB
- ❖ Lock N' Charge ELE7060
- ❖ Cambrionix Thundersync 3-16 & Supersync 15
- ❖ OWC ThunderBolt 3 Dock
- ❖ Coolgear USB3-16U1 (Intel Only)
- ❖ Intel Macs with lots of USB A ports

Known not to Work

- ❖ Griffin Storage Docks (Black & Silver)
- ❖ USBC to multi USB-A port adapters.. Very finicky.

As a former Apple SE on our account used to tell me all the time “Mileage May Vary”

Why Intel only Hubs?

Multi Transactional Translators (TT) per Hub

A transaction translator is an important component of a high-speed hub device which, in essence, provides a communication link between the upstream-facing port of the hub and a downstream-facing port when they are operating at different data transfer rates. This is the case when the hub is connected to a high-speed host with full-speed or low-speed device(s) plugged into the ports facing downstream.



What about Automator?

- ❖ MacOS Catalina and down? Yup. You can do that.

<https://support.apple.com/guide/apple-configurator-2/automator-actions-cade0f4df56f/mac>

<https://configautomation.com>

- ❖ Big Sur and up? Nope. You can't install a wifi profile so you can't finish the device. Always failed.

*As of October 2021

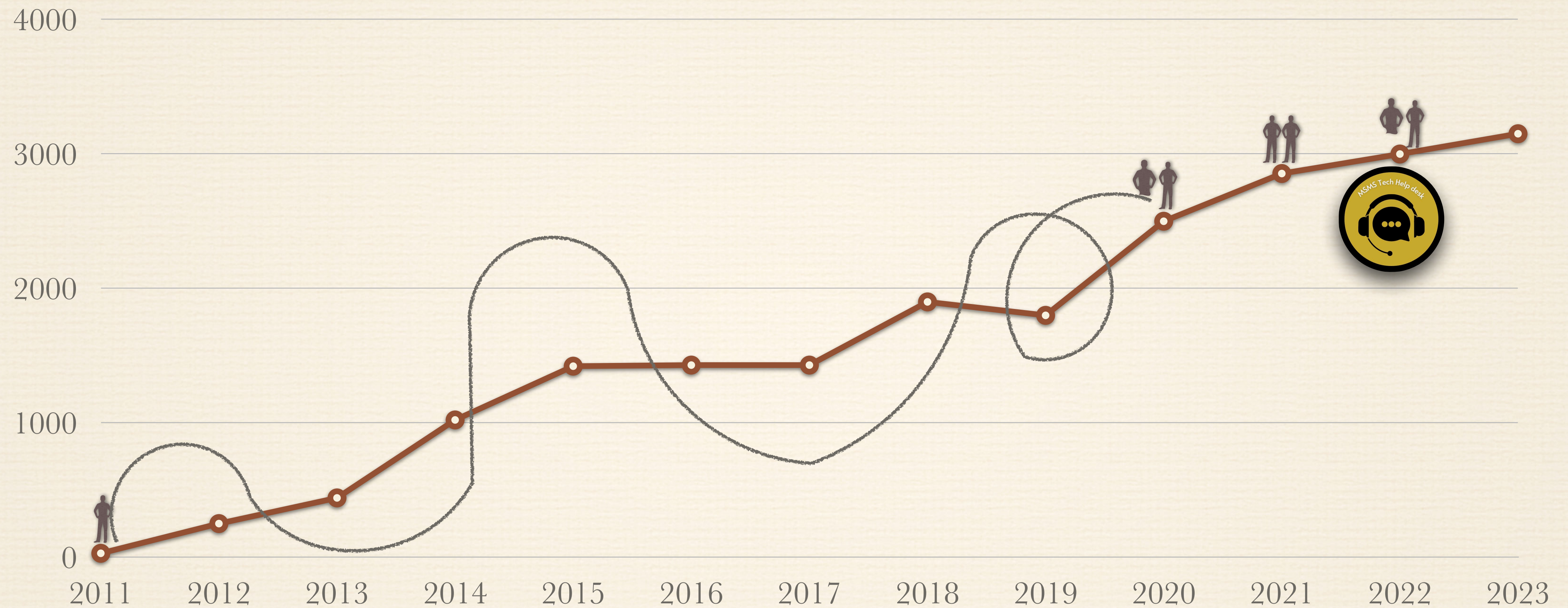
Am I reinventing the Wheel?

I mean I could wipe devices through Over the Air MDM Commands.

I could allow the Erase function locally to the end user.

There's plenty of ways to approach this problem.... So why this one?

iPad Growth in the District

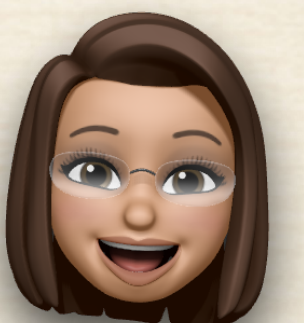


1 Tech

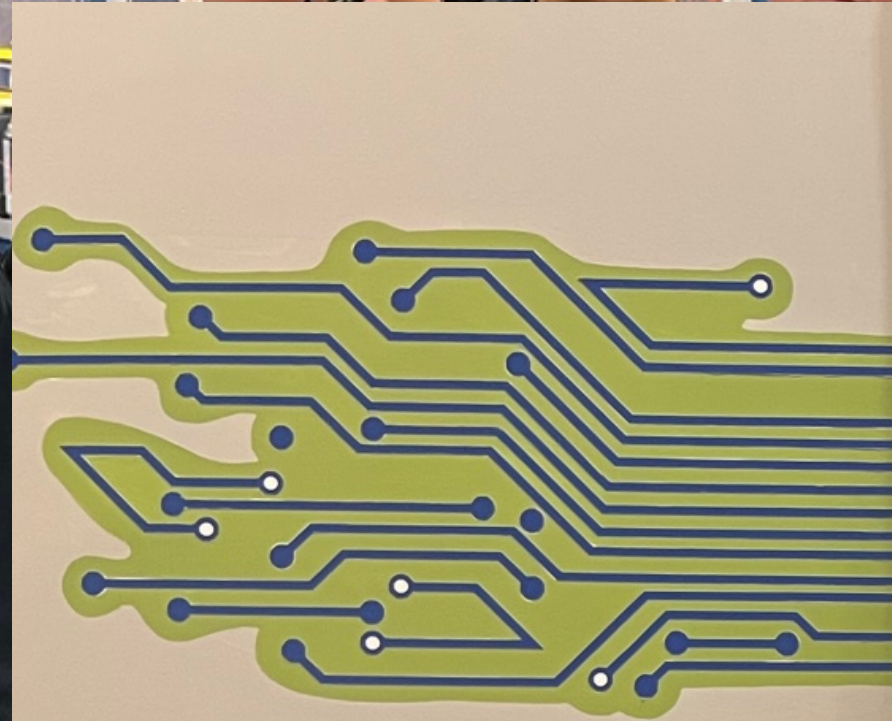


1/2 Tech

*Jennifer
Czyzerewski*



Help Desk Year 1 Group



AGiREPAIR

Pioneering the Future of Apple Parts & Repairs

K-12 Repair and Replacement Parts

iPad | MacBook Air | Chromebook



iPad issues/tickets/etc for instance:

- ❖ We handle all iPad recovery from outside. Mostly in cases where student withdrew but didn't return their hardware.... (Lost mode enabled, Inventory status change, etc.)
- ❖ New iPad assignments.
- ❖ Returned devices long thought lost to time.
- ❖ “My iPad does some weird thing” send ups from the Elementaries.

All these cases have something in common:

Device in my hand.

A script could.....

- ❖ Check the device (with trust.) Issue a Restore or Erase command depending on the state of the device.
- ❖ No Trust? Cool.. um do it some other way...
- ❖ Remove lost mode and clear back log commands too.
- ❖ Change device statuses and assignments in inventory.

**What could such a
script do for you?**

So what now?



PHASE 1

PHASE 2

PHASE 3

Learn this
stuff



Profit



Introducing.....

Shake N' Bake

For iPads

Fancy bag NOT included

This is where we get not universal...

GSD is a Mosyle shop since April 20'. The extra functions I bolted on top use API calls to Mosyle utilizing functions and actions they have already provided. If your not a Mosyle customer this part will take some investigating on your part.

This is not as hard as it sounds and honestly a little Google searching and your bound to find some decent examples you can bend to your will.

ShakeNBake.sh

- ❖ Combine MOSBasic Libraries with CFGUTIL to automate iPad cycle
- ❖ Automated Wipe, Update, Provision, and then let DEP bat clean up.
- ❖ Work around Trust issues by leveraging MDM wipe ability.
- ❖ Clear back logged commands in MDM before setting up device.
- ❖ Disable Lost mode on the iPad we are working on.

MOSBasic

Home grown command line utility to manage iPads in Mosyle doing the most common jobs:

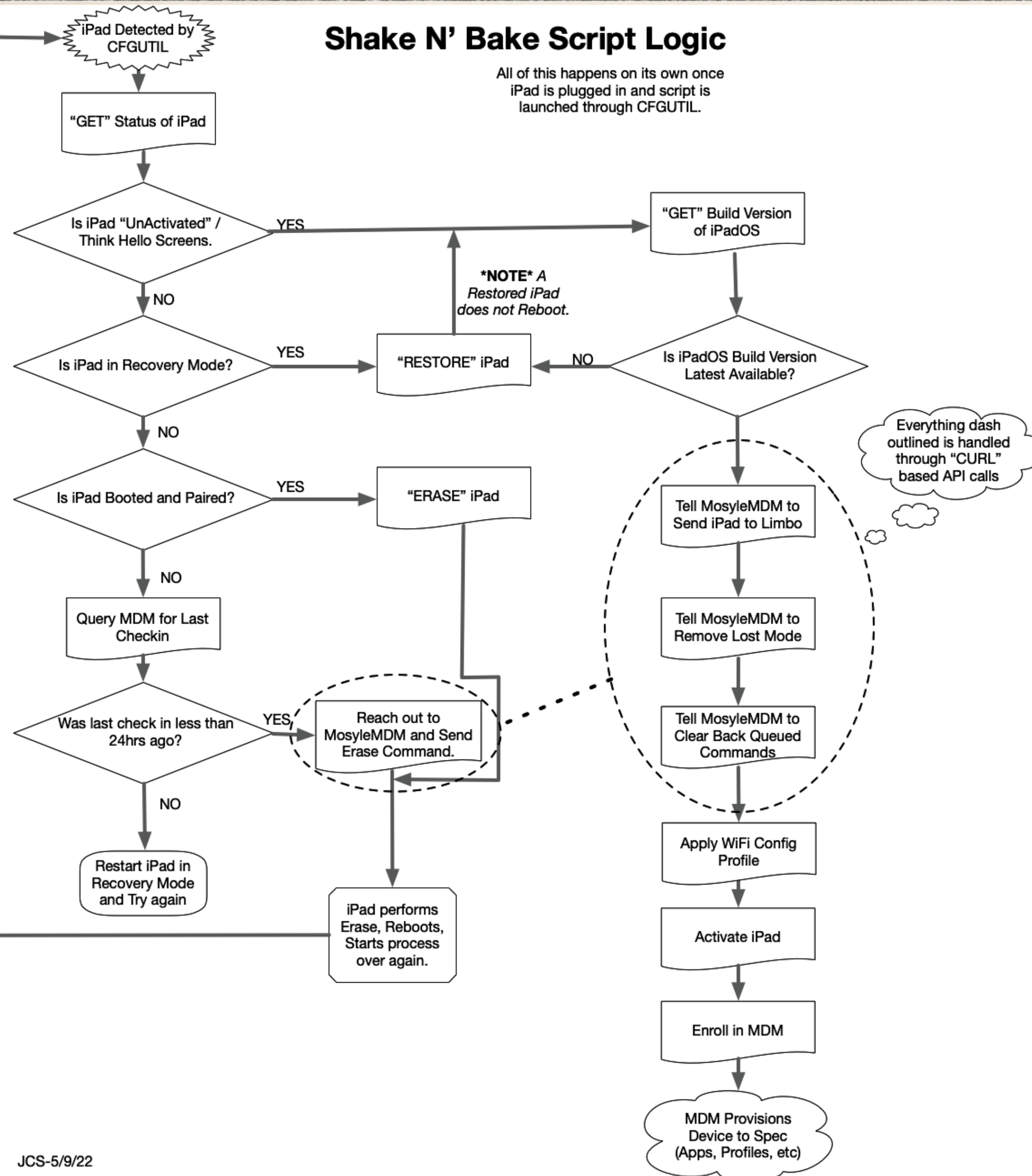
- ❖ Erase
- ❖ Assign
- ❖ Unassign
- ❖ Lost mode activities
- ❖ CSV export of all AppleTVs, iPads, and Macs

ShakeNBake.sh

- ❖ Combine MOSBasic Libraries with CFGUTIL to automate iPad cycle
- ❖ Automated Wipe, Update, Provision, and then let DEP bat clean up.
- ❖ Work around Trust issues by leveraging MDM wipe ability.
- ❖ Clear back logged commands in MDM before setting up device.
- ❖ Disable Lost mode on the iPad we are working on.

Shake N' Bake Script Logic

All of this happens on its own once iPad is plugged in and script is launched through CFGUTIL.



This is in the presentation
repo.. Reference *Shake N Bake*
Logic

Live Demo Warning....





ShellChe



Photos

Library

Memories

Shared with You

People

Places

Favorites

Recents

Imports

Duplicates

Albums

> Media Types

> Shared Albums

> My Albums

Projects

> My Projects



Places



Search

Places

Favorites

Recents

Applications

Desktop

Documents

GitHub

Downloads

RandomEx...

Creative Cl...

Shillie

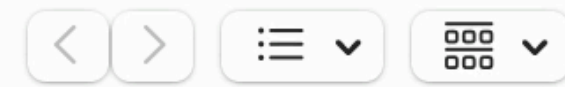
GatorShared

iCloud

QuickTime...

iCloud Drive

Shared



QuickTime Player — iCloud

Search

Name

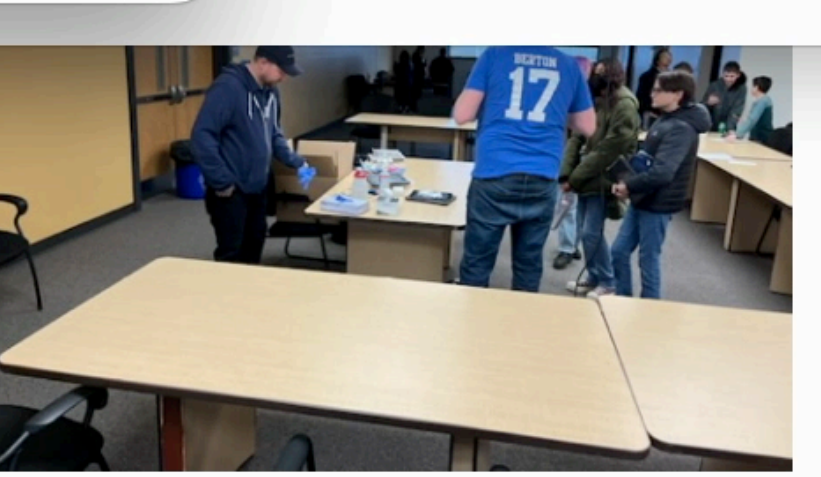
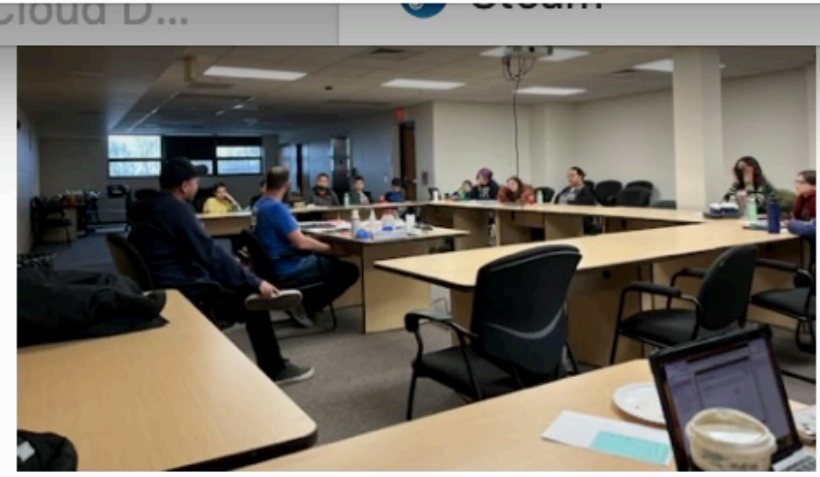
Date Modified

Size

THIS VIDEO OF MY AGONY IS NOT
IN THE REPO.... I'D LIKE TO
FORGET IT HAPPENED!

Cancel

Open



temporarily turn on when you reset the drive.

Live Demo Warning....





What about you Other MDM people?

- ❖ ShakeNBake.sh utilizes functions from MOSBasic as well as cache'd data.
- ❖ If you can get this data out of your existing MDM or Ticket system you could easily adapt. Code is commented where necessary.

Feel free to branch my Repo and adapt to your needs.

<https://github.com/JCSmillie/MOSBasic>

NOTE Watch for this repo to update after the presentation for a newer version of MOSBasic

Jesse C. Smillie

Mac & Linux Operations Engineer

MacAdmins Slack: @jsmillie

jsmillie@gatewayk12.org



Github Resources:

bit.ly/JesseSmiPSUMacPrez

Feedback:

bit.ly/psumac2023-134

