jamf Nation - London Roadshow

Sachin Parmar
Workplace Technology Manager

17 May 2018

JUST EAT

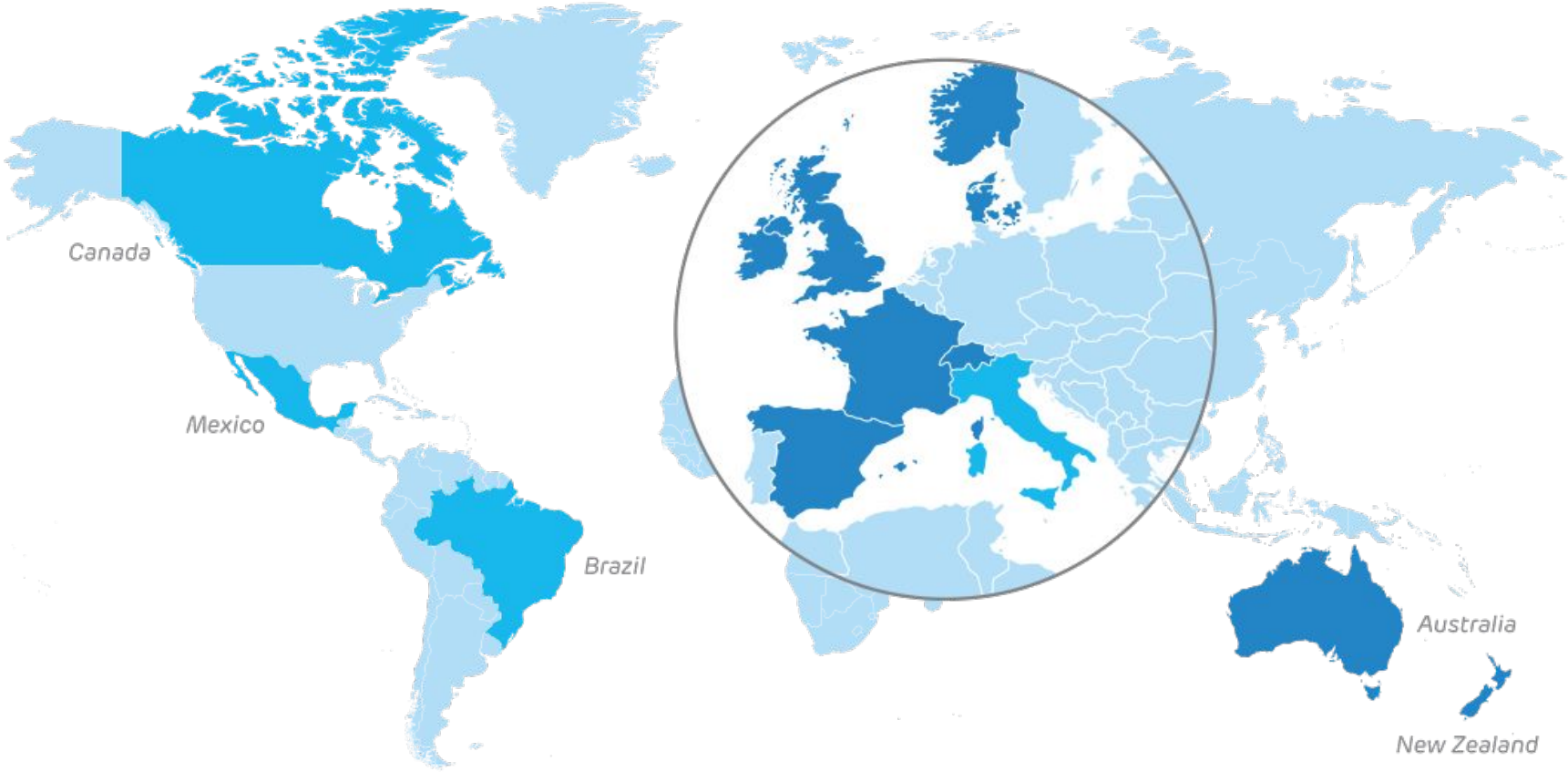
About Just Eat



Our vision

Creating the world's greatest food community





JUST EAT

About Me

JUST EAT

About Me

Sachin Parmar

- 8+ years professionally in the Technology industry
- Administrator of Microsoft System Centre Configuration Manager 2007, 2012
- Administrator of JAMF Software Casper Suite for 5 years over two companies:

  DMGT – Daily Mail and General Trust Plc
  JUST EAT

- Blog: www.sachinparmarblog.com
- LinkedIn: **Sachin Parmar** (https://uk.linkedin.com/in/sachin-parmar-12500360)
- Email: sachin.parmar@just-eat.com
- Slack: **@sachinparmar**
- JAMF Nation: https://jamfnation.jamfsoftware.com/viewProfile.html?userID=28787

JUST EAT

Our Journey

JUST EAT

Our Journey

Jan 2015

150 registered Mac's bound to the domain with Centrify

May 2015

Sachin Parmar joins Just Eat

Jun 2015

Gathered requirements on current processes and how technicians build Mac's, all manual process.

Jul 2015

Trial jamf Pro requested from jamf Software

JUST EAT

Aug - Sep 2015

Submitted Business Case for JAMF Software Casper Suite for approval to the Head of Technology

Nov 2015

Business Case was approved by Head of Technology and jamf Pro was purchased via reseller

Jan 2016

2 Day Jumpstart Booked with jamf to come on site and configure, at the time, Casper Suite

Feb 2016

Jumpstart completed and implementation began

JUST EAT

Feb - Mar 2016

180 policies and configurations built inside Casper Suite, to help automate imaging Mac's.

Mar 2016

Local distribution points placed in 8 locations over 6 countries

Mar 2016

Casper Suite rolled out to 400 Mac's, and all new Macs are imaged with Casper Suite

Mar - Sep 2016

All Centrify Macs Migrated to Casper Suite, Casper Suite now BAU

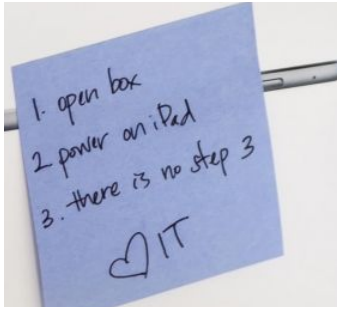
JUST EAT

Sep 2016

Day 1 - Deployment of macOS Sierra

JUST EAT

Our Future Journey - 18 Months Ago



TBD 2016/2017 – Casper Suite to manage a full Zero Touch Deployment of all Apple Devices to End Users

Oct 2016 – Merge our existing Apple Device Enrolment Program and Volume Purchase Program into Casper Suite

Oct 2016 – Implement ADFS Single Sign 2.0 with Azure Multi-Factor Authentication into Casper Suite

JUST EAT

Our Future Journey - Update

Oct 2016

Ten global entities enrolled and to Apple DEP and VPP programs, which were then configured into Jamf Pro

Oct 2016

Implemented ADFS Single Sign 2.0 with Azure Multi-Factor Authentication into Jamf Pro, access now controlled via AD security group

Dec - Jan 2017

Implemented NoMAD across the business to handle Kerberos tickets for mobile user accounts, trialing users with local accounts and NoMAD combination.

Dec - Jan 2017

Open source real time dashboard created using the jamf Pro API, reporting basic stats which is presented on a TV

JUST EAT

Feb 2017

Build process changed. User accounts are now local accounts in conjunction with NoMAD, machines are bound to AD

Feb - Mar 2017

All Macs are EAP-TLS 802.1x compliant in preparation for securing down the network. Machines require certificates for network authentication.

Mar - Oct 2017

100 x iOS Devices migrated from another MDM provider into Jamf Pro

Jul - Oct 2017

All Macs/iOS Devices now purchased through DEP via Apple or authorised resellers in preparation for macOS High Sierra. Zero Touch Build Process now live. NetBoot build process turned off, also implemented SplashBuddy alongside jamf Pro.

JUST EAT

Apr 2018

Rolled out G Suite for Enterprise iOS MDM Management enforcement security change in conjunction with jamf Pro MDM which allows Google iOS Apps to be approved by Google Administrators regardless of device ownership.

JUST EAT

Our Infrastructure - Basic Overview

JUST EAT

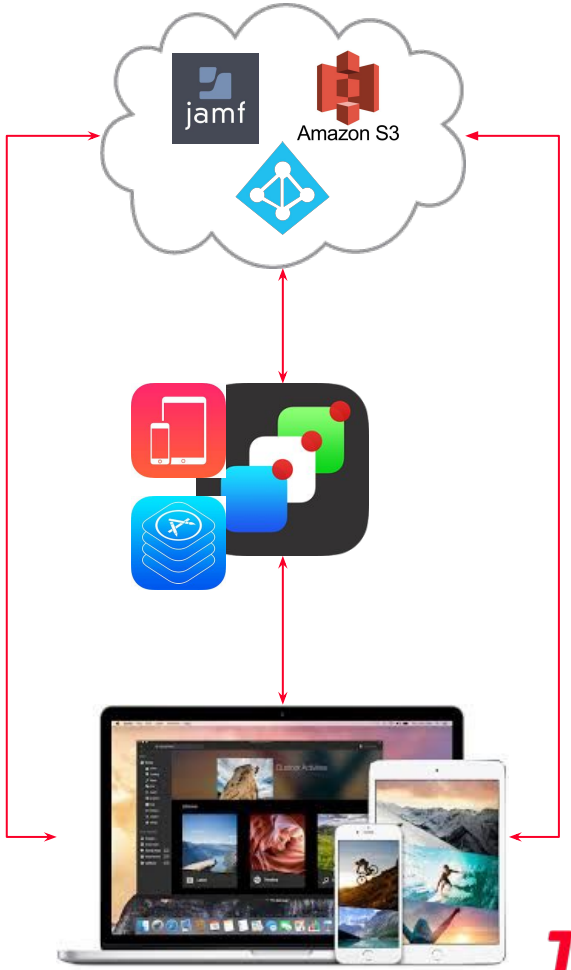
1. jamf Pro in the Cloud

2. Amazon S3 Bucket -
Cloud Distribution Point

3. Microsoft ADFS

4. DEP/VPP via Apple APNs

5. Apple Devices

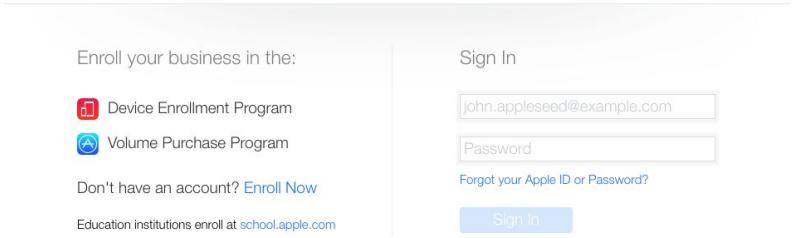


JUST EAT

Device Enrollment Program Process

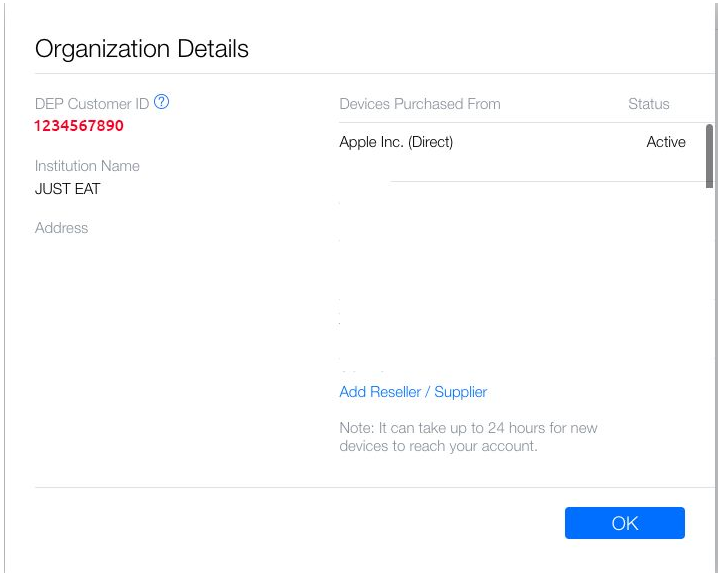
JUST EAT

Register to Apple DEP



First, Register the company to Apple Device Enrollment Program and wait to be approved by Apple.
https://deploy.apple.com

Inform your Suppliers



After approval, you'll receive your DEP Customer ID, inform your suppliers of your ID and obtain their DEP Reseller ID and add them into the DEP Portal

How To: https://sachinparmarblog.com/automated-managed-mac-estate-dep-splashbuddy-part-1

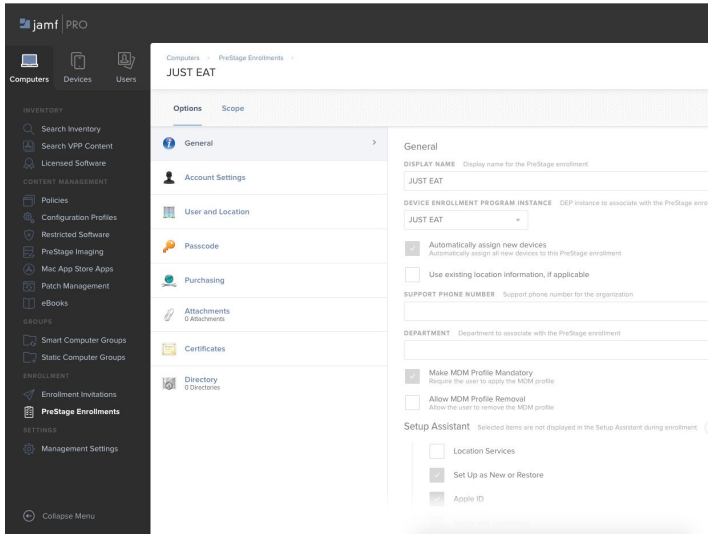
JUST EAT

Link DEP + jamf Pro

Next, you'll need to link your jamf Pro MDM with Apple DEP Portal.

Download the Public Key from your jamf Pro and upload this in Apple DEP Portal, a Server Key will be generated and this can be uploaded to jamf Pro and now both will be talking to each other

Pre-Stage Enrollment



Then a Pre-Stage enrollment needs to be created which will control the behaviour of any new device that is enrolled via DEP. For example, the Setup Assistant can be configured not to show Touch ID, plus more.

How To: https://sachinparmarblog.com/automated-managed-mac-estate-dep-splashbuddy-part-1

JUST EAT

Make the Build Process

Next, a set of policies need to be created that will configure the machine to the businesses specification all to run with a trigger of Enrollment Complete, as these tasks to run after the Pre-Stage enrollment has completed.

How To: https://sachinparmarblog.com/automated-managed-mac-estate-dep-splashbuddy-part-1

JUST EAT

SplashBuddy

JUST EAT

SplashBuddy

So what is SplashBuddy?

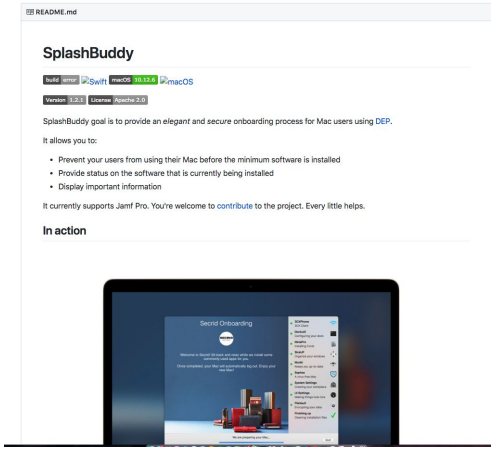
SplashBuddy's goal is to provide an elegant and secure onboarding process for Mac users using DEP.

It allows you to:

- Prevent your users from using their Mac before the minimum software is installed
- Provide status on the software that is currently being installed
- Display important information
- Brand your onboarding experience

JUST EAT

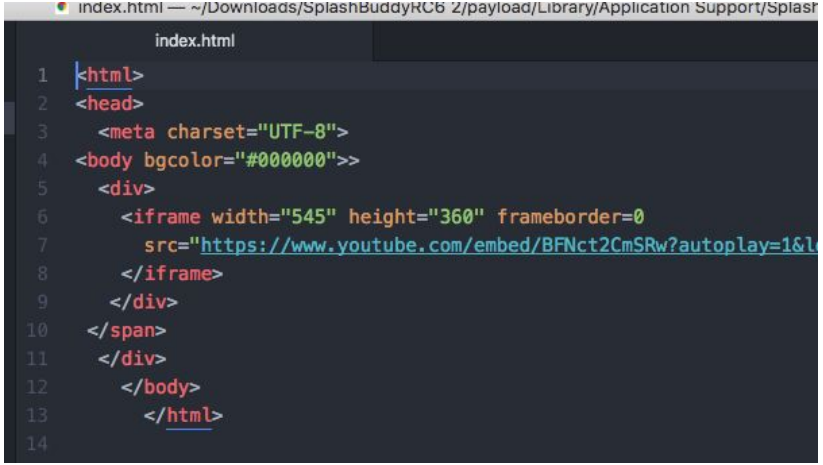
Get SplashBuddy



Firstly, head over to the SplashBuddy GitHub page and download the latest stable release.

https://github.com/Shufflepuck/SplashBuddy/releases

Modify index.html

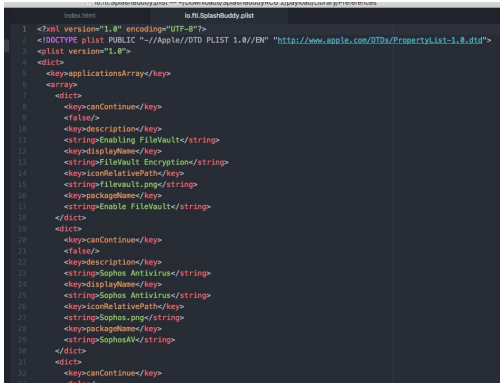


Next, you'll need to dive into the file structure and into the presentation.bundle, choose the relevant language, i.e. en.lproj and modify the index.html file located here, this controls what is displayed on the left hand pane.

How To: https://sachinparmarblog.com/automated-managed-mac-estate-dep-splashbuddy-part-1

JUST EAT

Modify SplashBuddy Preferences



Then, the SplashBuddy preference file needs to be modified to include all the steps that are required to be visible in the right hand pane, include all icon images.

This is where the behaviour is controlled for example the user can not continue to their machine until a step is actually completed.

More info on the preferences can be found, https://github.com/Shufflepuck/SplashBuddy/wiki/20---Components

File Structure

Name	Date Modified	Size	Kind
build_pkg.sh	2 Jul 2017 at 13:24	648 bytes	Shell Script
payload	13 Nov 2017 at 15:19	--	Folder
Library	13 Nov 2017 at 15:19	--	Folder
Application Support	13 Nov 2017 at 15:20	--	Folder
SplashBuddy	13 Nov 2017 at 15:20	--	Folder
AnyConnect.png	30 Jun 2017 at 12:55	46 KB	PNG image
chrome.png	31 Mar 2017 at 12:43	16 KB	PNG image
filevault.png	30 Jun 2017 at 09:35	94 KB	PNG image
JustEat.jpg	30 Jun 2017 at 09:56	9 KB	JPEG image
NoMAD.png	30 Jun 2017 at 12:57	81 KB	PNG image
postInstall.sh	3 Jul 2017 at 13:47	97 bytes	Shell Script
presentation.bundle	27 Sep 2017 at 14:12	821 KB	Bundle
Sophos.png	30 Jun 2017 at 09:38	59 KB	PNG image
SplashBuddy	2 Jul 2017 at 21:33	12 MB	Application
SplashBuddy.launch.sh	31 May 2017 at 10:11	1 KB	Shell Script
TeamViewer.png	30 Jun 2017 at 12:59	147 KB	PNG image
LaunchAgents	31 Mar 2017 at 12:43	--	Folder
io.fti.SplashBuddy.launch.plist	31 Mar 2017 at 12:43	493 bytes	Property List
Preferences	3 Jul 2017 at 11:37	--	Folder
io.fti.SplashBuddy.plist	25 Oct 2017 at 15:58	3 KB	Property List
scripts	31 Mar 2017 at 12:43	--	Folder
postinstall	31 Mar 2017 at 12:43	82 bytes	Unix executable

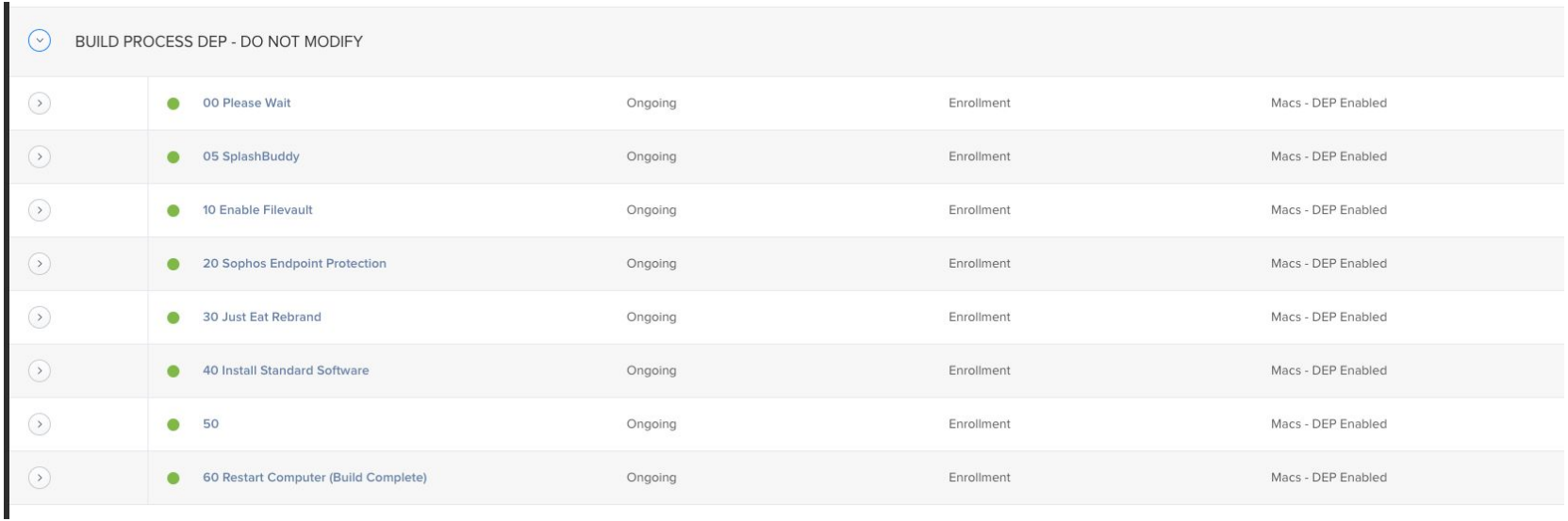
Once ready the file structure should look a little similar to the above, where icon images are placed under the SplashBuddy folder, etc.

When ready, run the **build_pkg.sh** script to create a PKG Installer, import this into jamf using jamf Admin

How To: https://sachinparmarblog.com/automated-managed-mac-estate-dep-splashbuddy-part-1

JUST EAT

Change Build Process



BUILD PROCESS DEP - DO NOT MODIFY

00 Please Wait	Ongoing	Enrollment	Macs - DEP Enabled
05 SplashBuddy	Ongoing	Enrollment	Macs - DEP Enabled
10 Enable Filevault	Ongoing	Enrollment	Macs - DEP Enabled
20 Sophos Endpoint Protection	Ongoing	Enrollment	Macs - DEP Enabled
30 Just Eat Rebrand	Ongoing	Enrollment	Macs - DEP Enabled
40 Install Standard Software	Ongoing	Enrollment	Macs - DEP Enabled
50	Ongoing	Enrollment	Macs - DEP Enabled
60 Restart Computer (Build Complete)	Ongoing	Enrollment	Macs - DEP Enabled

Finally, change the build process so it runs in the order set in your SplashBuddy preferences file.

Ensure to run with a Trigger of “Enrollment” and targeted to a Smart Group that contains the device registered via Pre-Stage enrollment.

How To: https://sachinparmarblog.com/automated-managed-mac-estate-dep-splashbuddy-part-1

JUST EAT

Final Result

JUST EAT

EAP-TLS

JUST EAT

EAP-TLS

802.1x Certificate Based Authentication

“Certificate-based authentication is the use of a Digital Certificate to identify a user, machine, or device before granting access to a resource, network, application, etc.” - www.globalsign.com

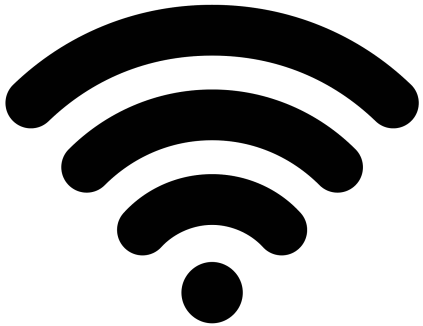
In an attempt to be an always evolving modern technology company Just Eat have adopted to to use Wi-Fi in all offices around the world. Although visionary as this is, it certainly comes with specific challenges in order to secure the network, as most companies have opted to adopt an 802.1x user or machine certificate style based network authentication going forward.

The ultimate aim was this was make this easy enough for the end user to be able to run themselves ideally with one click.

So how did we get around this?...

JUST EAT

Hidden Wi-Fi SSID Setup



Firstly, working with the network team we setup a hidden Wi-Fi SSID which can only be authenticated with one service account.

This service account has line of sight to Active Directory and the Certificate Enrollment Server.

Configuration Profile - Hidden SSID

A Configuration Profile created in jamf Pro to allow authentication and auto-join of the hidden SSID.

How To: https://sachinparmarblog.com/wireless-802-1x-eap-tls-on-mac-os-x/

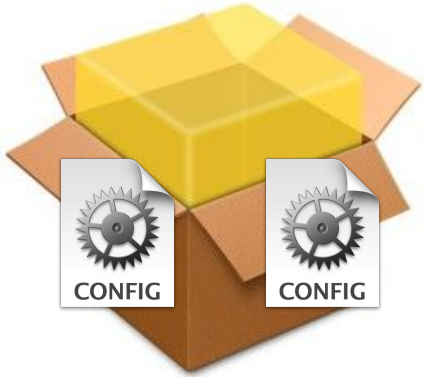
JUST EAT

Configuration Profile - EAP-TLS

Next, we made a secondary configuration profile in jamf Pro which contained the relevant configuration for requesting certificates from the AD certificate server.

This Configuration Profile also includes the Root Certificate and any Intermediate Certificates we require.

Configuration Profile - Hidden SSID



Then downloaded both mobile configuration profiles and placed then in an accessible location on the Mac, i.e. /Users/Shared and packaged this via jamf Composer.

Once packaged uploaded this to our jamf Pro using jamf Admin.

How To: https://sachinparmarblog.com/wireless-802-1x-eap-tls-on-mac-os-x/

JUST EAT

Active Directory Bind Policy

Then created a jamf Pro policy which would complete and Active Directory bind, with an OU path and AD Service Account configured.

This policy was then given a custom trigger

How To: https://sachinparmarblog.com/wireless-802-1x-eap-tls-on-mac-os-x/

JUST EAT

Script

```
#!/bin/sh
#PEAP PROFILE ACCOUNT
/usr/bin/profiles -I -F /Users/Shared/HiddenSSID.mobileconfig
sleep 5
#NETWORK CHECK
JustEatNetworkCheck=0
while [ $JustEatNetworkCheck -ne 1 ]; do
        ping -q -c 4 just-eat.com
        if [ $? -eq  0 ]; then
                echo "Just Eat Network Found";
                JustEatNetworkCheck=1;
                #802.1X Profile
                loggedInUser=`python -c 'from SystemConfiguration
                import SCDynamicStoreCopyConsoleUser; import sys; username =
                (SCDynamicStoreCopyConsoleUser(None, None, None) or [None])[0];
                username = [username,""][username in [u"loginwindow", None, u""]]; sys.stdout.write(username + "\n");'`
                echo $loggedInUser
                sleep 10
                sudo jamf setComputerName --useSerialNumber
                sleep 5
                sudo jamf policy -event ADBindCustomTrigger
                wait
                sudo -u $loggedInUser /usr/bin/profiles -I -F /Users/Shared/EAP-TLS.mobileconfig
                wait
        else
                echo "Just Eat Network Not Found"
                exit
        fi
done
#Remove PEAP ACCOUNT
/usr/bin/profiles -R -p [ConfigurationProfileIdentifier]
wait
sudo rm /Users/Shared/HiddenSSID.mobileconfig
sudo rm /Users/Shared/EAP-TLS.mobileconfig
```

How To: https://sachinparmarblog.com/wireless-802-1x-eap-tls-on-mac-os-x/

JUST EAT

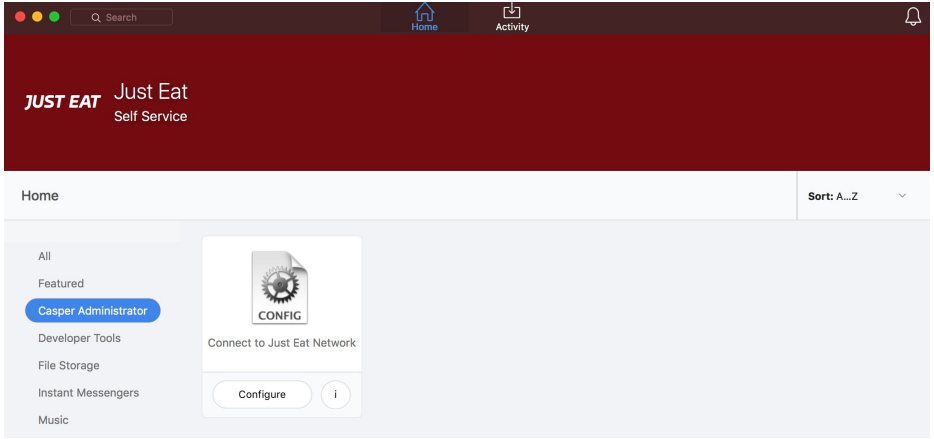
Connect to Just Eat Network Policy

Connect to Just Eat Network

Finally, we made a policy called “Connect to Just Eat Network” which takes the package we created and uploaded to jamf Pro and set that to install as a first step.

Then, we added the script we created to run after the install was completed, this then runs the script as described previously.

Finally, we made the policy available for users to run in Self Service with one-click!



How To: https://sachinparmarblog.com/wireless-802-1x-eap-tls-on-mac-os-x/

JUST EAT

Final Result

JUST EAT

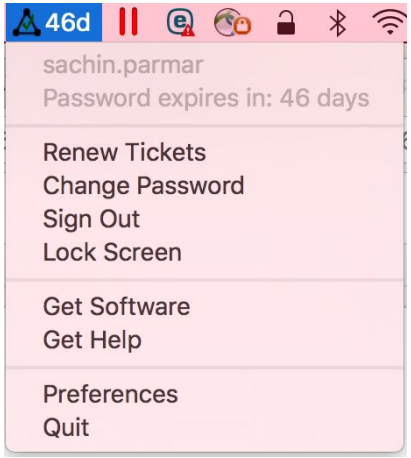
NoMAD

JUST EAT

NoMAD

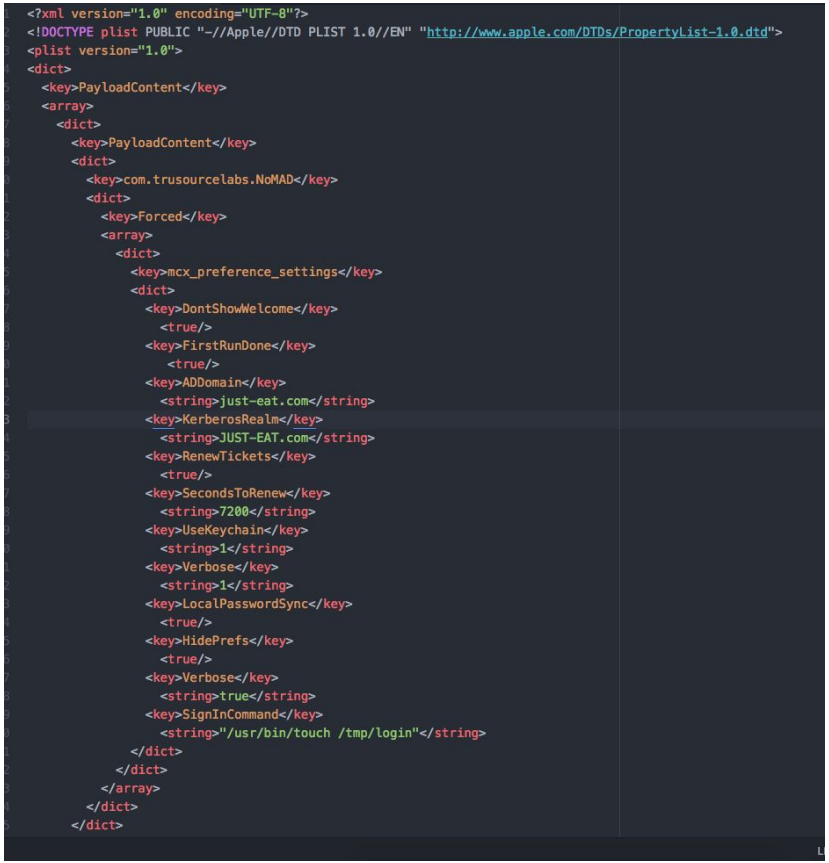
- *"While NoMAD can be a great tool to help users bound to Active Directory, its main purpose is to help move your Macs off binding to AD while still getting all of the functionality. Keep your users on local accounts and let NoMAD manage their interaction with AD by allowing them to sign in with their AD account to get Kerberos tickets and other functions without having to have a mobile account." - www.nomad.menu*

- Written by Joel Rennich and more (Mactroll)
- Available at https://nomad.menu
- macOS menu bar application
- Identity Option as indicated by Apple in the "Mac Deployment Overview" Guide (https://images.apple.com/business/resources/docs/Mac_Deployment_Overview.pdf)



JUST EAT

NoMAD

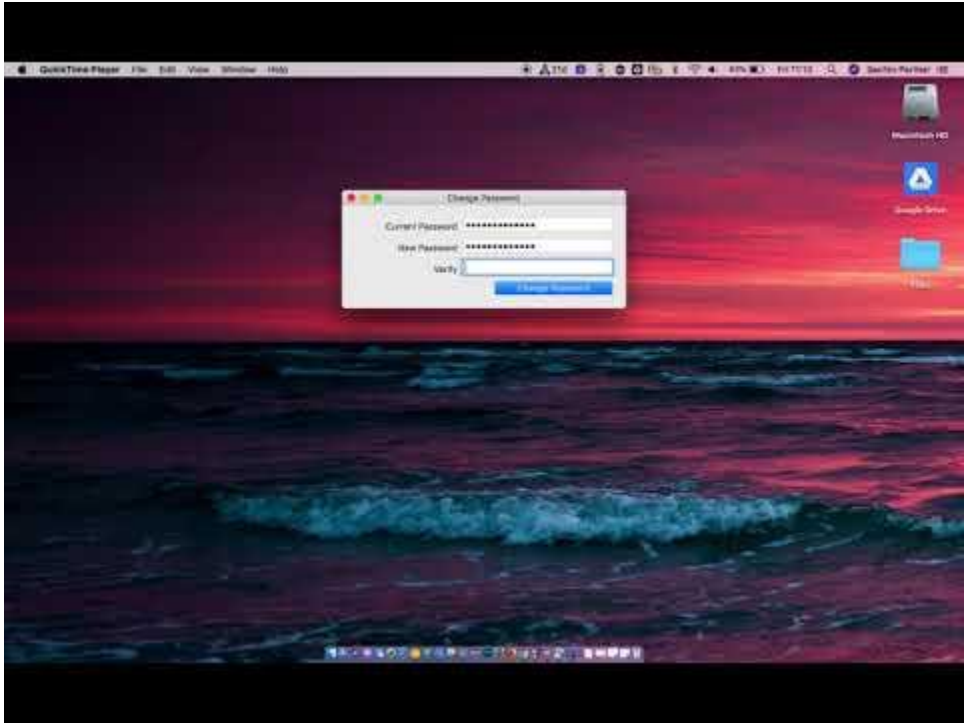


```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE plist PUBLIC "-//Apple//DTD PLIST 1.0//EN" "http://www.apple.com/DTDs/PropertyList-1.0.dtd">
<plist version="1.0">
<dict>
  <key>PayloadContent</key>
  <array>
    <dict>
      <key>PayloadContent</key>
      <dict>
        <key>com.trusourcelabs.NoMAD</key>
        <dict>
          <key>Forced</key>
          <array>
            <dict>
              <key>mcx_preference_settings</key>
              <dict>
                <key>DontShowWelcome</key>
                  <true/>
                <key>FirstRunDone</key>
                  <true/>
                <key>ADDomain</key>
                  <string>just-eat.com</string>
                <key>KerberosRealm</key>
                  <string>JUST-EAT.com</string>
                <key>RenewTickets</key>
                  <true/>
                <key>SecondsToRenew</key>
                  <string>7200</string>
                <key>UseKeychain</key>
                  <string>1</string>
                <key>Verbose</key>
                  <string>1</string>
                <key>LocalPasswordSync</key>
                  <true/>
                <key>HidePrefs</key>
                  <true/>
                <key>Verbose</key>
                  <string>true</string>
                <key>SignInCommand</key>
                  <string>"/usr/bin/touch /tmp/login"</string>
              </dict>
            </dict>
          </array>
        </dict>
      </dict>
```

- NoMAD utilises the pre-defined preferences you have set as an organisation
- Preferences and what they do are located at the following (https://nomad.menu/help-center/preferences-and-what-they-do/)
- Important to note that “LocalPasswordSync” controls the behaviour of Local Mac accounts
- Once the preferences have been defined this can be wrapped up into a MobileConfig Profile (upload custom property list option in Jamf Pro) and deployed to your machines
- Next, NoMAD application can be deployed

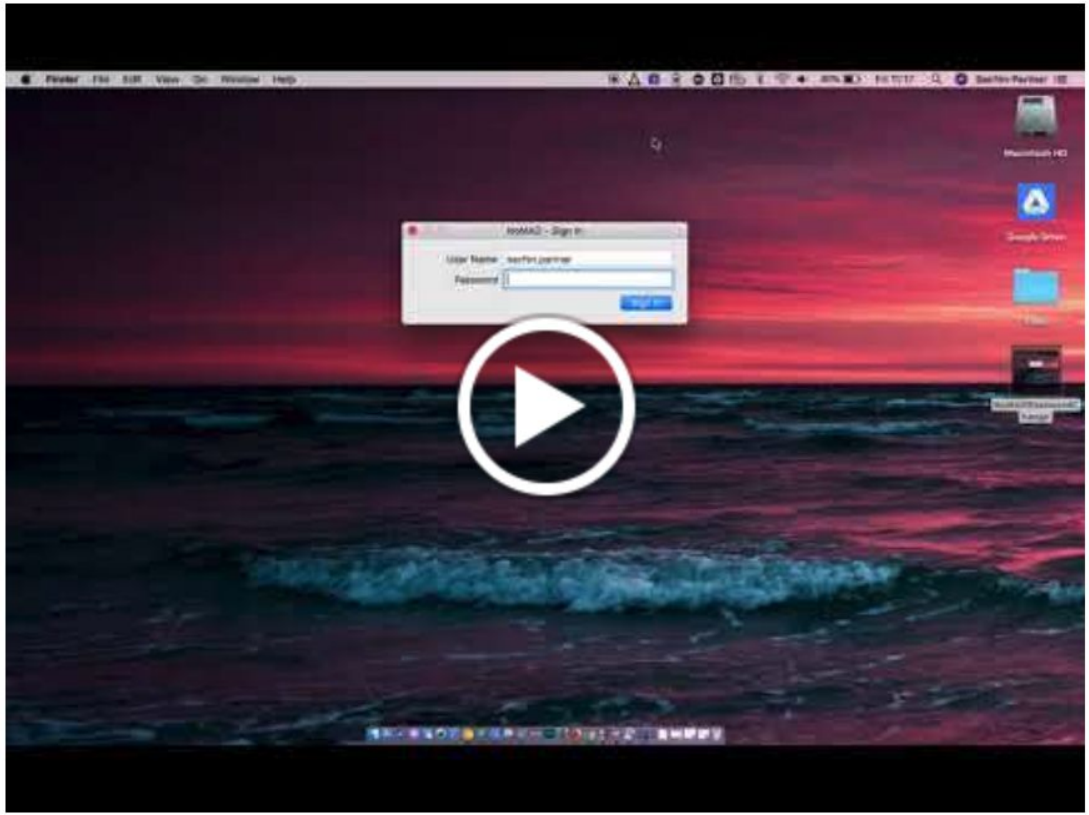
JUST EAT

NoMAD - Password Change



JUST EAT

NoMAD - *Password Change via Active Directory*

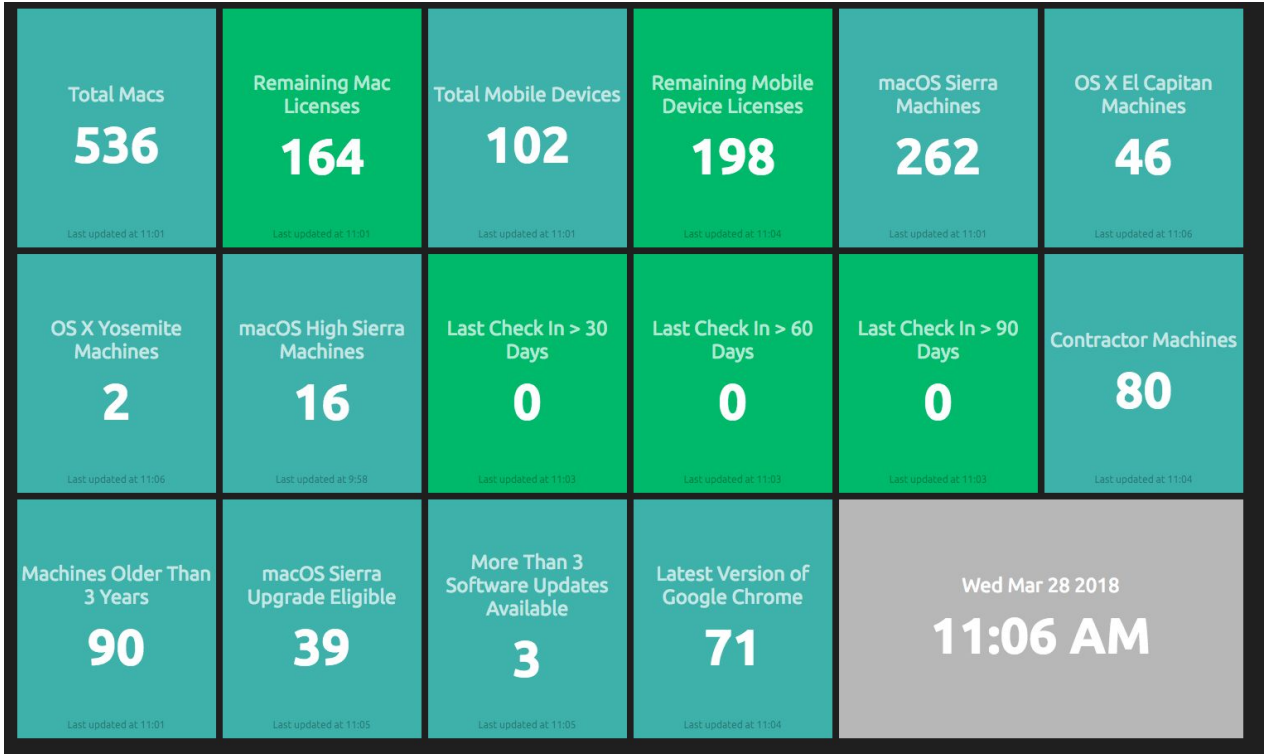


JUST EAT

Real Time Dashboard

JUST EAT

Real Time Dashboard



- By utilising the JSS API, we've managed to query our jamf Inventory Data and present it in a real time dashboard
- Dashboard uses the Dashing Framework (Open-Source) to display the information
- Clear way of identifying key areas of our Mac estate that require attention, i.e. If Mac's haven't checked back in in 90 Days there's clearly a problem and it will require investigating
- How To: https://sachinparmarblog.com/displaying-your-jamf-pro-casper-suite-stats-in-a-dashboard/

JUST EAT

Training

JUST EAT

Using a jamf Organisational Training Pass over the last 18 months, Just Eat now have:

11 x Casper Certified Technicians / Jamf 200
3 x Casper Certified Administrators / Jamf 300
2 x Casper Certified Experts / Jamf 400

JUST EAT

Questions?

JUST EAT