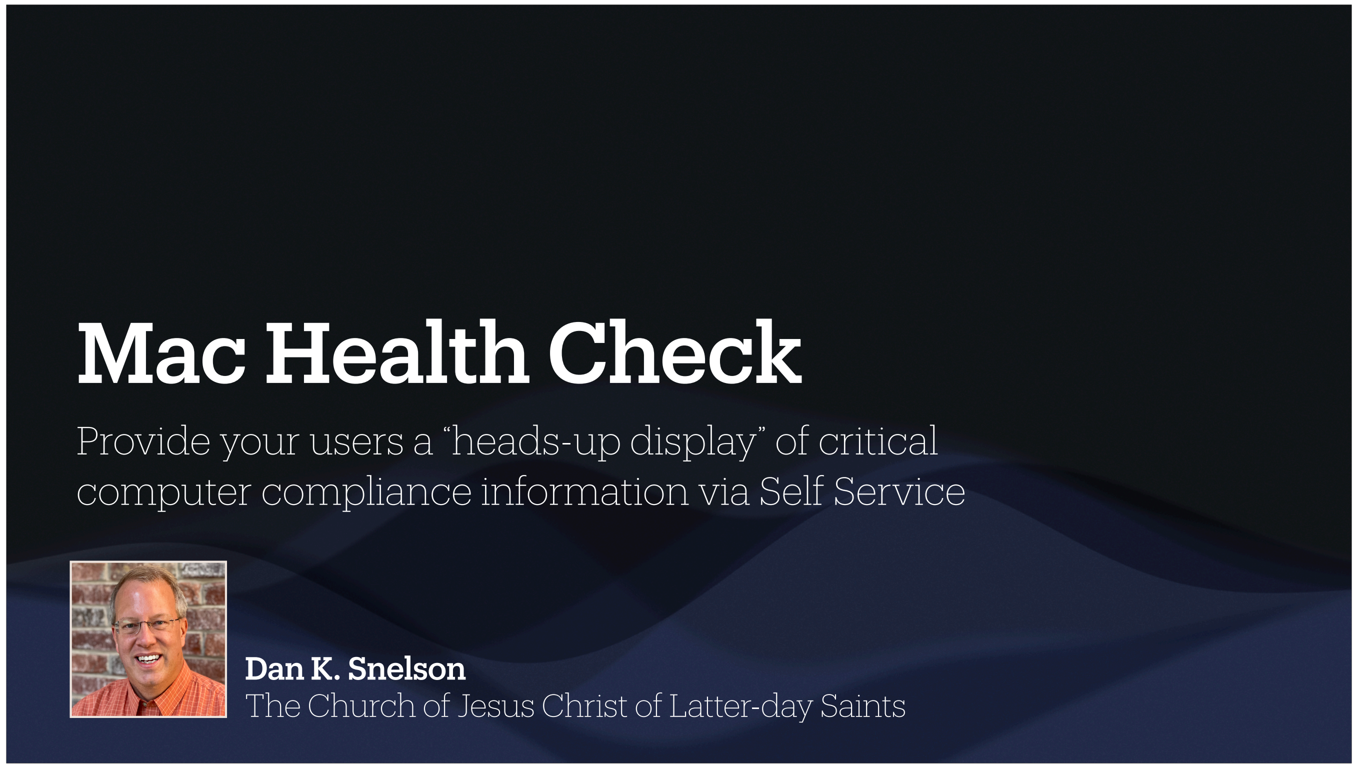


Mac Health Check

Provide your users a "heads-up display" of critical computer compliance information via Self Service

Dan K. Snelson
The Church of Jesus Christ of Latter-day Saints

[Advance]

Happy Wednesday, U of U Mac Admins!

[... Pause ...]

Great to be back! Thanks for having me again, Richard.

[... Pause ...]

My name is Dan Snelson, and I am a Senior System Engineer with The Church of Jesus Christ of Latter-day Saints.

We are sometimes referred to as "Mormons" because of our belief in a volume of companion scripture to the *Holy Bible* called *The Book of Mormon*.

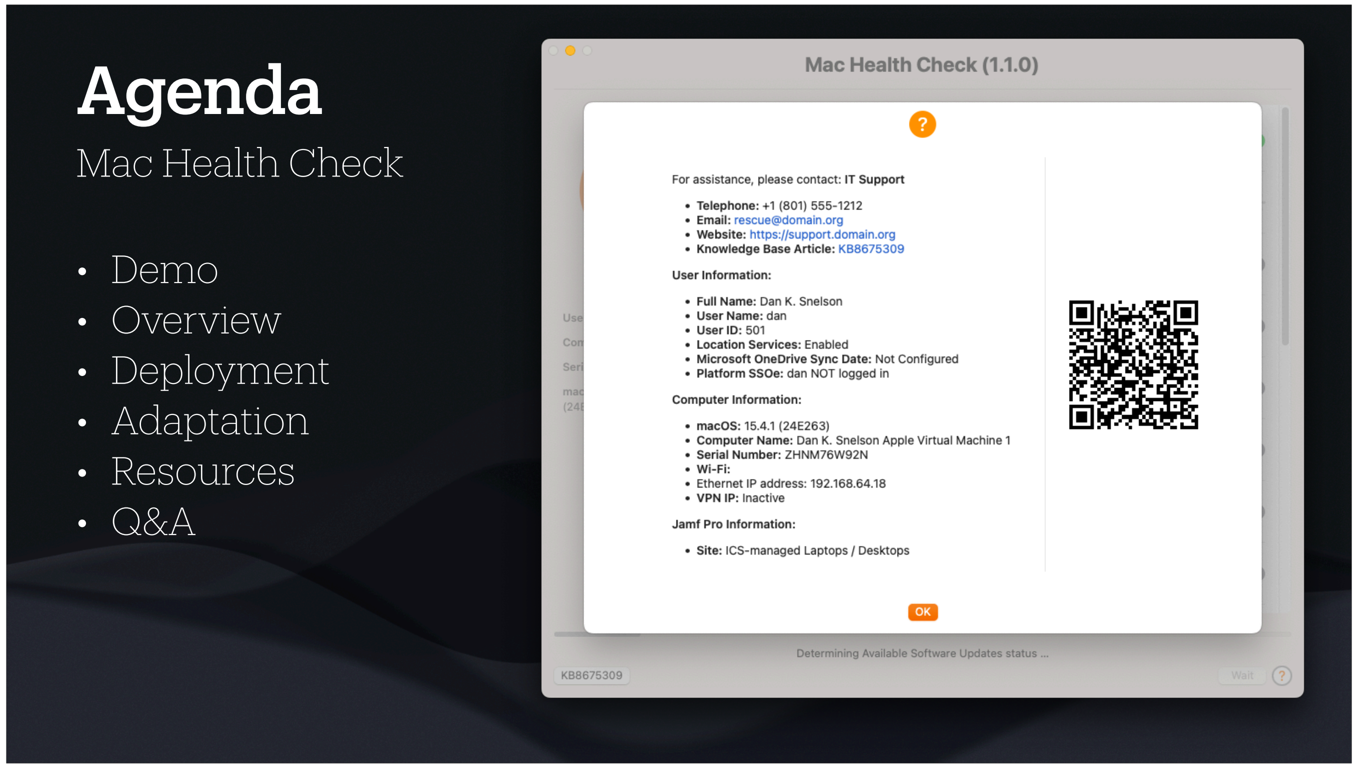
[... Pause ...]

Let's talk about how you can easily provide your users a "heads-up display" of critical computer compliance information via Self Service.

[... Pause ...]

[Advance]

[Advance]



Agenda

Mac Health Check

- Demo
- Overview
- Deployment
- Adaptation
- Resources
- Q&A

For assistance, please contact: **IT Support**

- **Telephone:** +1 (801) 555-1212
- **Email:** rescue@domain.org
- **Website:** https://support.domain.org
- **Knowledge Base Article:** KB8675309

User Information:

- **Full Name:** Dan K. Snelson
- **User Name:** dan
- **User ID:** 501
- **Location Services:** Enabled
- **Microsoft OneDrive Sync Date:** Not Configured
- **Platform SSOe:** dan NOT logged in

Computer Information:

- **macOS:** 15.4.1 (24E263)
- **Computer Name:** Dan K. Snelson Apple Virtual Machine 1
- **Serial Number:** ZHNM76W92N
- **Wi-Fi:**
- Ethernet IP address: 192.168.64.18
- **VPN IP:** Inactive

Jamf Pro Information:

- **Site:** ICS-managed Laptops / Desktops

Here's our agenda …

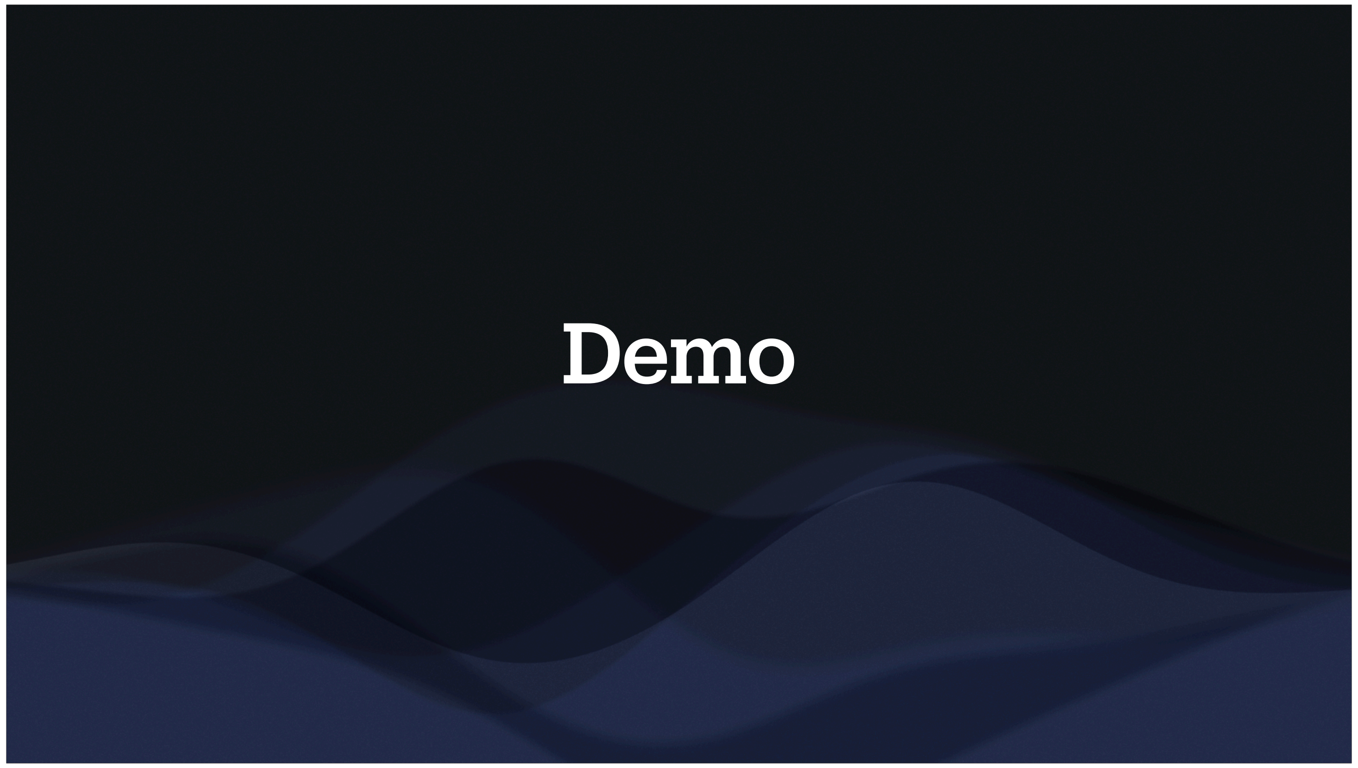
… we'll start off with a demo, followed by some short overviews, then we'll *briefly* discuss deployment.

We'll spend the bulk of our time together discussing how **you** can ***adapt*** the various checks for **your** environment.

We'll then finish up with some resources, after which I'll try to answer any questions you may have.

[… Pause …]

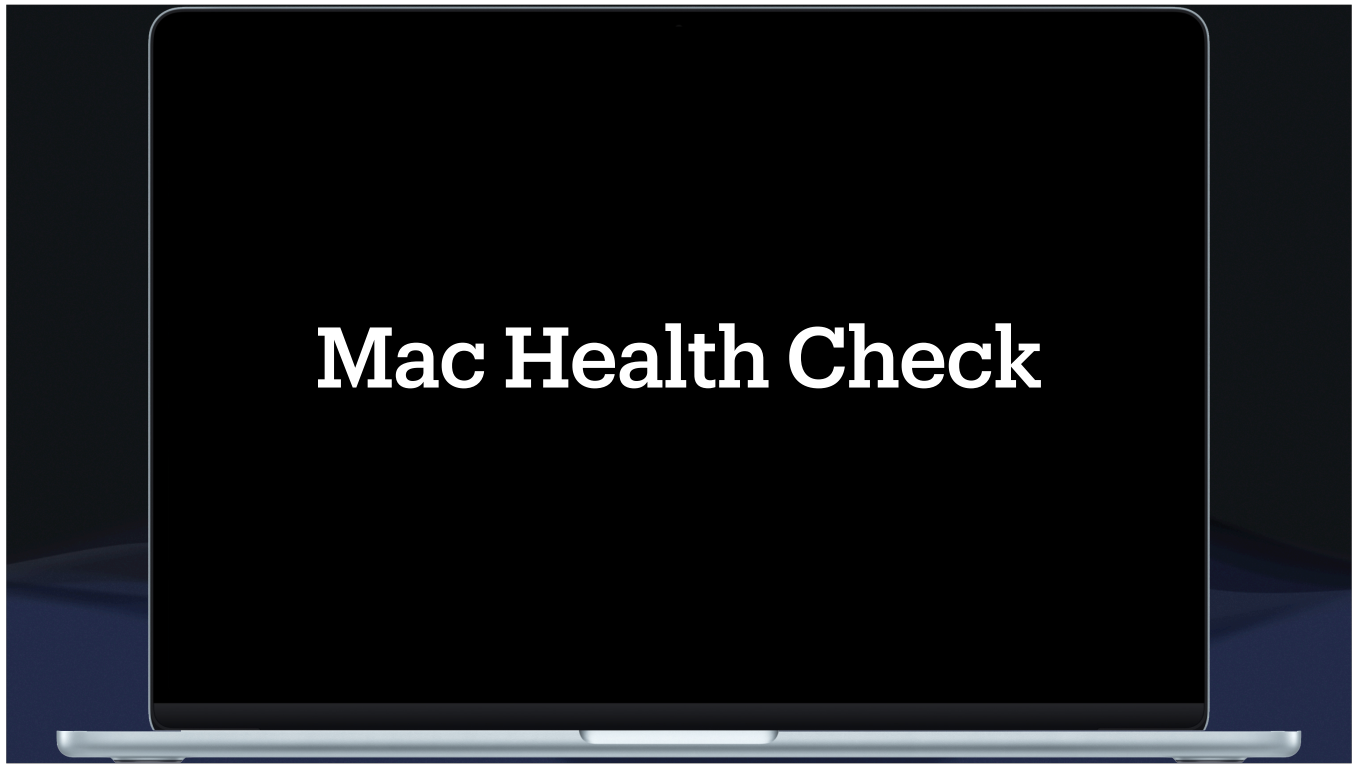
[Advance]



Demo

Let’s look at a demo of Mac Health Check …

[Advance]



[Advance]

Here's a demo of Mac Health Check being executed via Self Service, which we call the Workforce App Store.

[… Pause for half of the spinning wheel …]

The user is immediately shown the various compliance checks …

… and clicking the Help button reveals static information, all in a single, "heads-up" location.

[… Pause for ten seconds …]

Included compliance checks leverage online, third-party resources …

[… Pause for five seconds …]

… built-in macOS binaries …

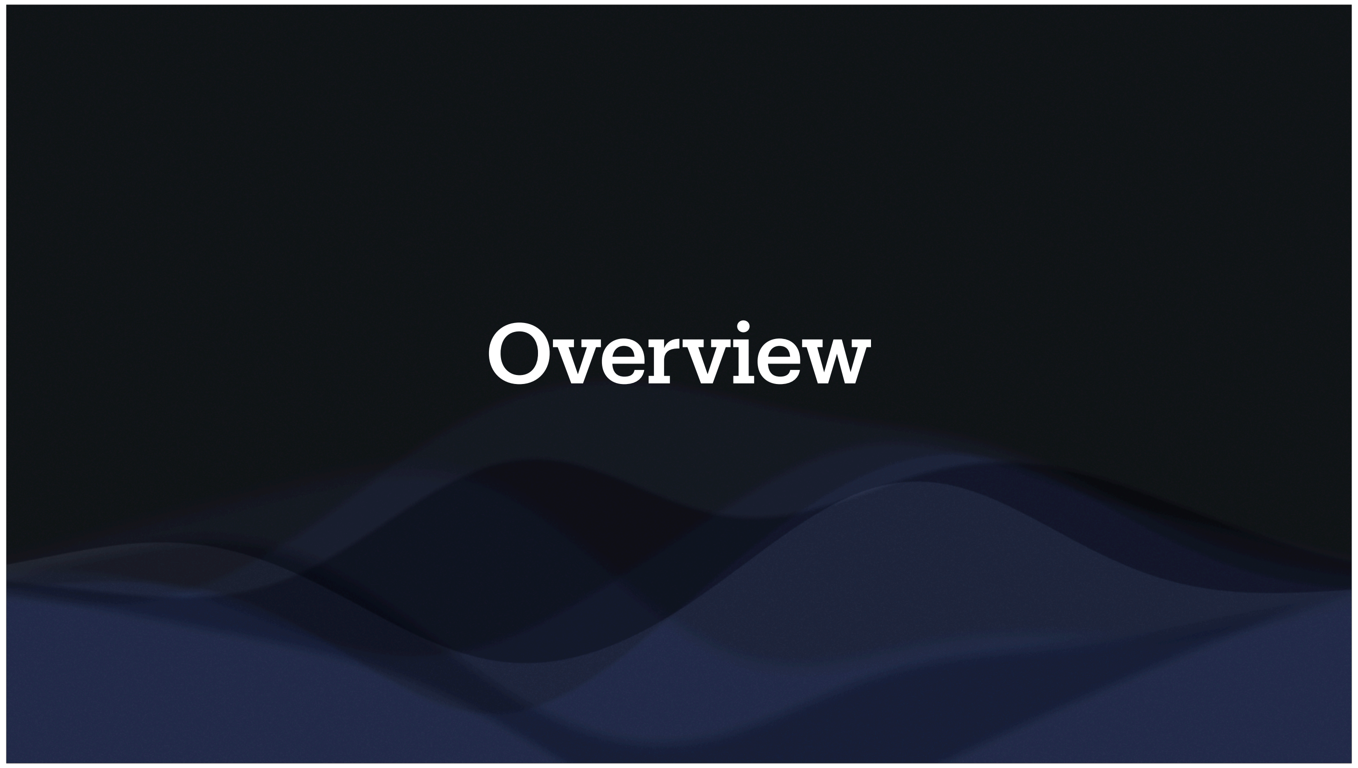
[… Pause …]

… as well as Jamf Pro policies.

[… Pause for script to complete …]

So, that was a demo of Mac Health Check being executed via Self Service.

[Advance]

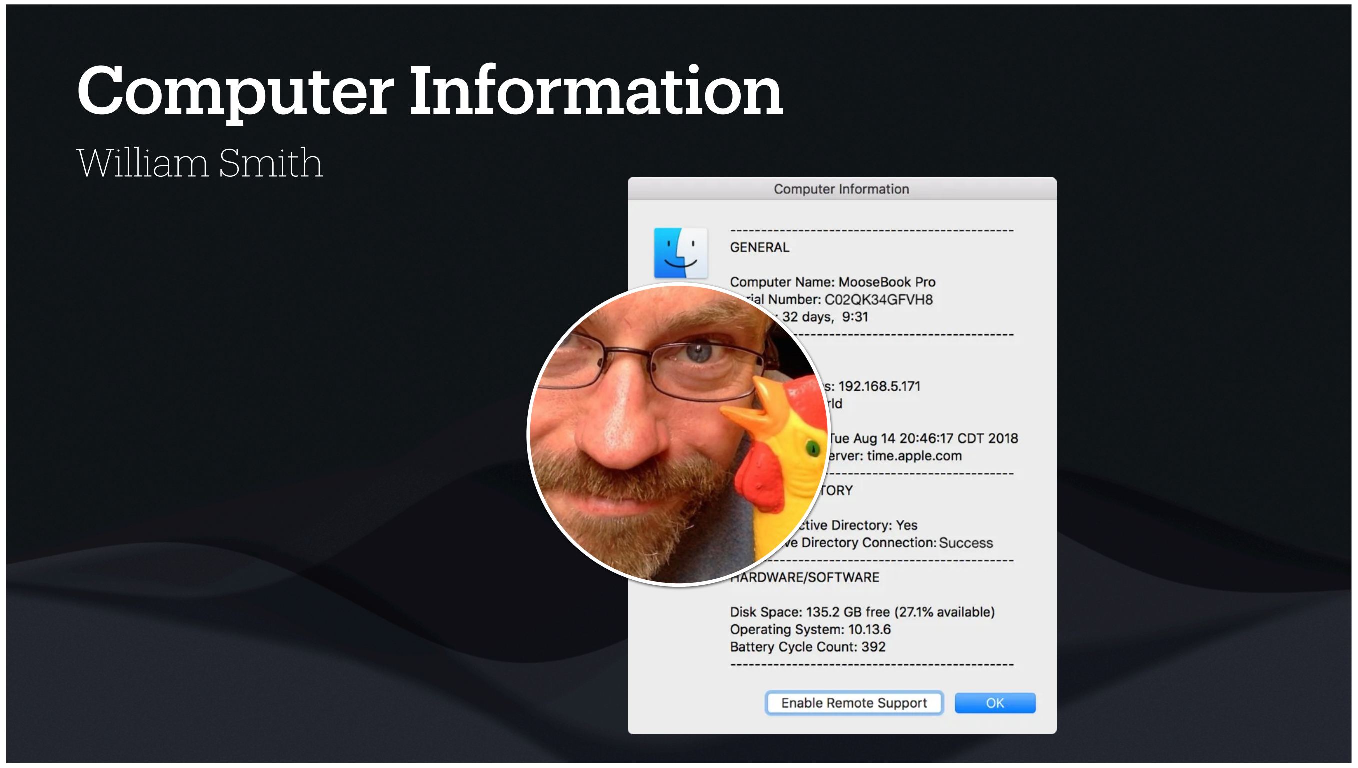


Overview

Let’s level-set with a few overviews …

[… Pause …]

[Advance]



Computer Information

William Smith

```
Computer Information

--------------------------------------------
GENERAL

Computer Name: MooseBook Pro
 rial Number: C02QK34GFVH8
 : 32 days,  9:31
--------------------------------------------

 s: 192.168.5.171
 rld

 Tue Aug 14 20:46:17 CDT 2018
 erver: time.apple.com
--------------------------------------------
 TORY

 ctive Directory: Yes
 ve Directory Connection: Success
--------------------------------------------
HARDWARE/SOFTWARE

Disk Space: 135.2 GB free (27.1% available)
Operating System: 10.13.6
Battery Cycle Count: 392
--------------------------------------------
```

First, the “original” inspiration for Mac Health Check came from none other than the “talkingmoose” himself (who is *also* mentioned on line number 9 of Installomator, for you trivia buffs.)

[… Pause …]

[Advance]

Bill’s version leveraged AppleScript.

[… Pause …]

Check out the 32 days of uptime from back on the 14th of August, 2018 …

… where were you seven years ago about the middle of next month?

[… Pause …]

Bound to Active Directory and rocking macOS 10.13.6, baby! …

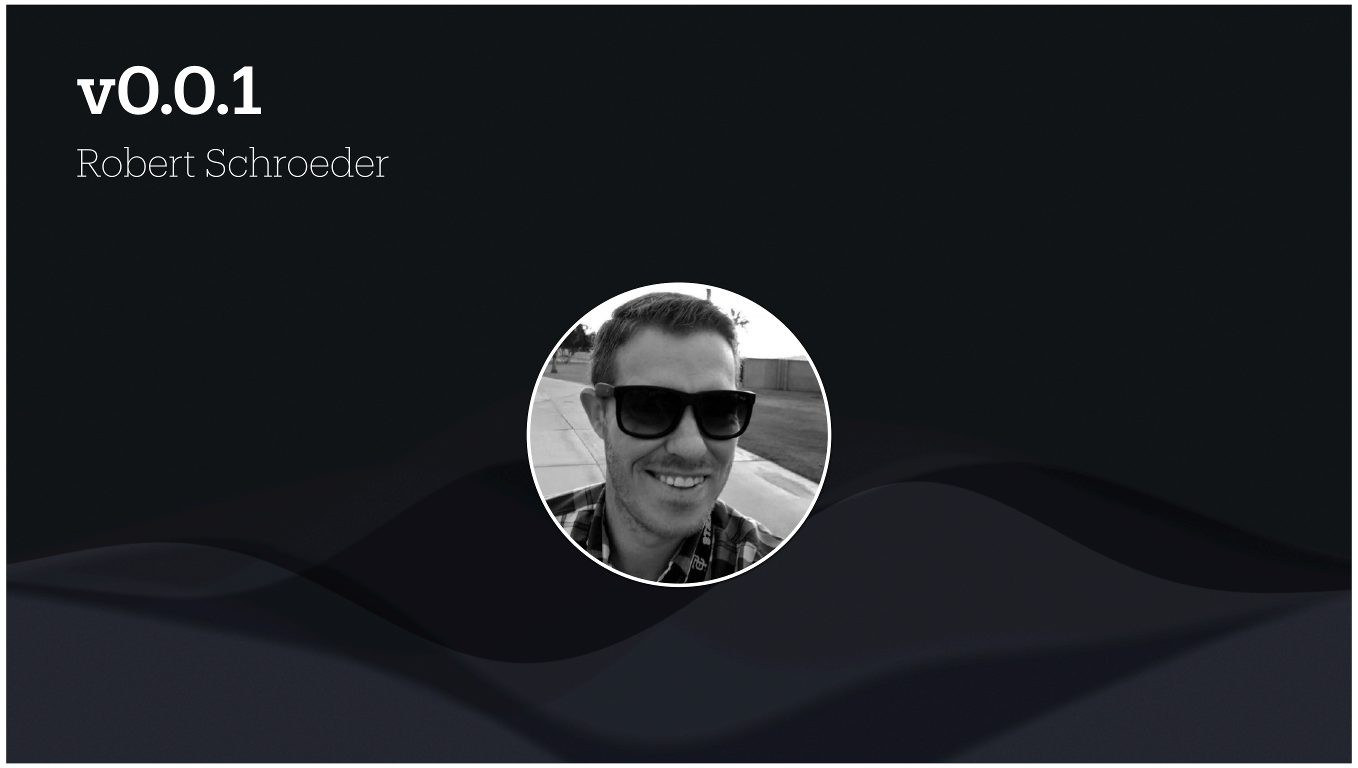
[… Pause …]

... love the Battery Cycle Count, too ...

... ahh ... good times.

[... Pause ...]

[Advance]



v0.0.1

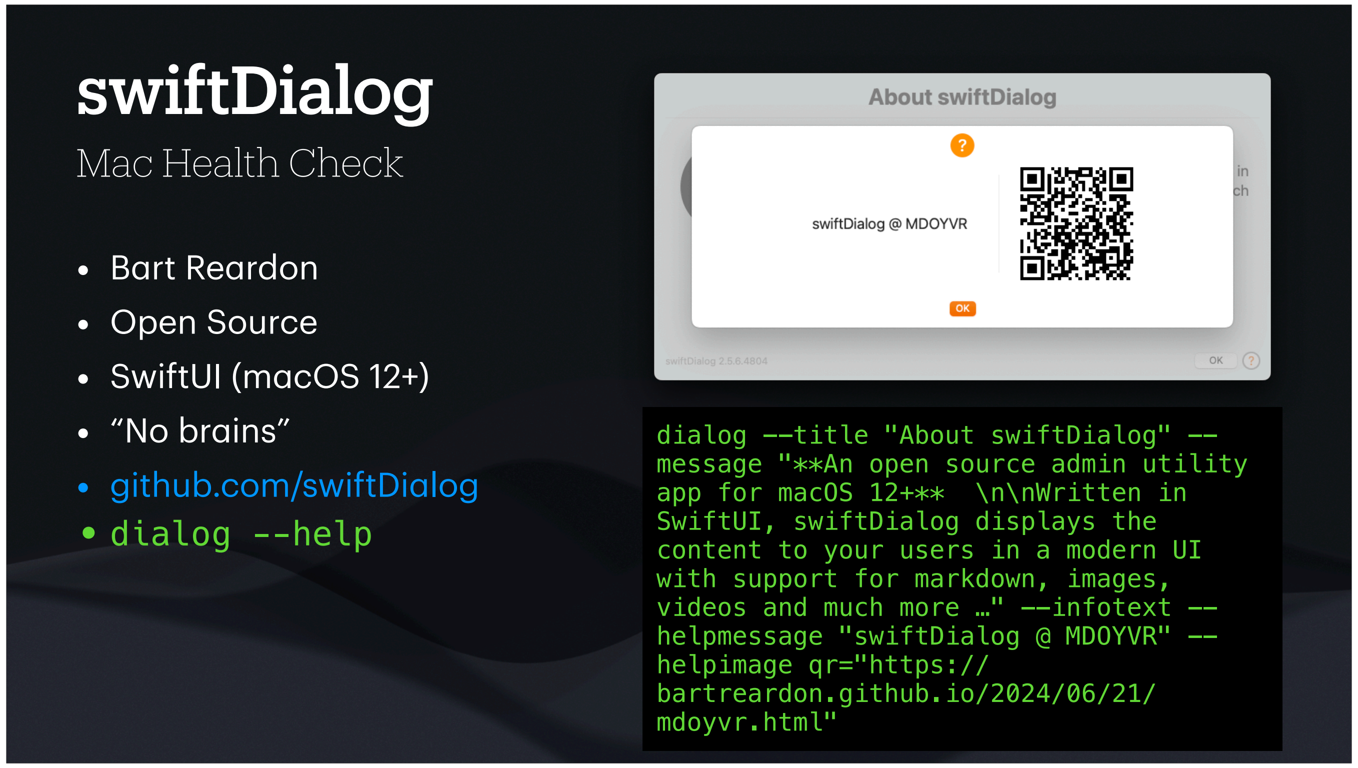
Robert Schroeder

Also, a shout-out to Robert Schroeder for helping me jumpstart version 0.0.1 …

… thanks again, Rob!

[… Pause …]

[Advance]



swiftDialog

Mac Health Check

- Bart Reardon
- Open Source
- SwiftUI (macOS 12+)
- “No brains”
- github.com/swiftDialog
- `dialog --help`

```
dialog --title "About swiftDialog" --
message "**An open source admin utility
app for macOS 12+**  \n\nWritten in
SwiftUI, swiftDialog displays the
content to your users in a modern UI
with support for markdown, images,
videos and much more …" --infotext --
helpmessage "swiftDialog @ MDOYVR" --
helpimage qr="https://
bartreardon.github.io/2024/06/21/
mdoyvr.html"
```

Next, ***none*** of this would be possible without Bart and swiftDialog!

[Advance]

Bart Reardon has provided the Mac Admin community swiftDialog:

An open-source utility written in SwiftUI that displays a popup dialog, which can include content-rich messages for your end-users.

While Bart is fond of saying that swfitDialog ***itself*** has “no brains,” swiftDialog ***can*** return various values from user input, on which your script can act and continue processing.

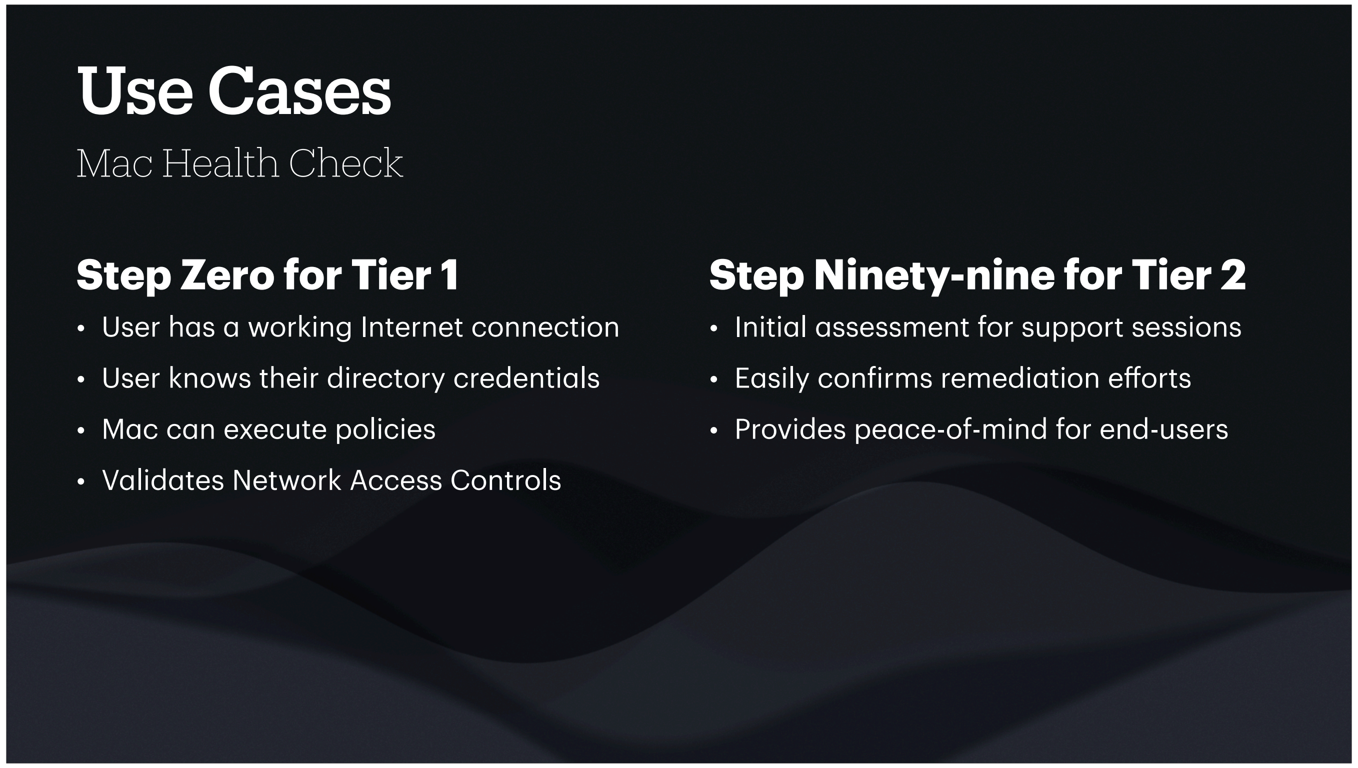
The project is available on GitHub, and Bart has an extensive Wiki with detailed examples.

Additionally, there’s built-in help when you need a quick answer.

Here’s a QR code where you can learn more.

[… Pause …]

[Advance]



Use Cases

Mac Health Check

Step Zero for Tier 1

- User has a working Internet connection
- User knows their directory credentials
- Mac can execute policies
- Validates Network Access Controls

Step Ninety-nine for Tier 2

- Initial assessment for support sessions
- Easily confirms remediation efforts
- Provides peace-of-mind for end-users

Let's talk about just **two** use-cases: Both help your internal support representatives to succeed (and I'm confident you can think of additional use-cases for Mac Health Check.)

[… Pause …]

[Advance]

For your Tier 1 support group, Mac Health Check can be used as "Step Zero" to quickly execute a variety of basic troubleshooting steps:

- Does the user have a working Internet connection?
- Does the user know their directory credentials?
- Can the Mac actually execute policies?
- Will the Mac pass your organization's NAC requirements?

These questions and more can be answered in about two minutes by having the end-user run Mac Health Check via Self Service at the **start** of every Tier 1 call.

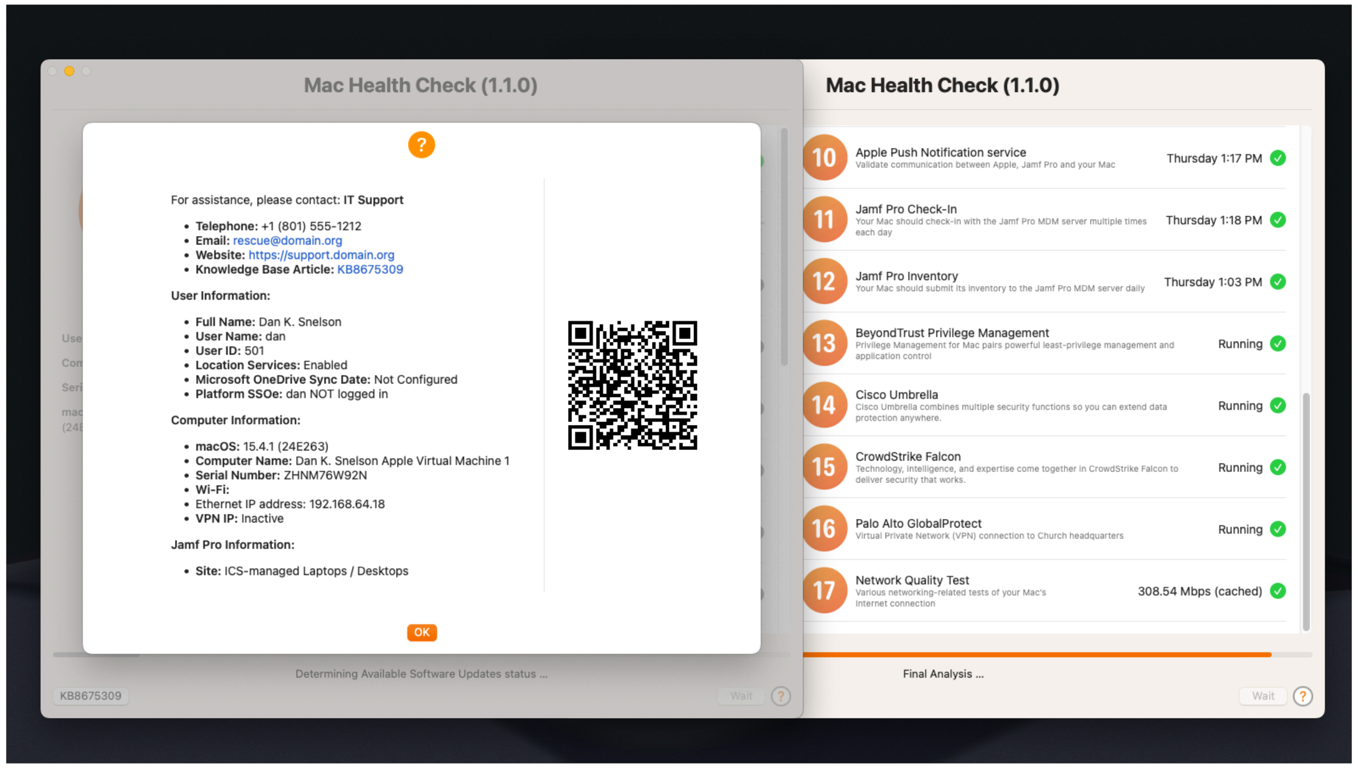
[… Pause …]

[Advance]

For Tier 2, Mac Health Check can be used as “Step 99” to easily confirm remediation efforts were successful (and as the initial assessment for remote support sessions, to help ensure an accurate diagnosis by Tier 1 support).

[… Pause …]

[Advance]



For assistance, please contact: **IT Support**

- **Telephone:** +1 (801) 555-1212
- **Email:** rescue@domain.org
- **Website:** https://support.domain.org
- **Knowledge Base Article:** KB8675309

User Information:

- **Full Name:** Dan K. Snelson
- **User Name:** dan
- **User ID:** 501
- **Location Services:** Enabled
- **Microsoft OneDrive Sync Date:** Not Configured
- **Platform SSOe:** dan NOT logged in

Computer Information:

- **macOS:** 15.4.1 (24E263)
- **Computer Name:** Dan K. Snelson Apple Virtual Machine 1
- **Serial Number:** ZHNM76W92N
- **Wi-Fi:**
- Ethernet IP address: 192.168.64.18
- **VPN IP:** Inactive

Jamf Pro Information:

- **Site:** ICS-managed Laptops / Desktops

Determining Available Software Updates status ...

10	Apple Push Notification service Validate communication between Apple, Jamf Pro and your Mac	Thursday 1:17 PM ✓
11	Jamf Pro Check-In Your Mac should check-in with the Jamf Pro MDM server multiple times each day	Thursday 1:18 PM ✓
12	Jamf Pro Inventory Your Mac should submit its inventory to the Jamf Pro MDM server daily	Thursday 1:03 PM ✓
13	BeyondTrust Privilege Management Privilege Management for Mac pairs powerful least-privilege management and application control	Running ✓
14	Cisco Umbrella Cisco Umbrella combines multiple security functions so you can extend data protection anywhere.	Running ✓
15	CrowdStrike Falcon Technology, intelligence, and expertise come together in CrowdStrike Falcon to deliver security that works.	Running ✓
16	Palo Alto GlobalProtect Virtual Private Network (VPN) connection to Church headquarters	Running ✓
17	Network Quality Test Various networking-related tests of your Mac's Internet connection	308.54 Mbps (cached) ✓

Final Analysis ...

Let's discuss the *general* approach for the information displayed to the end-user in Mac Health Check.

[… Pause …]

[Advance]

Generally, compliance checks are the more time-intensive processes.

For example, Rob Schroeder's bundled function which queries the Mac Admins "Simple Organized Feed for Apple Software Updates (SOFA)" and compares it to your organizationally defined requirements for compliant versions of macOS (for example, "N minus two").

Or, the Isaac Mann-inspired, MDM-agnostic Apple Push Notification service check, which looks at the last 24 hours of log entries for the "com.apple.ManagedClient" service.

Or, Jordy Witteman's bundled functions from his excellent Support app, which validate the most recent Jamf Pro check-in and inventory update.

These checks and more are concluded with the built-in, macOS `networkQuality` binary, which can help determine the speed of the Mac's Internet connection, the results of which can be cached for any duration of your choosing.

So, any time-intensive test — or anything you want displayed front-and-center to the end-user — should be considered a "Compliance Check."

[… Pause …]

[Advance]

Static Information Reporting is contained in the `helpmessage` variable, which users can view by clicking the question mark in the bottom, right-hand corner.

This is the “heads-up display” of critical information support representatives need when triaging a Mac.

Additionally, a few under-the-hood settings are included in the Jamf Pro policy log *only,* which can prove helpful for a more holistic understanding of each Mac.

For example, Location Services, SSH and Battery Cycle Count; information a Mac Admin is interested in, but which you may not want to directly expose to your end-users.

[Advance]

[… Pause …]

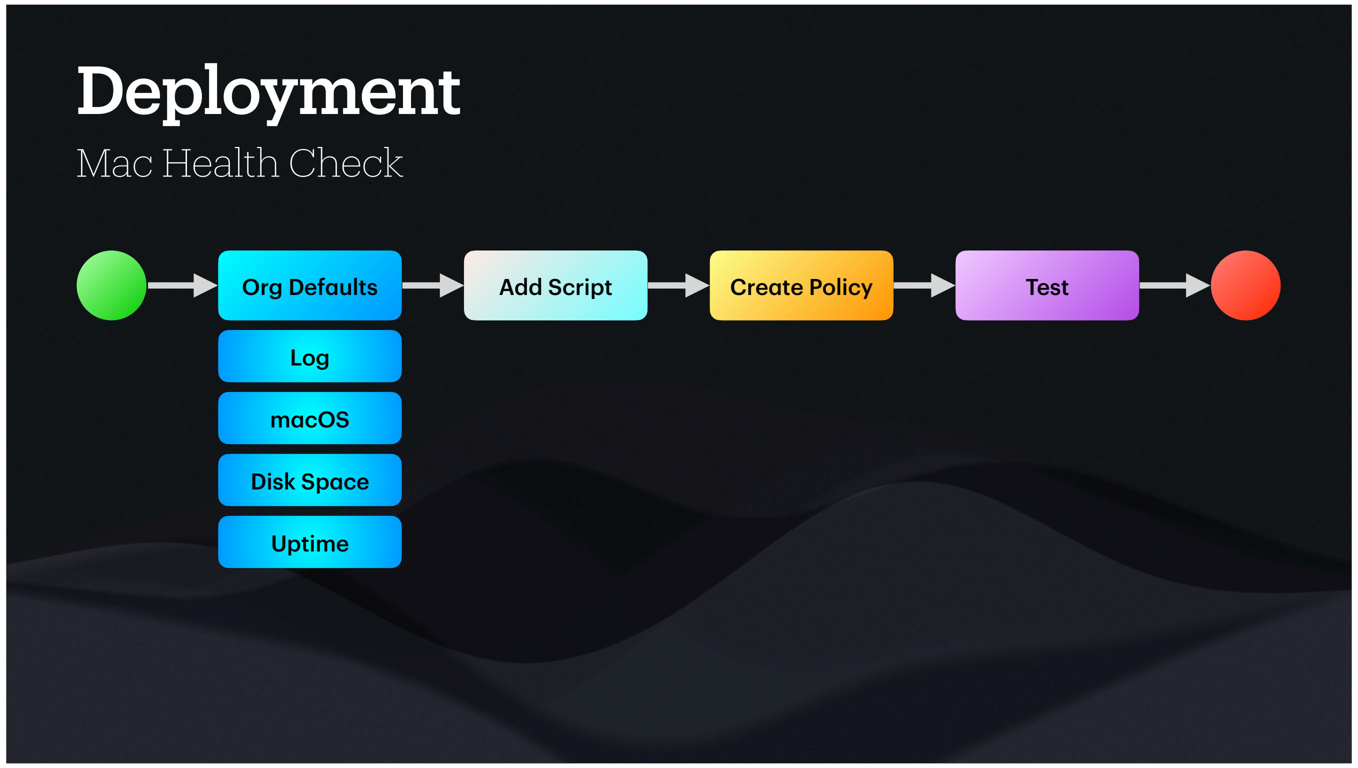
[Advance]

Deployment

Let’s briefly discuss deployment …

[… Pause …]

[Advance]



For your initial deployment …

[Advance]

… you'll modify the script in your preferred code editor, starting with specifying your organizational defaults. For example:

- The path to your client-side script logs
- Your organization's color scheme
- How many previous versions of macOS are considered compliant
- How much much free disk space is required
- Allowed uptime

You get the idea.

[… Pause …]

[Advance]

You'll then add the script to your Jamf Pro server …

[… Pause …]

[Advance]

… then create a policy to execute the script …

[… Pause …]

[Advance]

… then test, test, test.

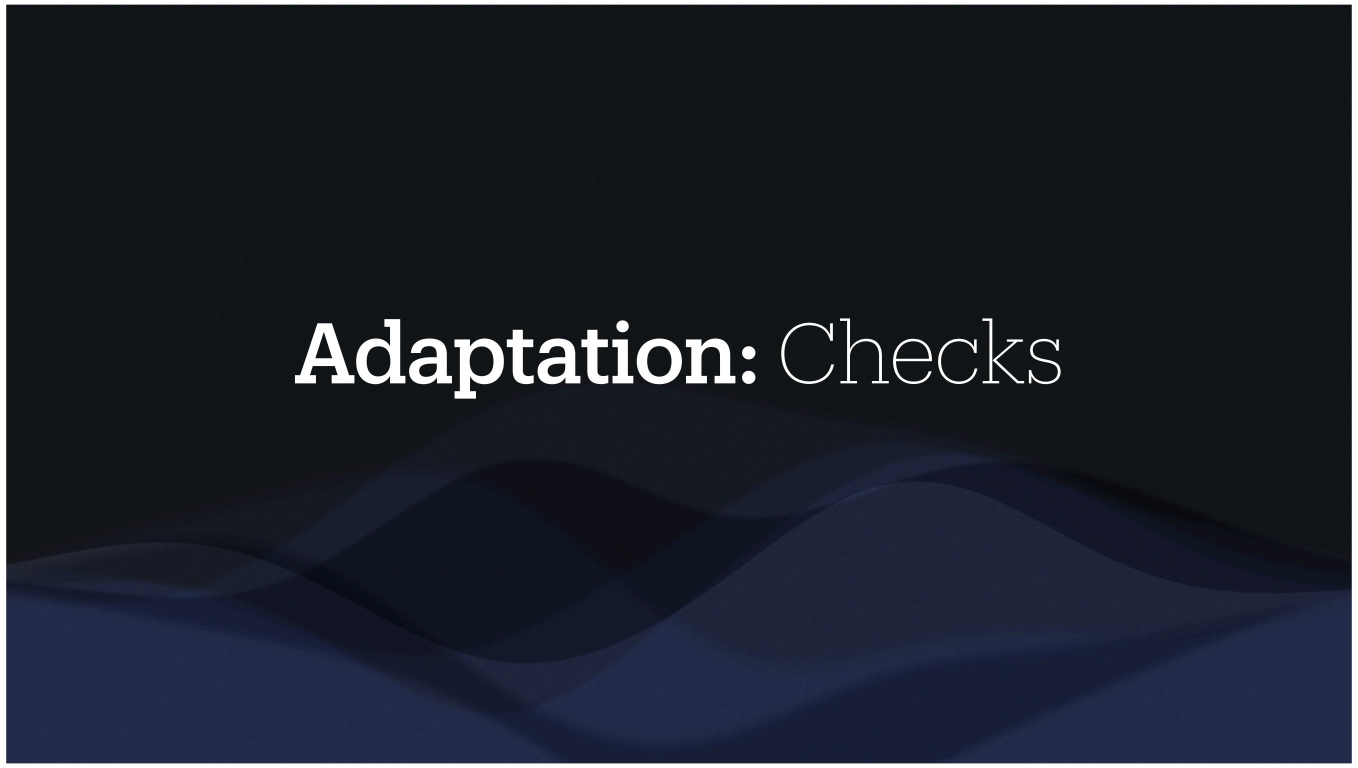
[… Pause …]

[Advance]

Pretty straightforward and nothing you haven't done hundreds of times already.

[… Pause …]

[Advance]



Adaptation: Checks

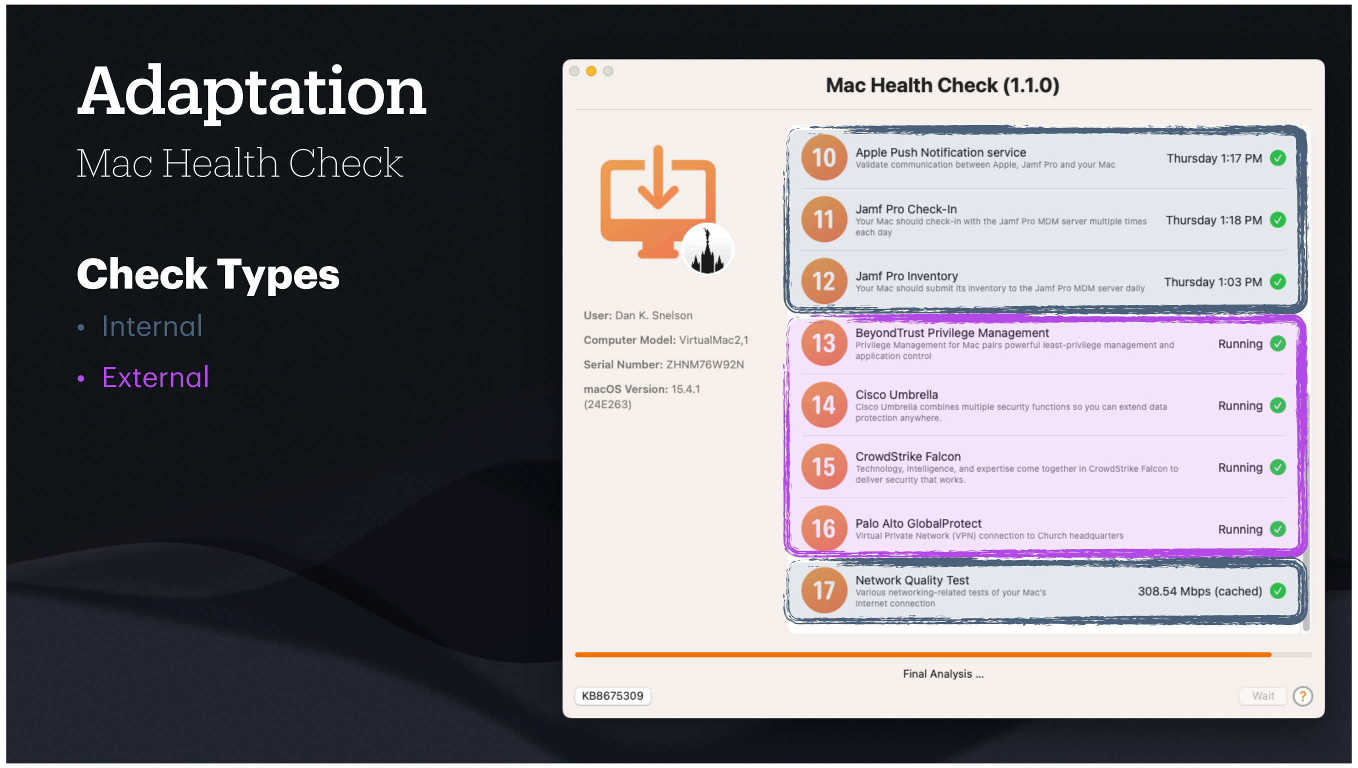
Where I wanted to spend the bulk of our time together is on adaptation for **your** environment.

[… Pause …]

Before you start adapting Mac Health Check, it’s important to understand that there are **two** types of checks …

[… Pause …]

[Advance]



Adaptation

Mac Health Check

Check Types

- Internal
- External

Mac Health Check (1.1.0)

User: Dan K. Snelson

Computer Model: VirtualMac2,1

Serial Number: ZHNM76W92N

macOS Version: 15.4.1 (24E263)

10	Apple Push Notification service Validate communication between Apple, Jamf Pro and your Mac	Thursday 1:17 PM ✓
11	Jamf Pro Check-In Your Mac should check-in with the Jamf Pro MDM server multiple times each day	Thursday 1:18 PM ✓
12	Jamf Pro Inventory Your Mac should submit its inventory to the Jamf Pro MDM server daily	Thursday 1:03 PM ✓
13	BeyondTrust Privilege Management Privilege Management for Mac pairs powerful least-privilege management and application control	Running ✓
14	Cisco Umbrella Cisco Umbrella combines multiple security functions so you can extend data protection anywhere.	Running ✓
15	CrowdStrike Falcon Technology, intelligence, and expertise come together in CrowdStrike Falcon to deliver security that works.	Running ✓
16	Palo Alto GlobalProtect Virtual Private Network (VPN) connection to Church headquarters	Running ✓
17	Network Quality Test Various networking-related tests of your Mac's Internet connection	308.54 Mbps (cached) ✓

Final Analysis ...

KB8675309

The first type of check is …

[Advance]

… internal.

For example, parsing client-side log files to determine when the Mac last submitted its inventory, or leveraging built-in macOS binaries like `networkQuality` to determine the user's Internet connection speed.

[… Pause …]

The second type of check is …

[Advance]

… external.

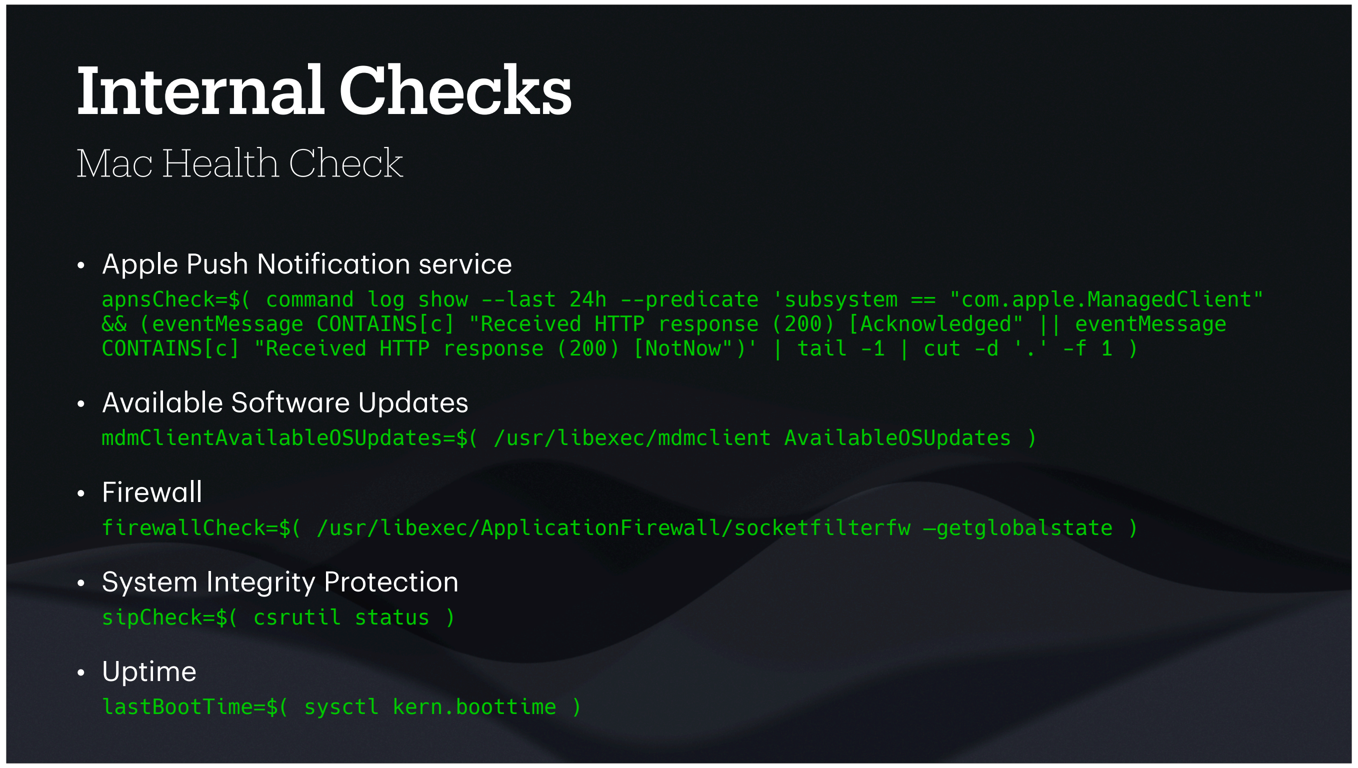
For example, yes, CrowdStrike Falcon is *installed*, but is it **running**?

External checks leverage scripts which *themselves* can be hundreds of lines of code — and for which you may be rapidly iterating — so having the code sourced *externally*, in a single location, makes the most sense.

[... Pause ...]

Let’s take a look at both types of checks in greater detail.

[Advance]



Internal Checks

Mac Health Check

- Apple Push Notification service

```
apnsCheck=$( command log show --last 24h --predicate 'subsystem == "com.apple.ManagedClient"
&& (eventMessage CONTAINS[c] "Received HTTP response (200) [Acknowledged" || eventMessage
CONTAINS[c] "Received HTTP response (200) [NotNow")' | tail -1 | cut -d '.' -f 1 )
```

- Available Software Updates

```
mdmClientAvailableOSUpdates=$( /usr/libexec/mdmclient AvailableOSUpdates )
```

- Firewall

```
firewallCheck=$( /usr/libexec/ApplicationFirewall/socketfilterfw —getglobalstate )
```

- System Integrity Protection

```
sipCheck=$( csrutil status )
```

- Uptime

```
lastBootTime=$( sysctl kern.boottime )
```

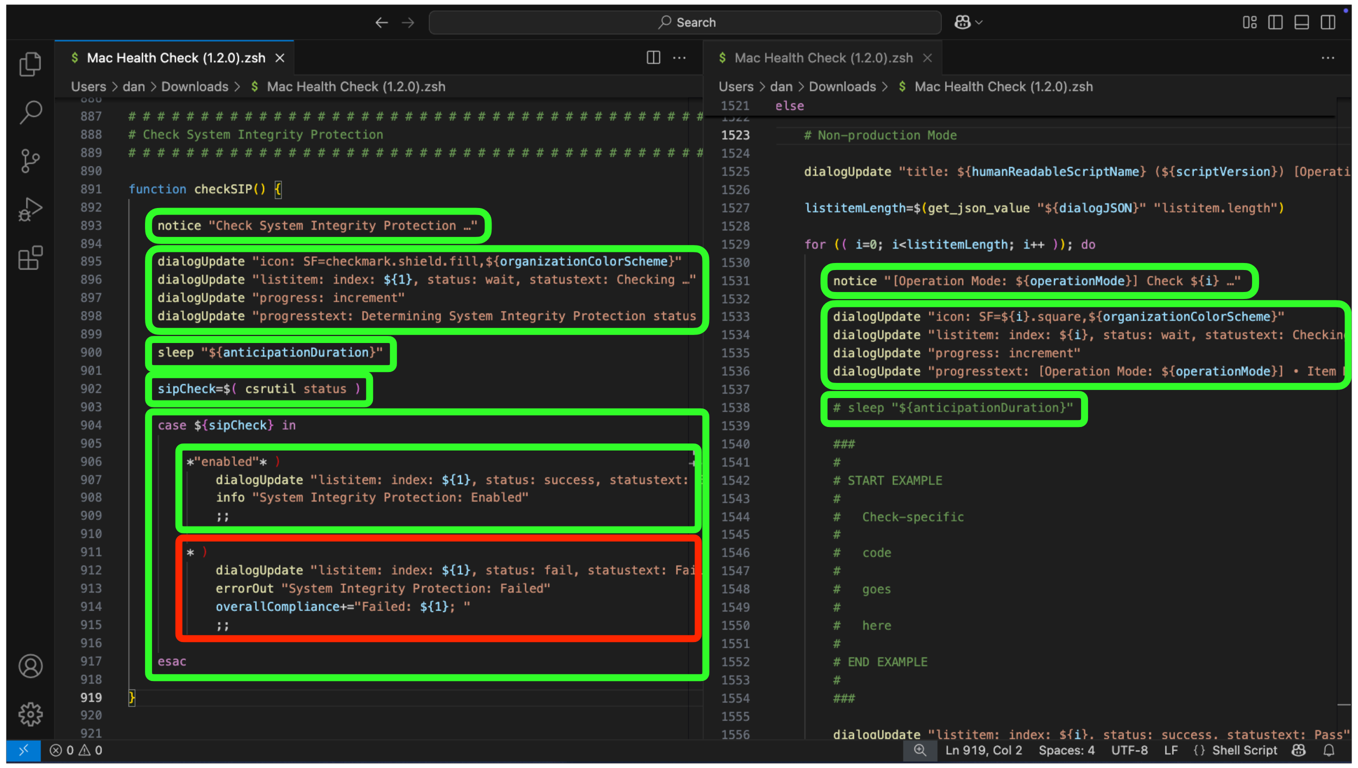
Internal checks leverage built-in macOS binaries and then analyze the results to determine if the Mac “passes” or “fails” a specific check.

[… Pause for three seconds …]

As just a random example, let’s take a deeper-dive into the check for System Integrity Protection.

[… Pause …]

[Advance]



```
887  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
888  # Check System Integrity Protection
889  # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # # #
890
891  function checkSIP() {
892
893      notice "Check System Integrity Protection …"
894
895      dialogUpdate "icon: SF=checkmark.shield.fill,${organizationColorScheme}"
896      dialogUpdate "listitem: index: ${1}, status: wait, statustext: Checking …"
897      dialogUpdate "progress: increment"
898      dialogUpdate "progresstext: Determining System Integrity Protection status
899
900      sleep "${anticipationDuration}"
901
902      sipCheck=$( csrutil status )
903
904      case ${sipCheck} in
905
906          *"enabled"* )
907              dialogUpdate "listitem: index: ${1}, status: success, statustext:
908              info "System Integrity Protection: Enabled"
909              ;;
910
911          * )
912              dialogUpdate "listitem: index: ${1}, status: fail, statustext: Fai
913              errorOut "System Integrity Protection: Failed"
914              overallCompliance+="Failed: ${1}; "
915              ;;
916
917      esac
918
919  }
920
921
```

```
1521  else
1523      # Non-production Mode
1524
1525      dialogUpdate "title: ${humanReadableScriptName} (${scriptVersion}) [Operati
1526
1527      listitemLength=$(get_json_value "${dialogJSON}" "listitem.length")
1528
1529      for (( i=0; i<listitemLength; i++ )); do
1530
1531          notice "[Operation Mode: ${operationMode}] Check ${i} …"
1532
1533          dialogUpdate "icon: SF=${i}.square,${organizationColorScheme}"
1534          dialogUpdate "listitem: index: ${i}, status: wait, statustext: Checkin
1535          dialogUpdate "progress: increment"
1536          dialogUpdate "progresstext: [Operation Mode: ${operationMode}] • Item
1537
1538          # sleep "${anticipationDuration}"
1539
1540          ###
1541          #
1542          # START EXAMPLE
1543          #
1544          #   Check-specific
1545          #
1546          #   code
1547          #
1548          #   goes
1549          #
1550          #   here
1551          #
1552          # END EXAMPLE
1553          #
1554          ###
1555
1556          dialogUpdate "listitem: index: ${i}, status: success, statustext: Pass"
```

Here's a deeper-dive into the internal check for System Integrity Protection.

[… Pause …]

On the left, we've navigated to the "checkSIP" function, starting on Line 887.

On the right, we've scrolled down to the "for loop" for ***non-production*** mode, starting on Line 1,523.

[… Pause …]

[Advance]

Each checks starts by updating the script log with "notice" and a description of the specific check.

[… Pause …]

[Advance]

Next, the "dialogUpdate" function is leveraged multiple times to change what's displayed to the end-user:

- The icon which is displayed in the upper-left
- The listitem itself (a status of “wait” will display the ever-familiar spinning pinwheel)
- The progress bar is incremented
- Last of all, the text below the progress bar is updated

[… Pause …]

[Advance]

The script is optionally paused to build anticipation.

[… Pause …]

[Advance]

A variable is set to the specific check

[… Pause …]

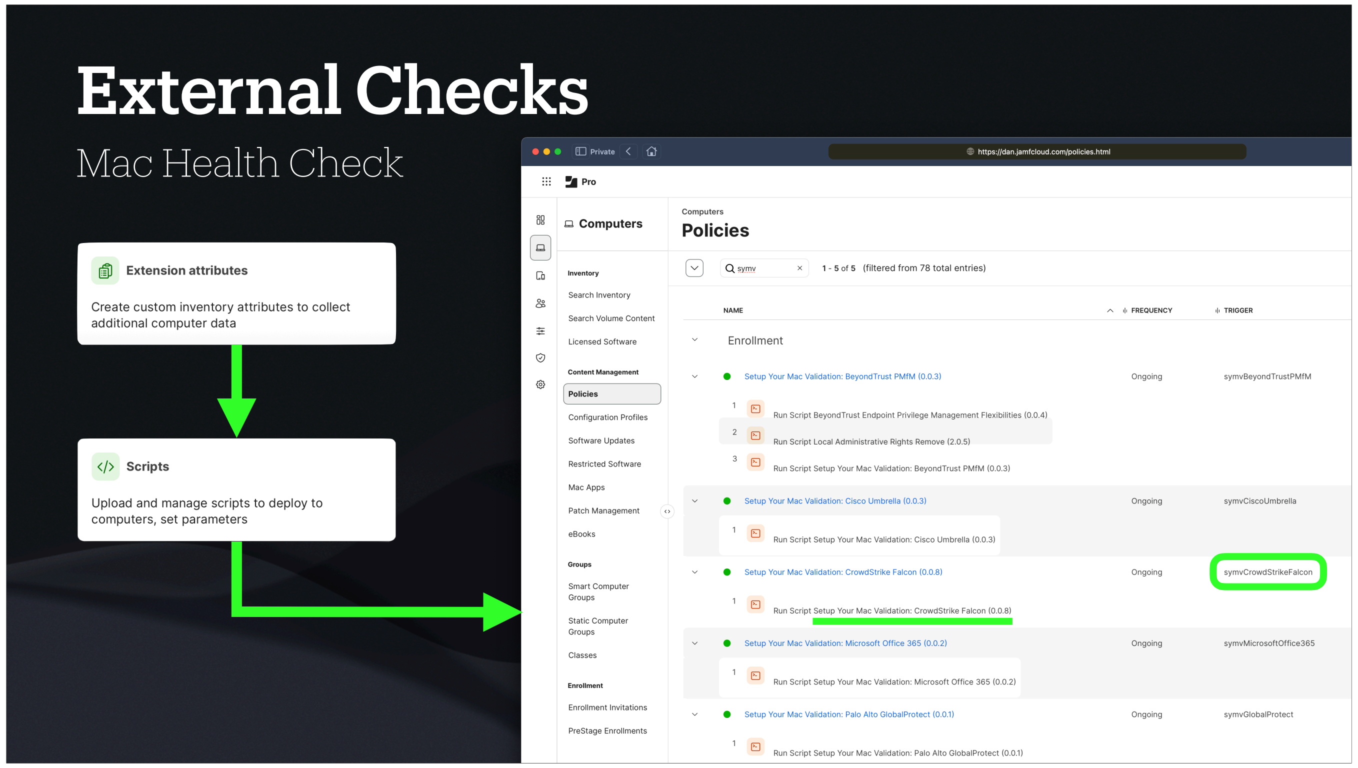
[Advance]

Then a case statement is leveraged for the various outcomes.

You’ll notice the unique “dialogUpdate” settings for each option.

[… Pause for five seconds …]

[Advance]



External Checks

Mac Health Check

Extension attributes

Create custom inventory attributes to collect additional computer data

Scripts

Upload and manage scripts to deploy to computers, set parameters

Computers

Policies

symv — 1 - 5 of 5 (filtered from 78 total entries)

NAME	FREQUENCY	TRIGGER
Enrollment		
Setup Your Mac Validation: BeyondTrust PMfM (0.0.3)	Ongoing	symvBeyondTrustPMfM
1 Run Script BeyondTrust Endpoint Privilege Management Flexibilities (0.0.4)		
2 Run Script Local Administrative Rights Remove (2.0.5)		
3 Run Script Setup Your Mac Validation: BeyondTrust PMfM (0.0.3)		
Setup Your Mac Validation: Cisco Umbrella (0.0.3)	Ongoing	symvCiscoUmbrella
1 Run Script Setup Your Mac Validation: Cisco Umbrella (0.0.3)		
Setup Your Mac Validation: CrowdStrike Falcon (0.0.8)	Ongoing	symvCrowdStrikeFalcon
1 Run Script Setup Your Mac Validation: CrowdStrike Falcon (0.0.8)		
Setup Your Mac Validation: Microsoft Office 365 (0.0.2)	Ongoing	symvMicrosoftOffice365
1 Run Script Setup Your Mac Validation: Microsoft Office 365 (0.0.2)		
Setup Your Mac Validation: Palo Alto GlobalProtect (0.0.1)	Ongoing	symvGlobalProtect
1 Run Script Setup Your Mac Validation: Palo Alto GlobalProtect (0.0.1)		

External checks leverage code you've already written, simply added to a **new,** single-script Jamf Pro policy.

[… Pause …]

[Advance]

The code of an existing Extension Attribute …

[Advance]

… is saved as a Script in your Jamf Pro server.

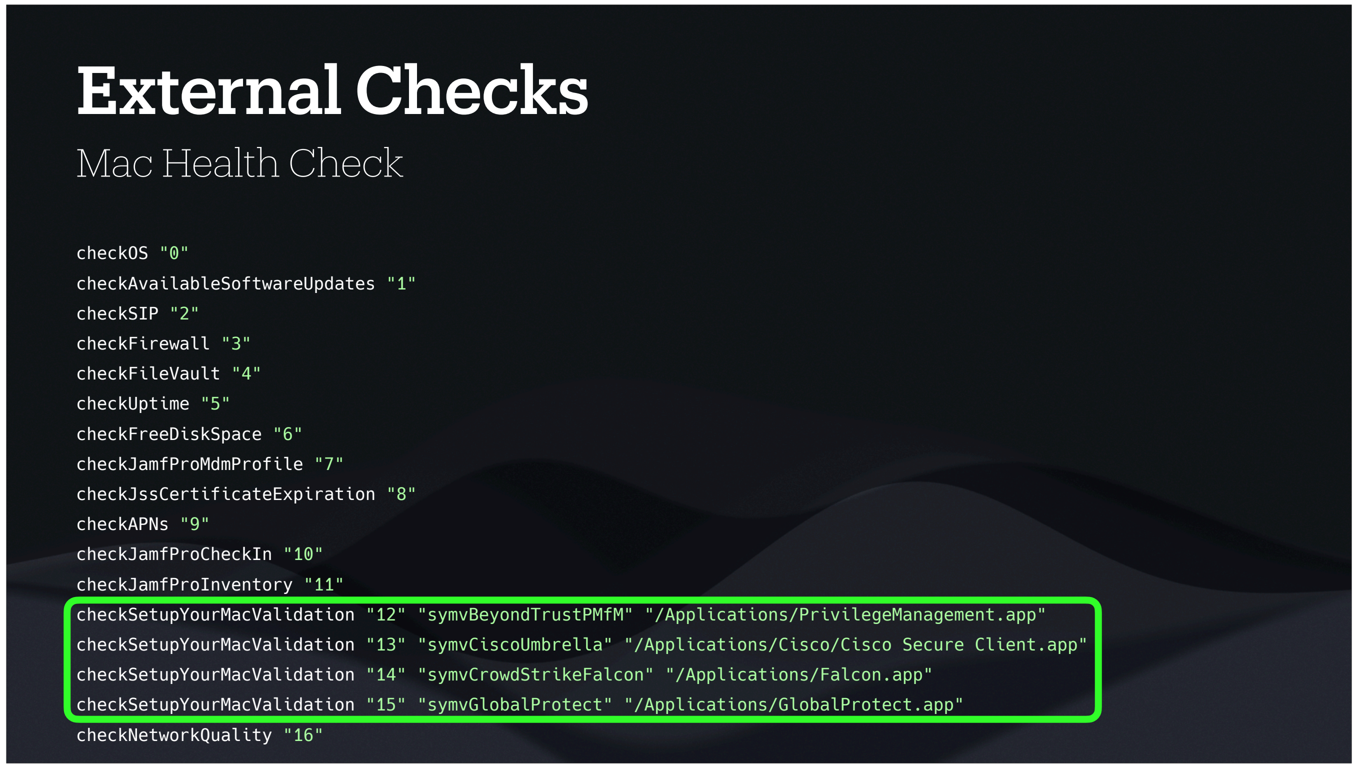
[… Pause …]

[Advance]

This script is then added to a simple Jamf Pro policy, with a custom Trigger.

[… Pause for five seconds …]

[Advance]



External Checks

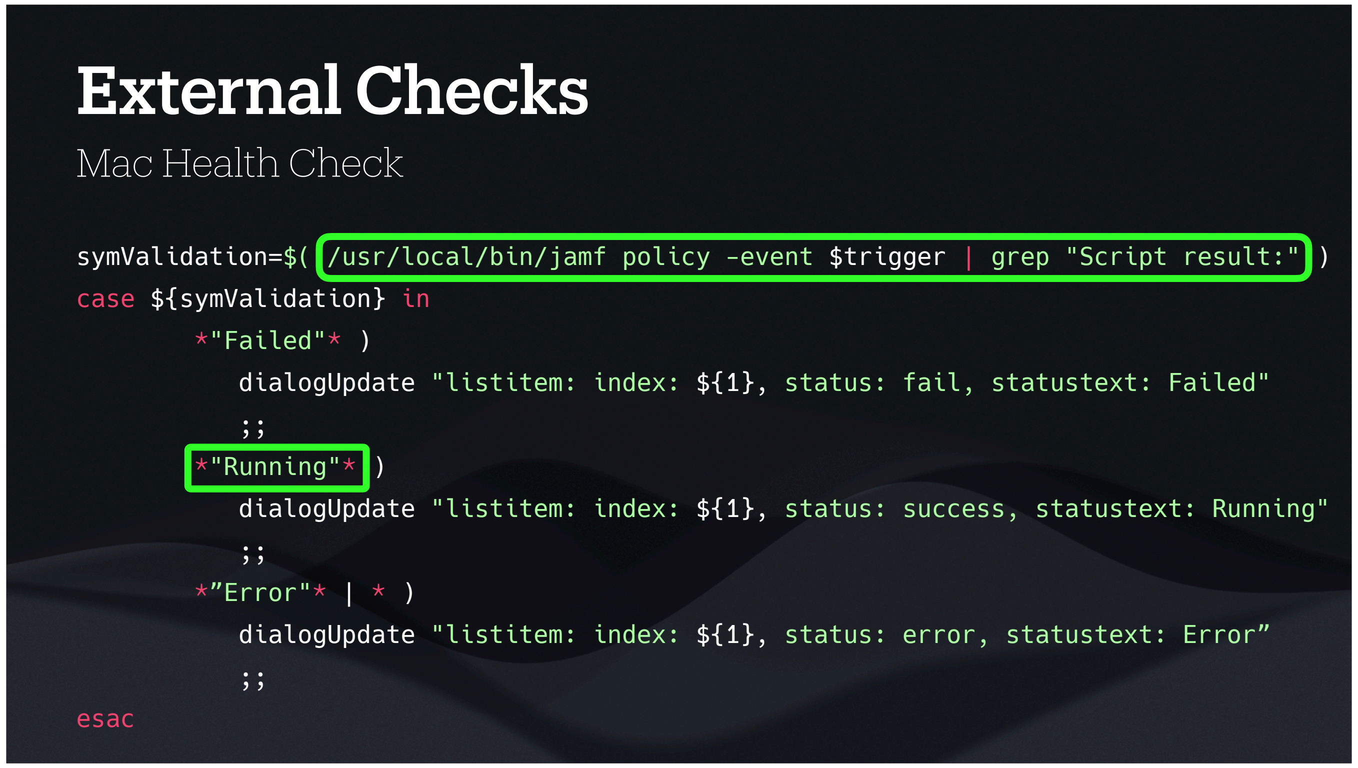
Mac Health Check

```
checkOS "0"
checkAvailableSoftwareUpdates "1"
checkSIP "2"
checkFirewall "3"
checkFileVault "4"
checkUptime "5"
checkFreeDiskSpace "6"
checkJamfProMdmProfile "7"
checkJssCertificateExpiration "8"
checkAPNs "9"
checkJamfProCheckIn "10"
checkJamfProInventory "11"
checkSetupYourMacValidation "12" "symvBeyondTrustPMfM" "/Applications/PrivilegeManagement.app"
checkSetupYourMacValidation "13" "symvCiscoUmbrella" "/Applications/Cisco/Cisco Secure Client.app"
checkSetupYourMacValidation "14" "symvCrowdStrikeFalcon" "/Applications/Falcon.app"
checkSetupYourMacValidation "15" "symvGlobalProtect" "/Applications/GlobalProtect.app"
checkNetworkQuality "16"
```

This custom Trigger — and the path to the app itself, so that its icon can be displayed to the end-user — is then specified in Mac Health Check.

[… Pause for five seconds …]

[Advance]



External Checks

Mac Health Check

```
symValidation=$( /usr/local/bin/jamf policy -event $trigger | grep "Script result:" )
case ${symValidation} in
        *"Failed"* )
           dialogUpdate "listitem: index: ${1}, status: fail, statustext: Failed"
           ;;
        *"Running"* )
           dialogUpdate "listitem: index: ${1}, status: success, statustext: Running"
           ;;
        *”Error"* | * )
           dialogUpdate "listitem: index: ${1}, status: error, statustext: Error”
           ;;
esac
```

When the policy successfully executes, the returned output must include the keyword “Running,” meaning the service which you’re validating is running.

[… Pause …]

Several validations are included in the Setup Your Mac GitHub repo, which is cross-referenced on my blog post.

[… Pause …]

[Advance]

Adaptation: UI

Let’s look at a few demos of adapting the user interface …

[Advance]

Name & Color

Here’s a demo of modifying the name and color.

[Advance]

We’ll start by visiting the GitHub repo and downloading a fresh copy.

[… Pause for Terminal to be displayed …]

In a previously elevated Terminal window, we’ll execute the script without modification.

We’ll observe that the latest production version of swiftDialog is automatically installed. (Thanks, Adam!)

Why is the title “Computer Compliance” ?

[… Pause for script to complete …]

Let’s rename this script “Mac Health Check.”

[… Pause for Terminal to be displayed …]

OK, that’s better.

[… Pause for VS Code to be displayed …]

Let’s fix the colors.

[… Pause for dialog to be displayed …]

Much better.

[… Pause for fade out …]

So, that was a demo of modifying the name and colors used in Mac Health Check.

[Advance]

List Items

Here's a demo of modifying the list items.

In Visual Studio Code, we'll open the same file in both the left and right windows.

[Advance]

On the left, we've scrolled to the "listitem" block, which controls the information displayed to the end-user.

On the right, we've scrolled to the "Production Mode" section, which shows the order in which the various checks are executed.

Let's change some of the "eye candy" for the displayed number style …

[… Pause for Terminal to be displayed …]

… and we'll run the script.

OK, OK … not bad.

Let's try another option …

[… Pause for Terminal to be displayed …]

… I don't know … circles … squares … filled … empty …

[… Pause for VS Code to be displayed …]

Let’s get dangerous and change the order of operations.

You could convince me that having the Network Quality Test executed first makes sense; I’ll then know the user’s bandwidth for the rest of the support call.

However, now it get tedious.

[… Pause for error to be displayed in Terminal …]

ARGH!

JSON import failed?!?

What’d I do?

Let’s take a look.

[… Pause for VS Code to be displayed …]

Oh! I see … it’s always the line endings.

I’ve got a good feeling about this attempt …

[… Pause for Terminal to be displayed …]

I’ve got a good feeling about this attempt …

… Whew!

OK, we just need to fix the displayed numbers.

[… Pause …]

So, that was a demo of modifying the list items.

[Advance]

Here’s a resource …

[Advance]

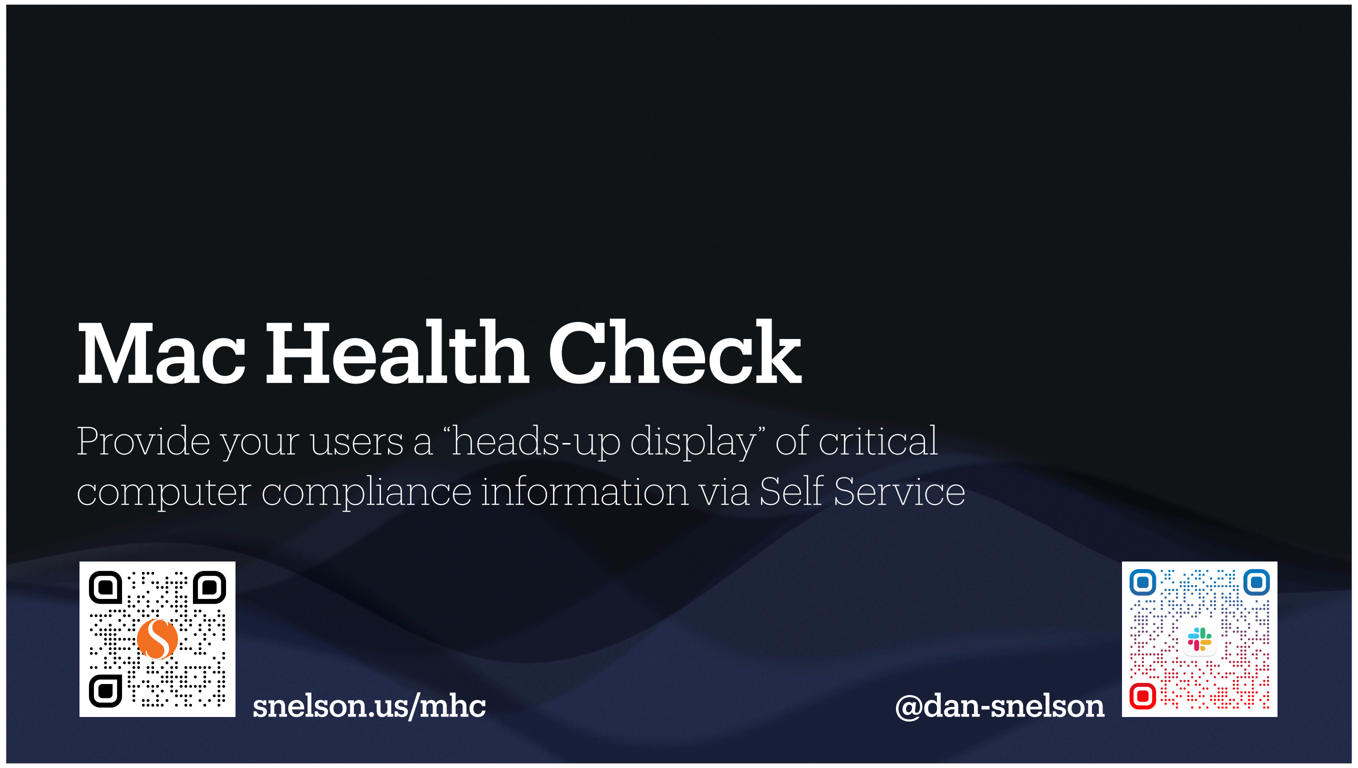
We’ll, that’s it!

[… Pause …]

What questions can I answer?

[… Pause …]

[Advance]



Mac Health Check

Provide your users a "heads-up display" of critical computer compliance information via Self Service

snelson.us/mhc

@dan-snelson

Thank you!

[… Pause …]

[Advance]