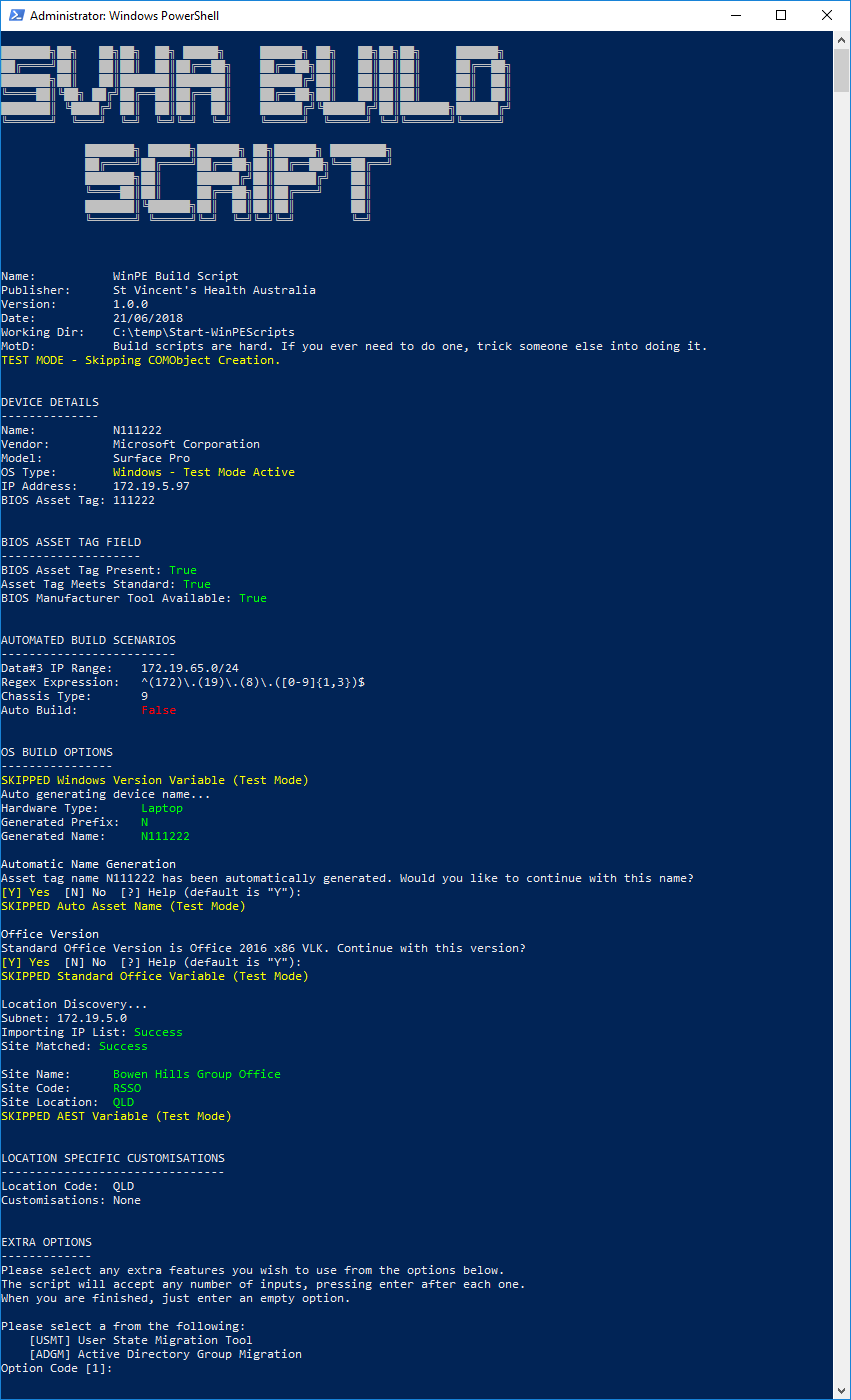
# Overview

The SVHA Build Script has been written to replace the existing UI implemented late last year. The replacement was driven by a need to perform tasks that were not possible using UI++ (or needed VBScript experience).

The replacement will allow more features without the need to tack on extra scripts such as the ability to build Surface Pro devices at Data#3 and set the asset tag for Surface Pro devices in the BIOS.

This document contains all the necessary information and tricks to using the new build script.

## Example



# But Why?

But the old UI was so pretty, why would you change to something CLI based?

The new script brings with it the ability to add a whole host of additional functionality without having to tack it on after the fact. With the existing UI I was scripting a lot of this in PowerShell but calling it based on variables passed from the UI. Doing it 100% in PowerShell allows it to all be done at once in a clean fashion and makes it easy to troubleshoot if I ever get hit by a bus.

This new script brings with it other benefits in the form of no longer needing to add a computer to an AD group for it to build! HOORAY!

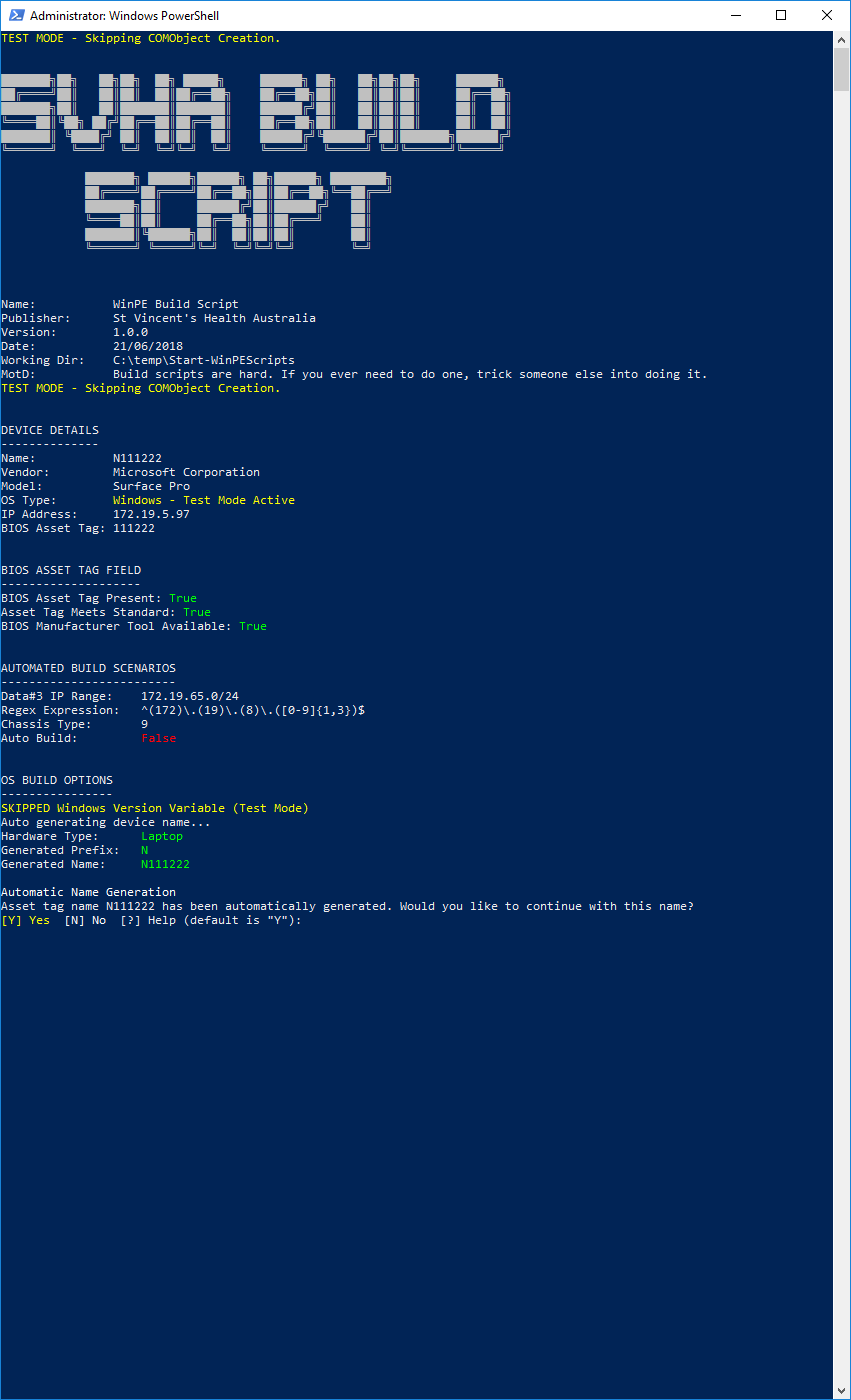
# Script Components

The script contains several distinct sections that perform different functions. Each is detailed below with descriptions of the different possibilities in those sections should you encounter any issues. If you have any issues or are unsure of a script output please check this document before asking for assistance.

The script is as automated as possible, to assist with this please make sure the asset tag is set in the BIOS where possible.

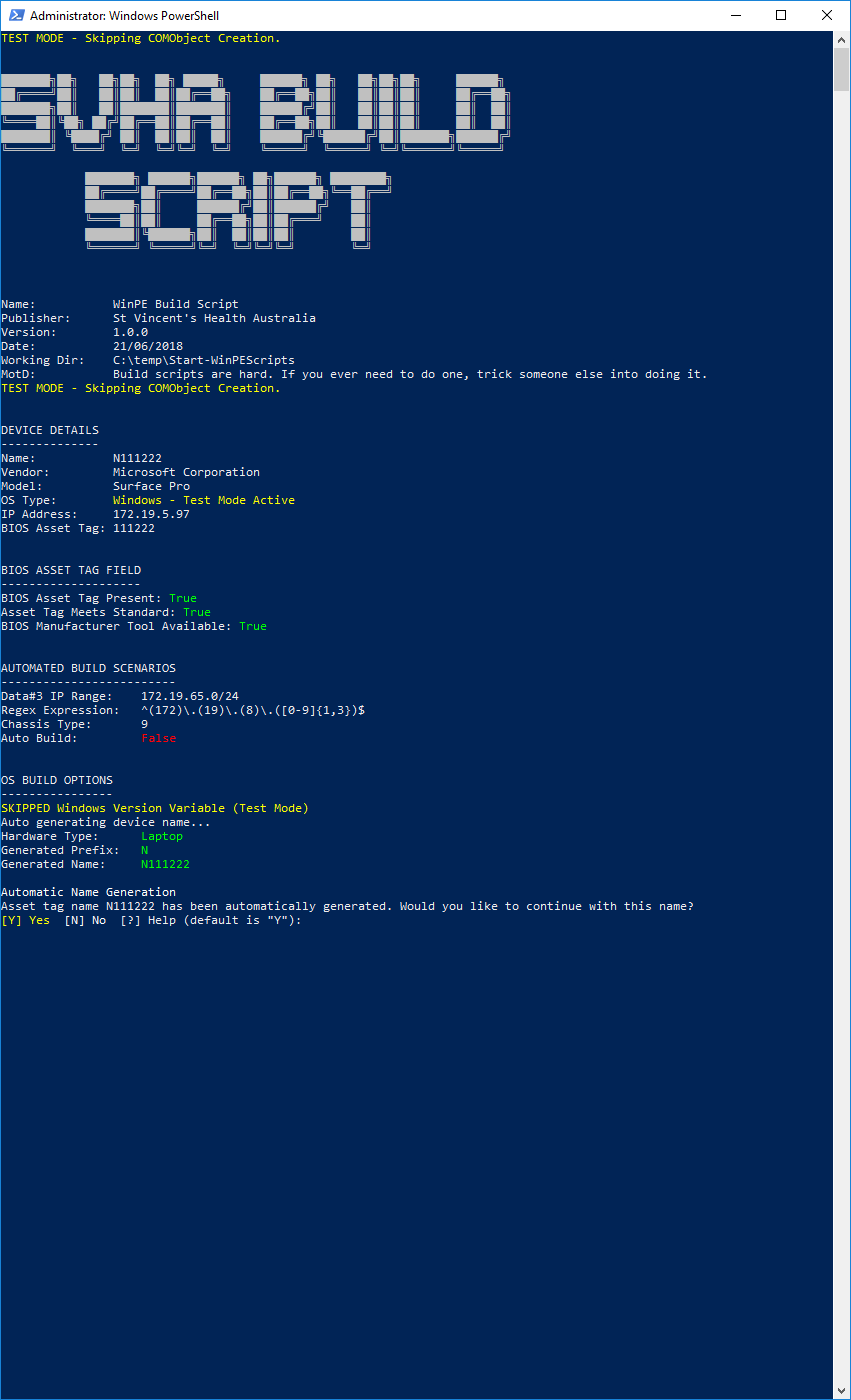
## Script Details

The first section of the script details information about the script itself which can be helpful when troubleshooting issues occurring within the script. The primary items to take note of when reporting a problem is the version, and working directory.



## Device Details

The device details section lists automatically generated information on the device being built. The information is pulled primarily from WMI and informs several of the later steps in the process. This information can be invaluable when troubleshooting certain issues within the build. Refer to the troubleshooting section below for more information

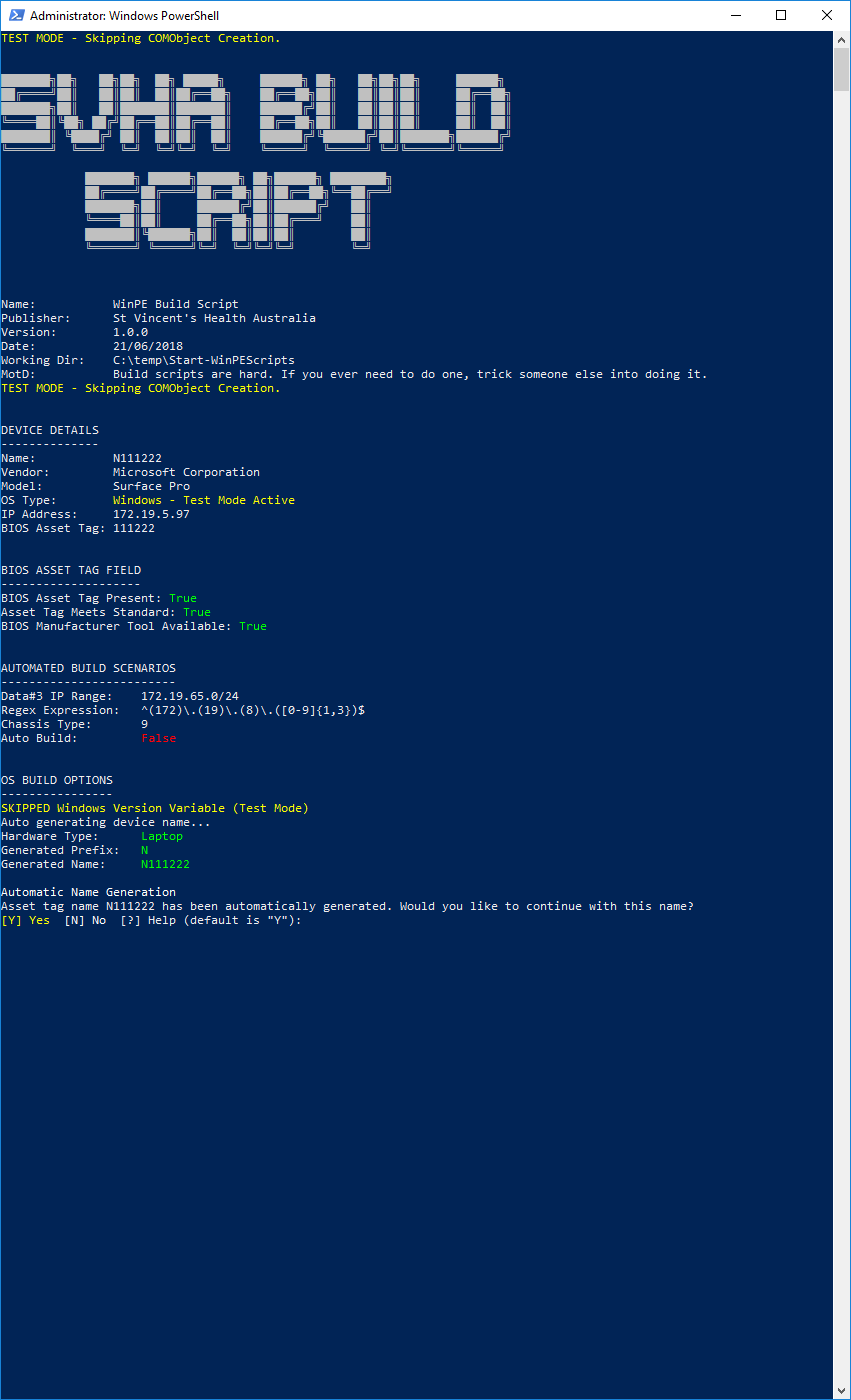


## BIOS Asset Tag Field

This is the first of the interactive sections. The script will check whether there is an asset tag set within the BIOS, whether that asset tag conforms to the SVHA standard of six digits and if a tool has been set up within the script to automatically set the asset tag within the BIOS. Currently only Surface Pro devices can be set automatically. All other device types will need to be entered manually within the BIOS setup page.

### BIOS Set Correctly

Should the BIOS be set correctly the script will continue without requesting any user input and will display the following.



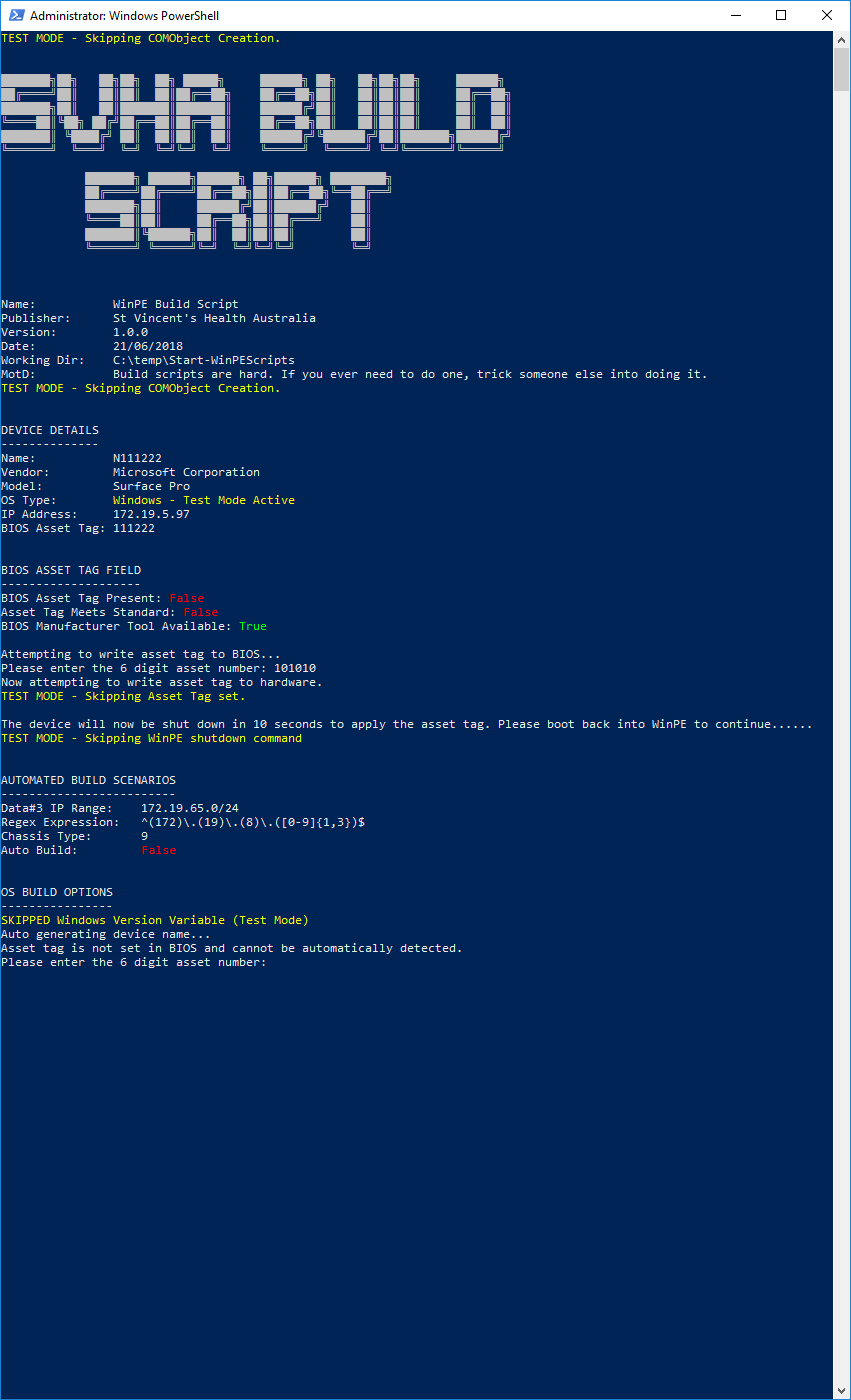
### BIOS Set Incorrectly

If the asset tag has not been set in the BIOS, has been set incorrectly (e.g. does not match a 6-digit number) or is a Surface Pro that has not yet had its asset tag set in the BIOS, depending on the type of device you will be presented with the option to input the asset tag.

#### Surface Pro

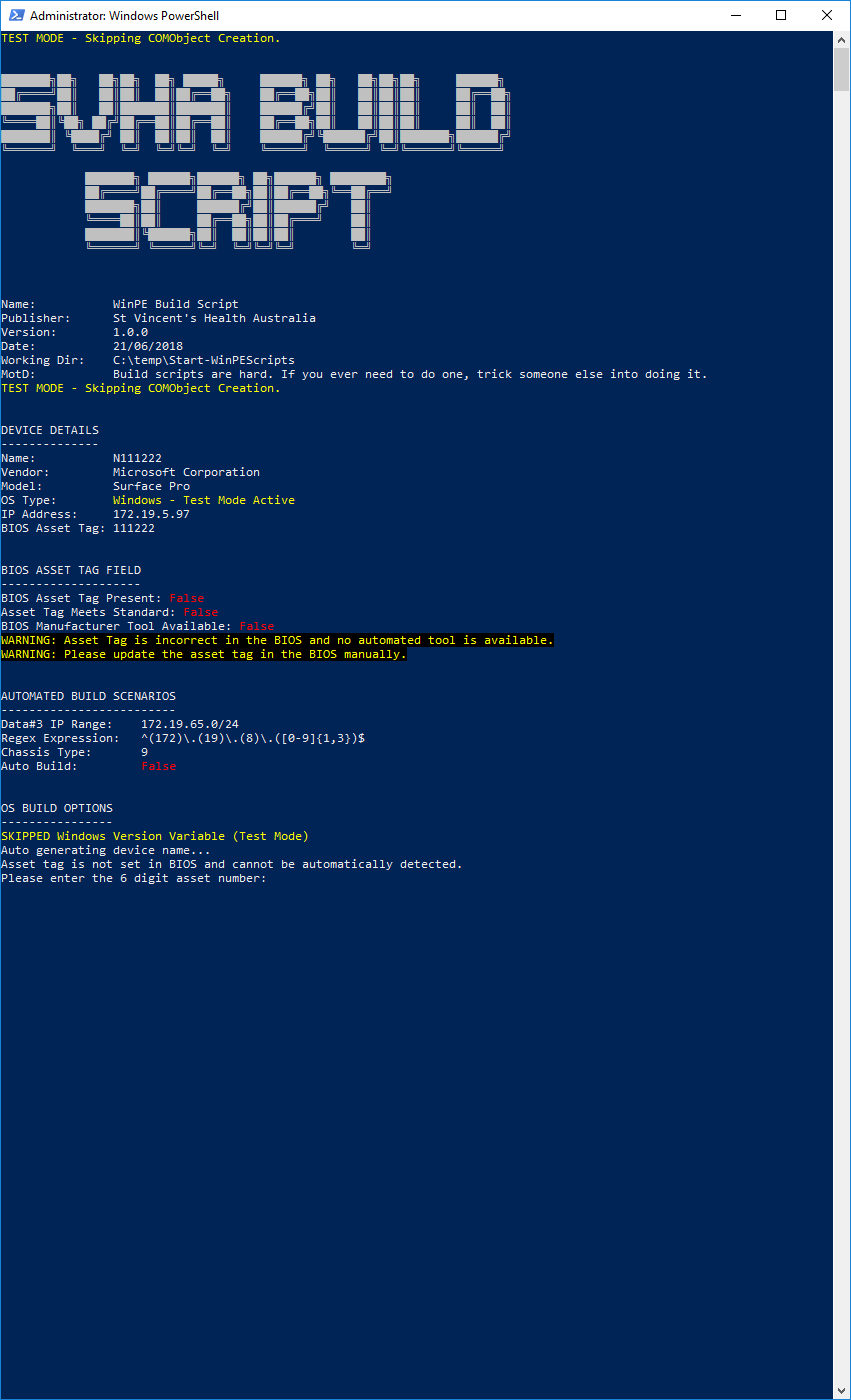
For a Surface Pro device, you will see the following screen. After inputting a 6-digit asset tag the script will use a Microsoft provided utility to write that tag to the BIOS. To complete this, the script will shut down the Surface Pro. Please boot back into WinPE to continue.

This will hopefully be expanded in the future to include more manufacturer types.



#### All other devices

For any other device where the asset tag cannot be automatically set during the script, a warning will display advising the user to set the asset tag in the BIOS. This is highly recommended to assist with future builds however not required.



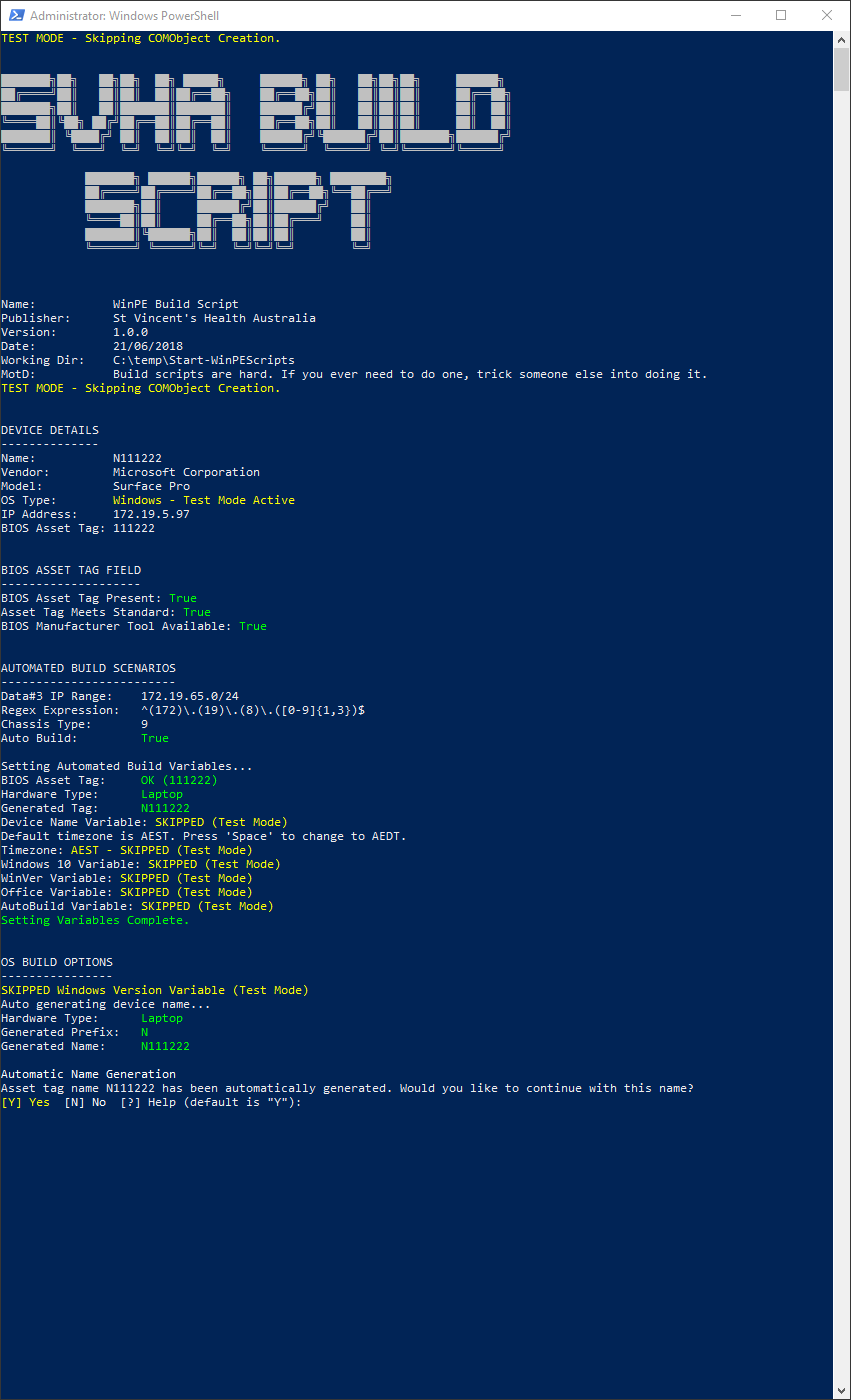
## Automated Build Scenarios

This component deals with automated build scenarios. Presently this only includes automated builds for Data#3 but can be expanded in the future when and where needed. This does not require any user input however there is a short window where user input is accepted to switch the time zone.

### Automated Build Site (Data#3)

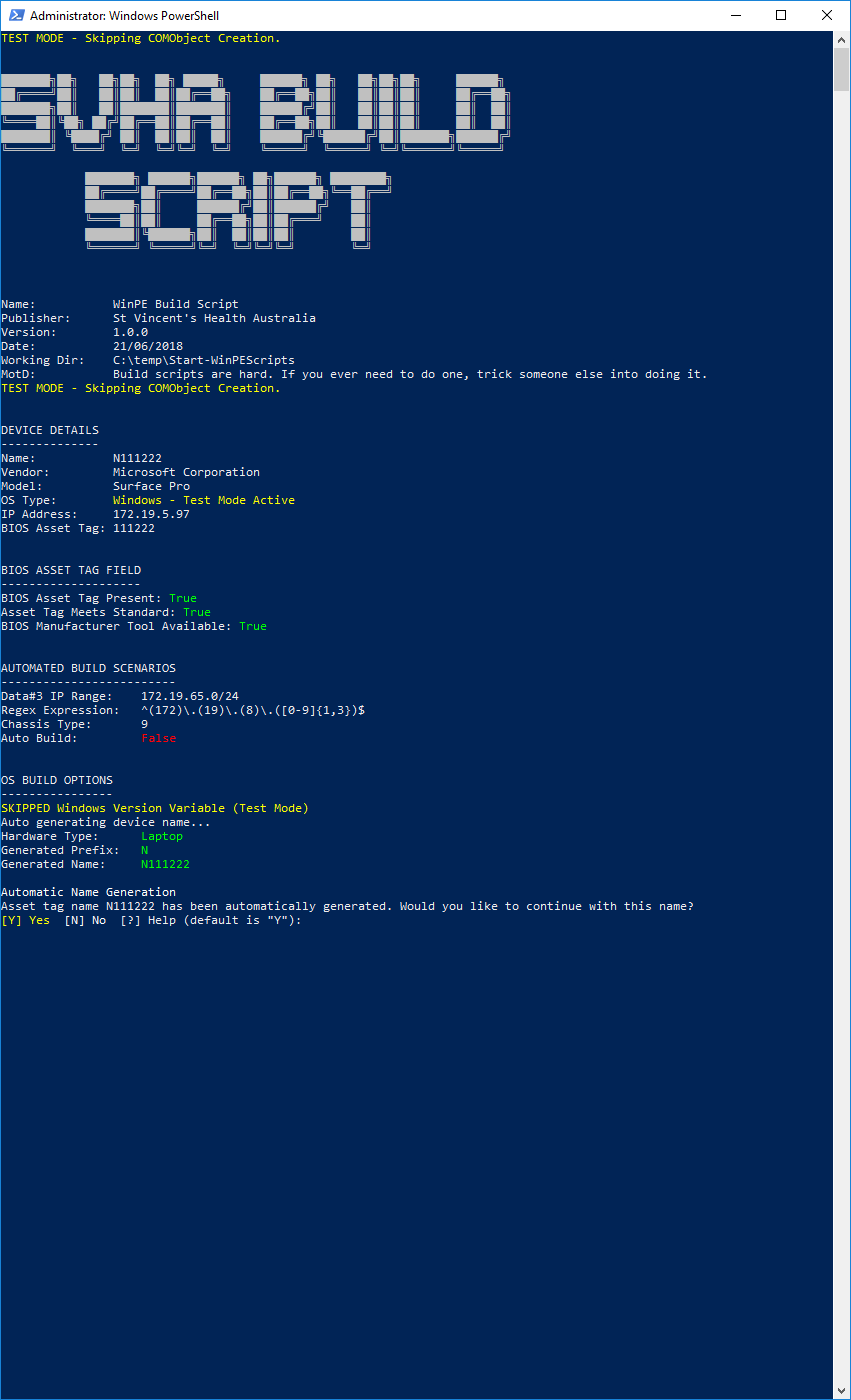
For Data#3 subnets the script will automatically set all required variables for a standard build. This includes:

* Automatically generated name
* Windows 10
* Office 2016 x86 MSI VLK
* AEST Timezone (unless switched by user input)
* Auto Build Variable



### Normal Build Site

For all other sites this section will skip without any further action. This check is subnet based.



## OS Build Options

The OS Build Options section is where the most user input is required. This section is the equivalent to the UI we all know and love where the name is input (or automatically generated), the time zone of the computers location is set, the Office version is set and any location based customisations or changes are made (e.g. SVPHM Custom App set). Due to the number of options in this section each will be under their own header for ease of use when referring to this documentation.

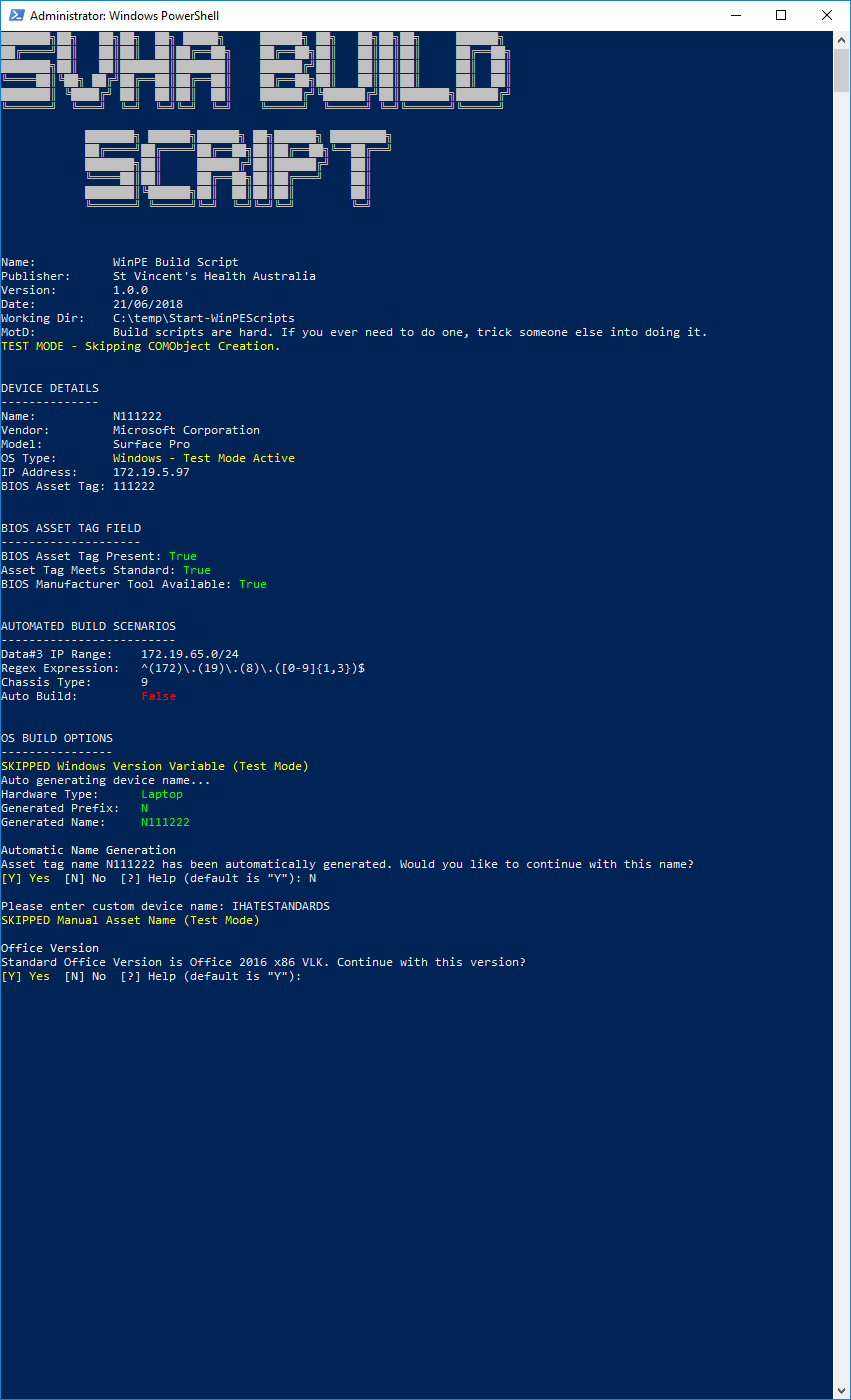
### Automatic Name Generation

Now that most sites have moved to the new naming standard (with most new devices being supplied built from Data#3, automated naming has now become the default. This is where the asset tag being set in the BIOS begins to make life easier.

If the asset tag is set in the BIOS and the “BIOS Asset Tag Field” section displayed “True” for both the Asset Tag Present and “Meets Standard” options, the script will automatically generate the name based on the type of device and the 6-digit asset number.

Once generated it will ask whether to proceed with the automatically generated name. If you wish to proceed you can just press “enter” (yes is the default if nothing is input). If you absolutely need to use a custom name, type “N” and press “enter”. You will then be prompted for the custom name.

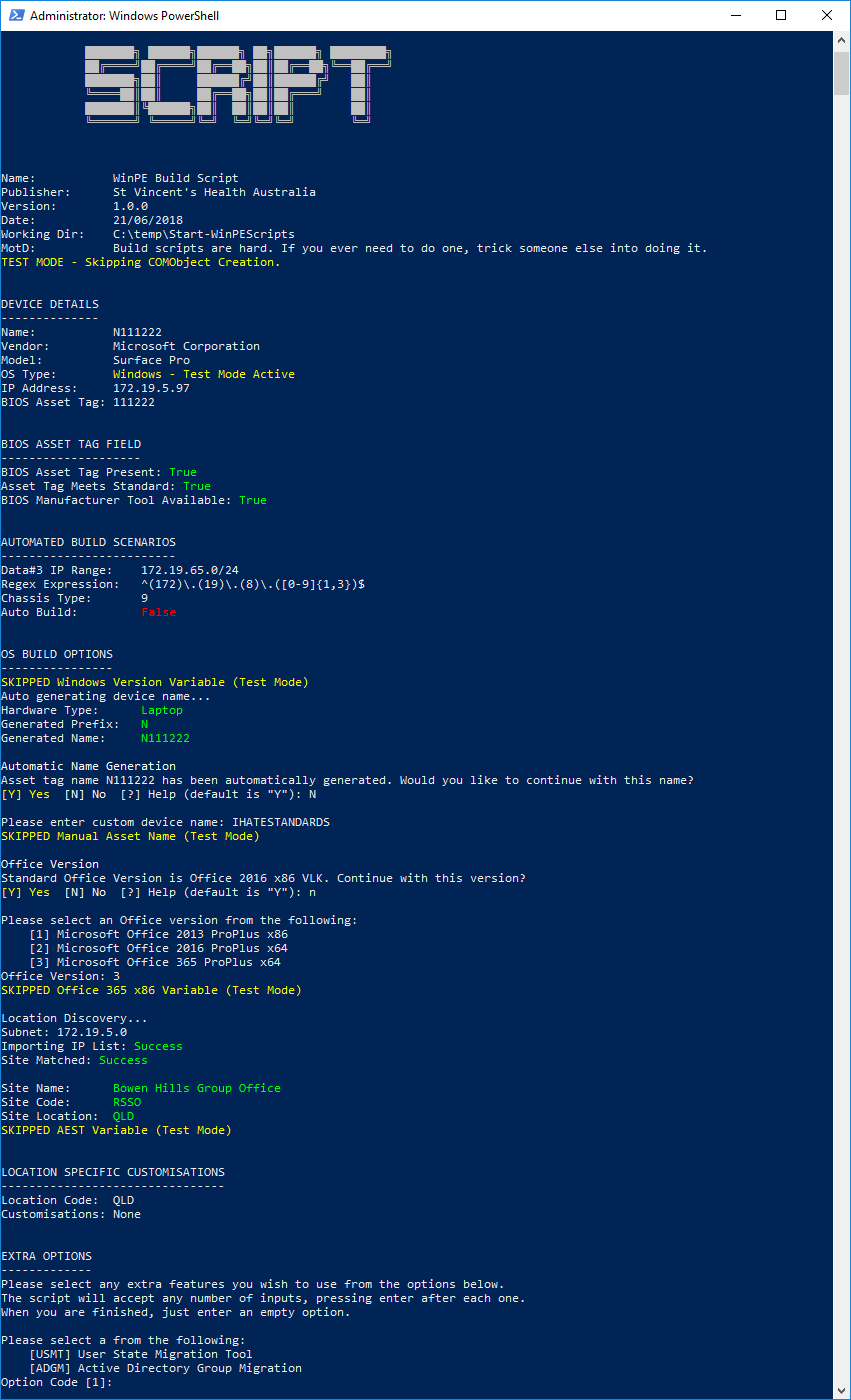
**Note:** You will not be asked to double check the custom name, please ensure you have entered it correctly before proceeding.



### Office Version

You will then be asked if you wish to continue with the standard Microsoft Office installation. Currently the standard is Microsoft Office 2016 x86 MSI. Again, if you wish to proceed with the standard office just press “enter” (no input needed). Otherwise type “N” and press enter.

If you select “No” you will be presented with a menu requesting the Office install you would like to proceed with. Type the number displayed to the left of the Office version you would like and press “enter”.



### Location Discovery

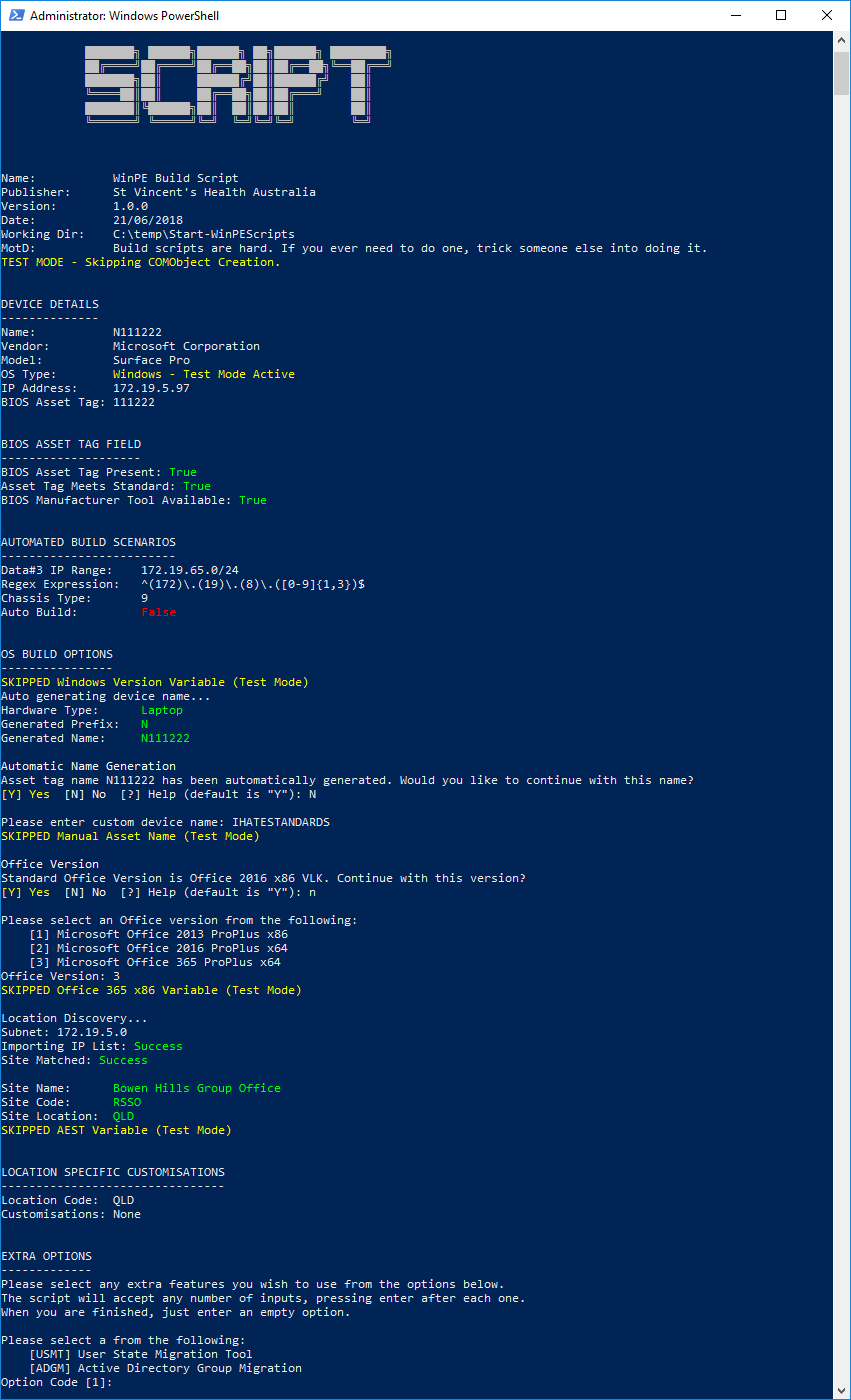
The location discovery component uses the IP subnet the device is currently connected to and a custom created list to determine where the device is located and automatically set the location and time zone for the build.

If the site you are building from is not currently in the list (currently extremely small but will grow as I find time) you will be prompted for the location of the device to continue the process. As with the previous section, enter the number to the left of the option you wish to select and press “enter”.

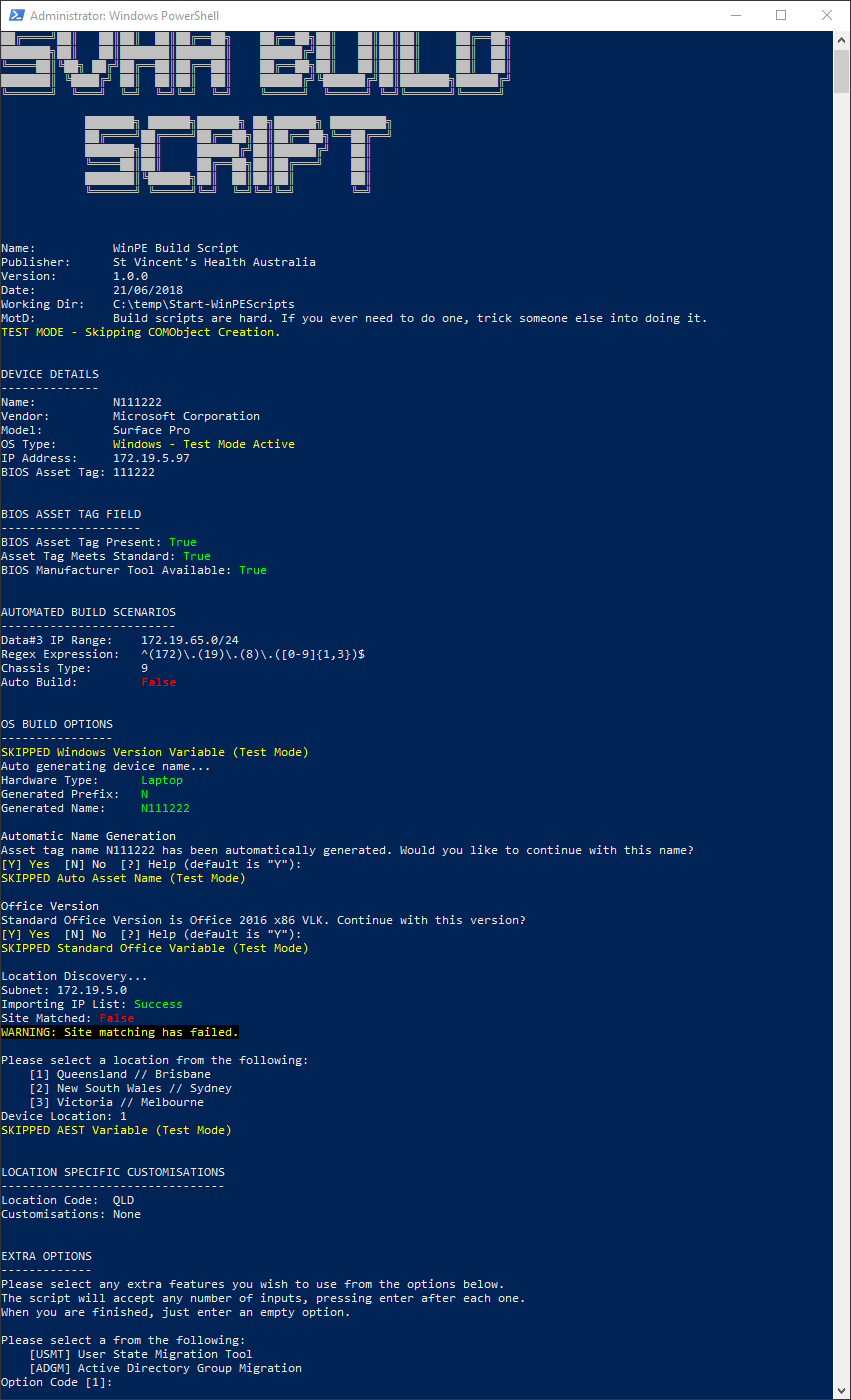
To assist me in creating the list, if you have the time please send through the following details when you encounter locations that are not detected automatically:

* IP Address
* State
* Site Code
* Full Site Name

#### Automatic Example



#### Manual Example

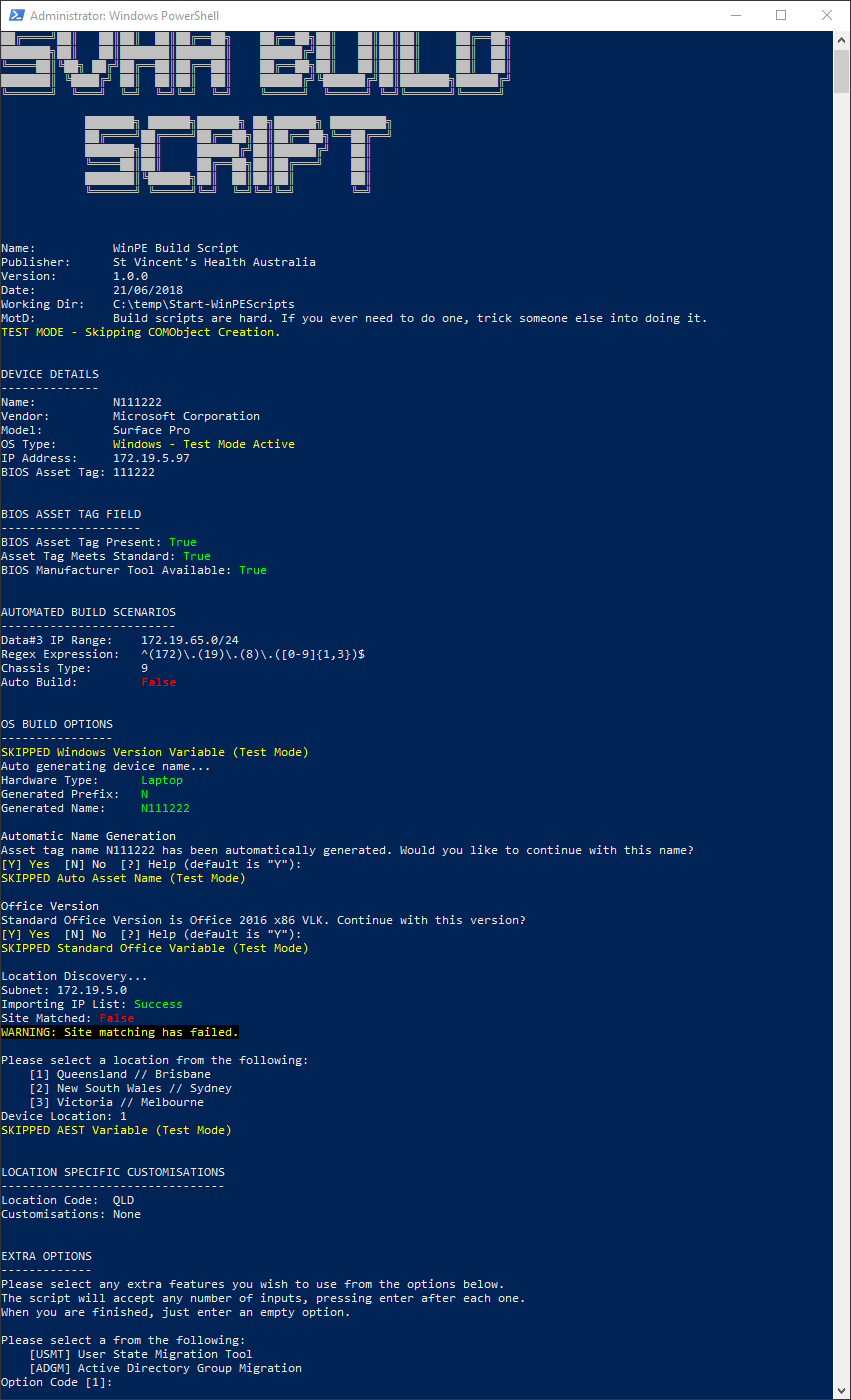


### Location Specific Customisations

The location specific customisations components provide the same options the NSW and VIC pages did in the existing UI.

#### Queensland

Currently Queensland does not have any specific customisations.

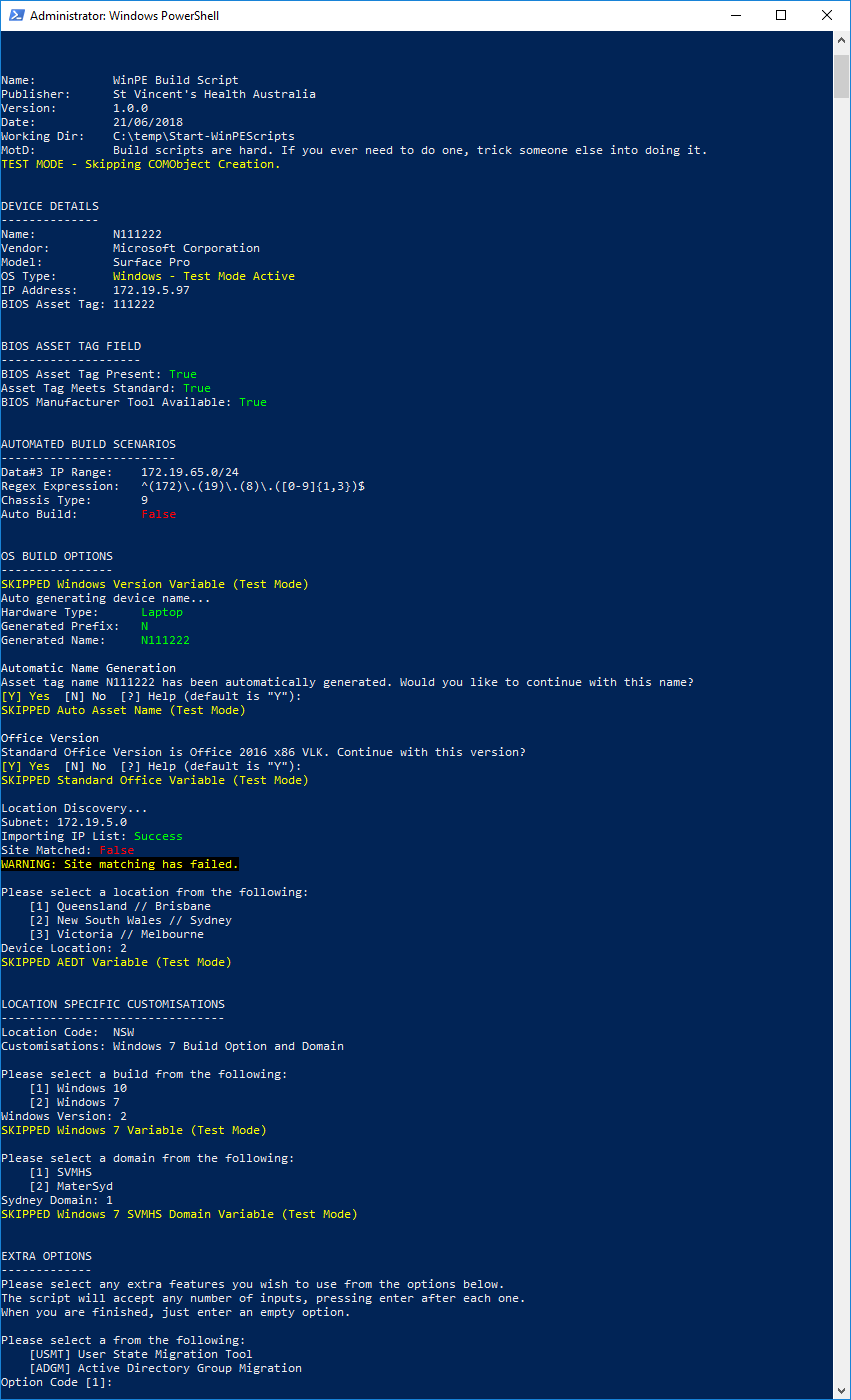


#### New South Wales

New South Wales are presented with the option of building to Windows 7 or Windows 10. This option will remain available until their migration later this year.

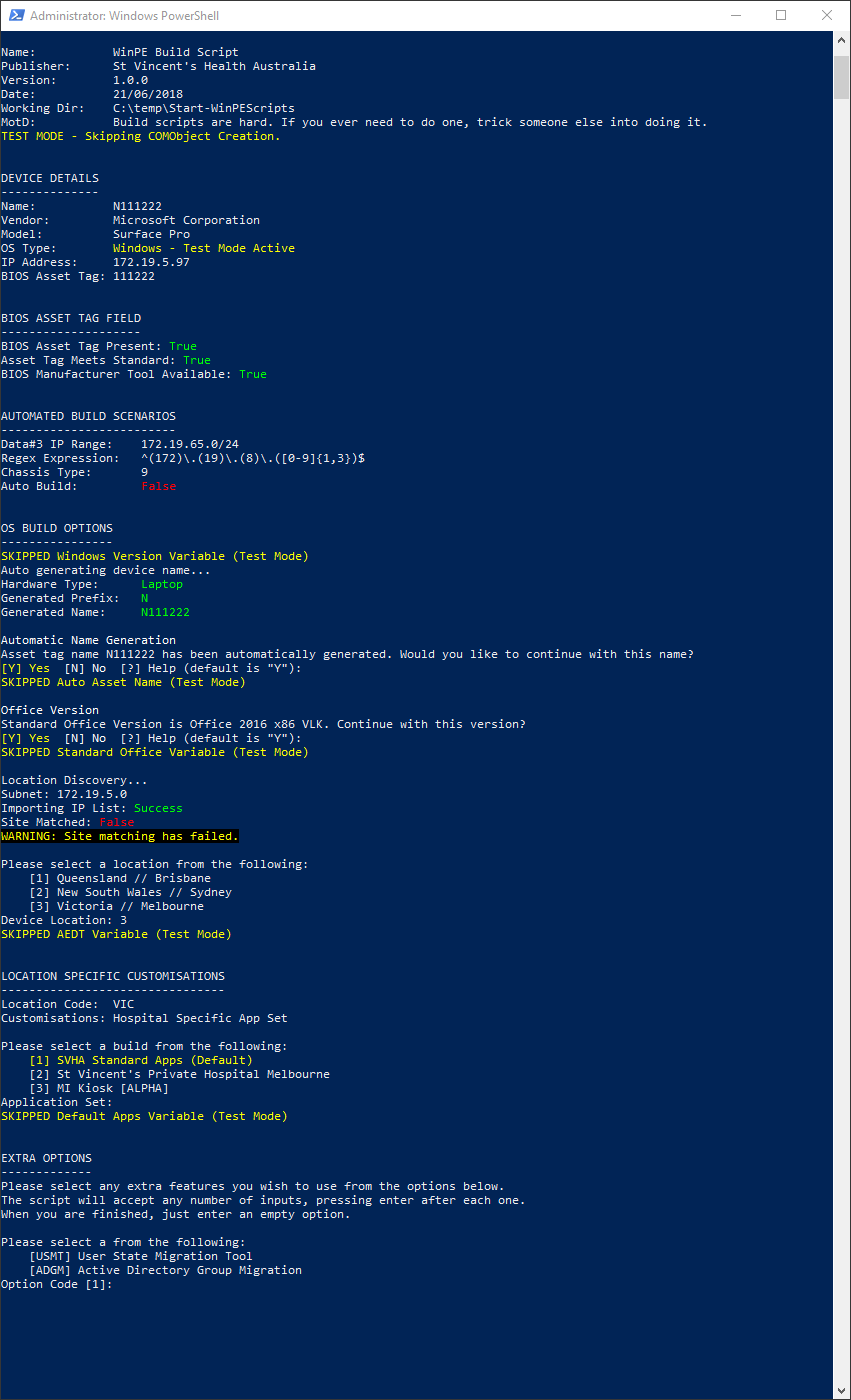
If Windows 7 is selected, a further menu asking whether it should be joined to the SVMHS or MATERSYD domain will be shown.

**Note: There is no default option for either of these 2 prompts so pressing enter without entering a value will return an “invalid input” error. This will be corrected in the future.**



#### Victoria

Victoria has a specific app set required for SVPHM which was available in the existing UI, this has been migrated to the new solution. As with previous menu’s, enter the option required and press “enter”. If the default app set is required, just hit “enter”.

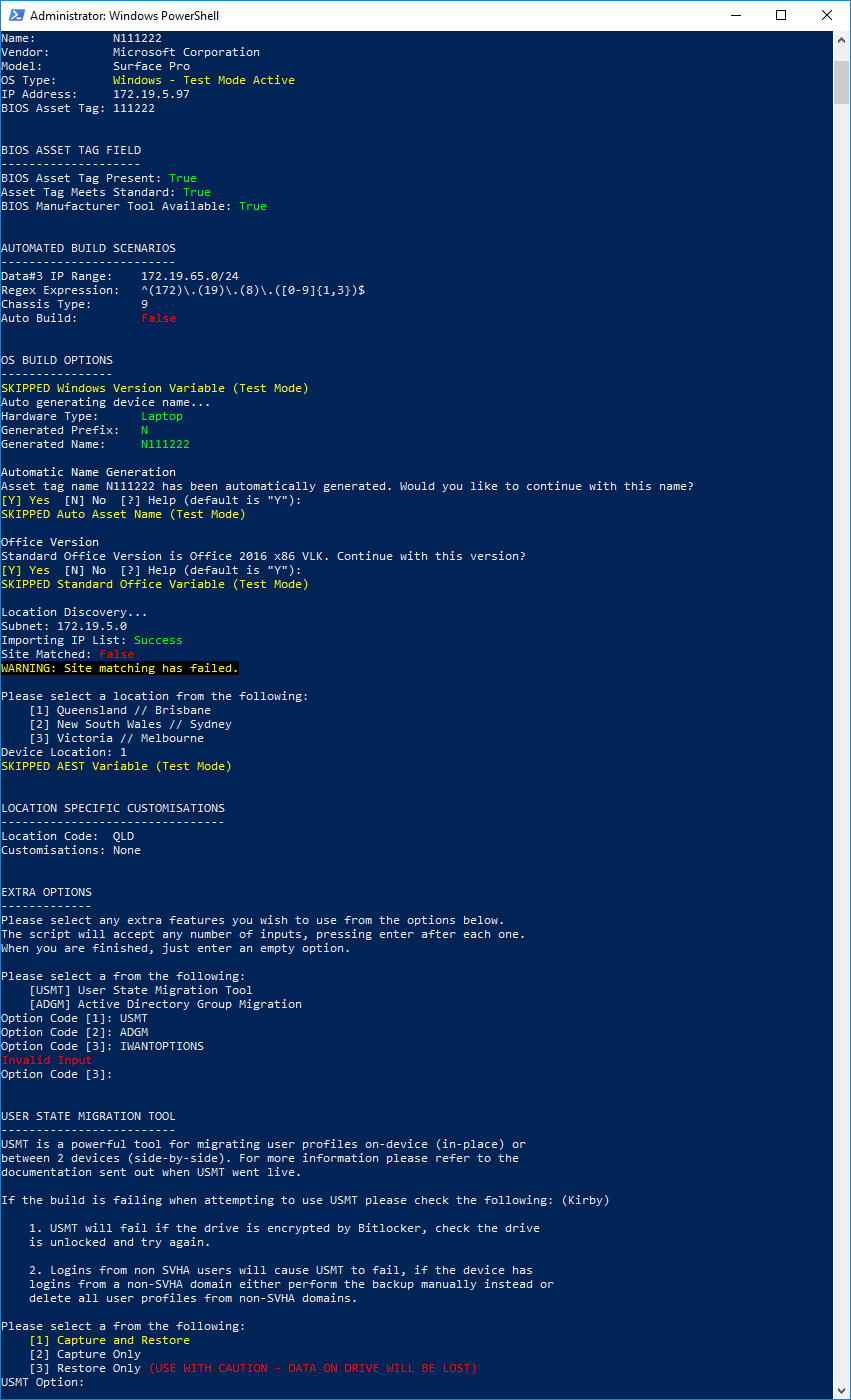


### Extra Options

The final options that needed to be migrated were the USMT and AD Group Migration options available in the existing UI. Data showed that neither of these has been used in anger so the solution here is a little different to previous menu’s. If you find it is too troublesome let me know and I can switch it around.

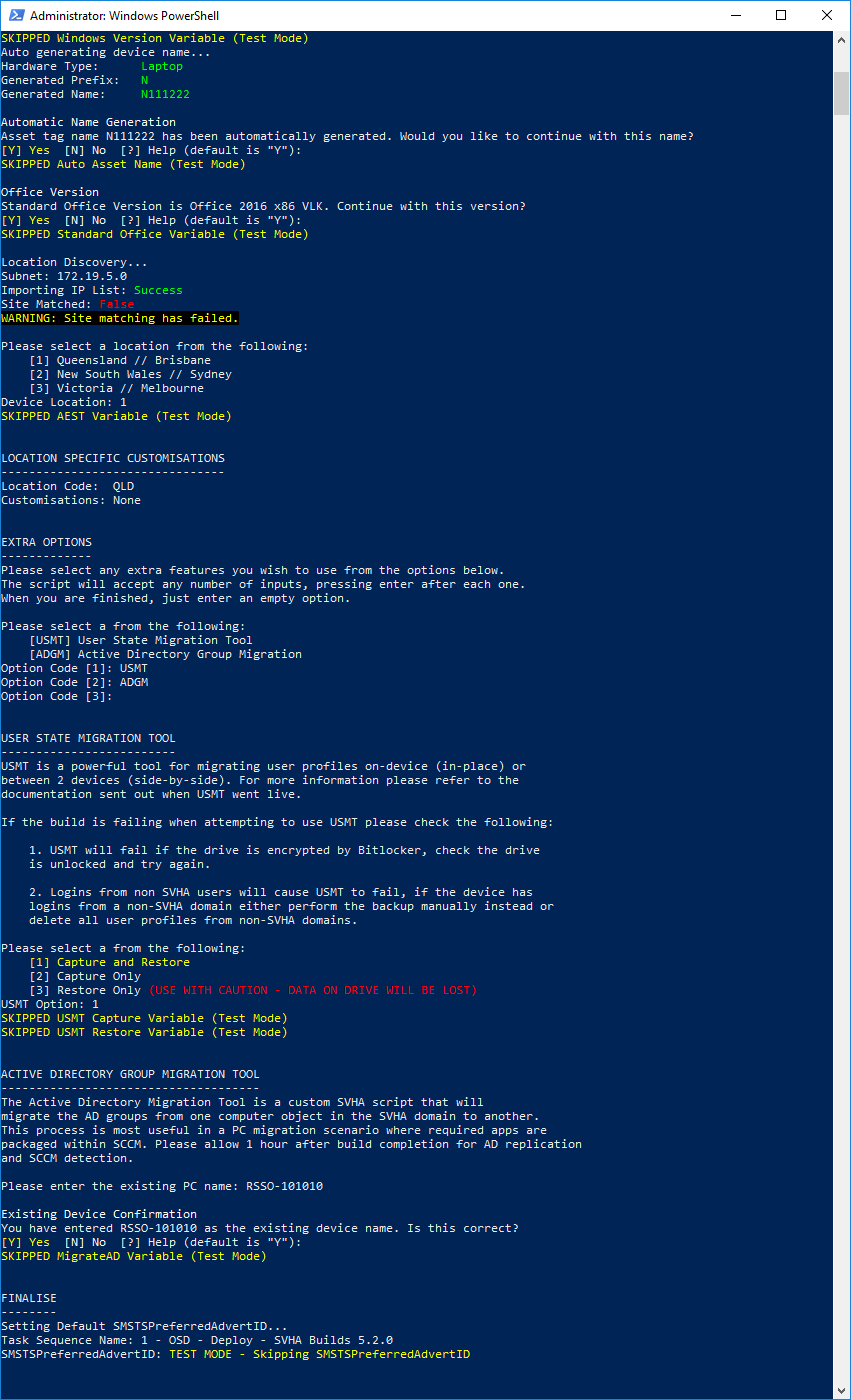
You will be presented with the Extra Options section as the last piece of the script. From here you have the option of entering any number of options, pressing enter after each one. If you type an option that isn’t available the script will let you know.

Select any options you wish to proceed with. If you do not wish to use any just press enter to proceed.



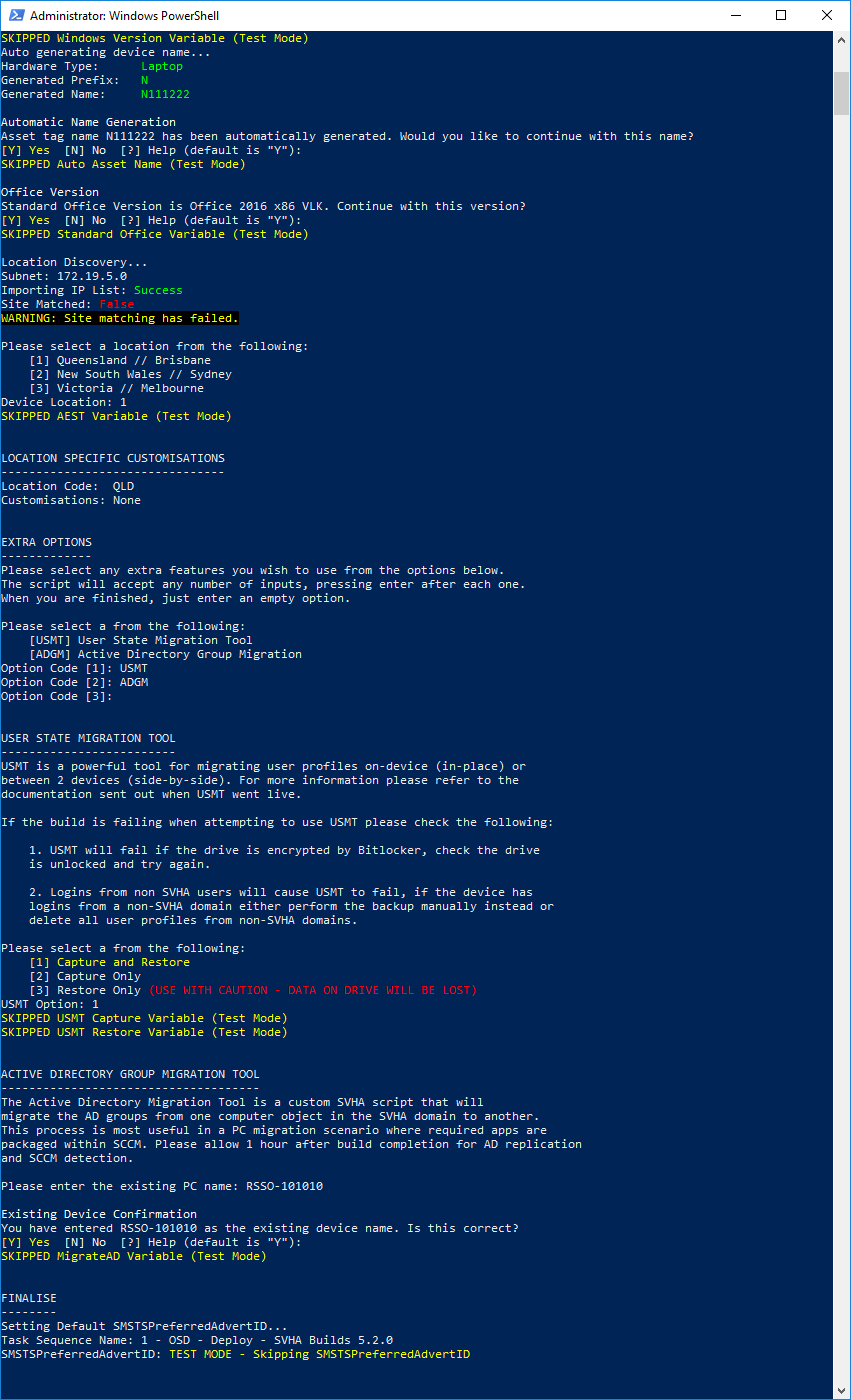
#### User State Migration Tool

If you chose to use USMT you will be presented with the following options. For more information on each one, please refer to the USMT documentation that was sent out when USMT went live. As with all previous menu’s, please enter the option you wish to use and press “enter”.



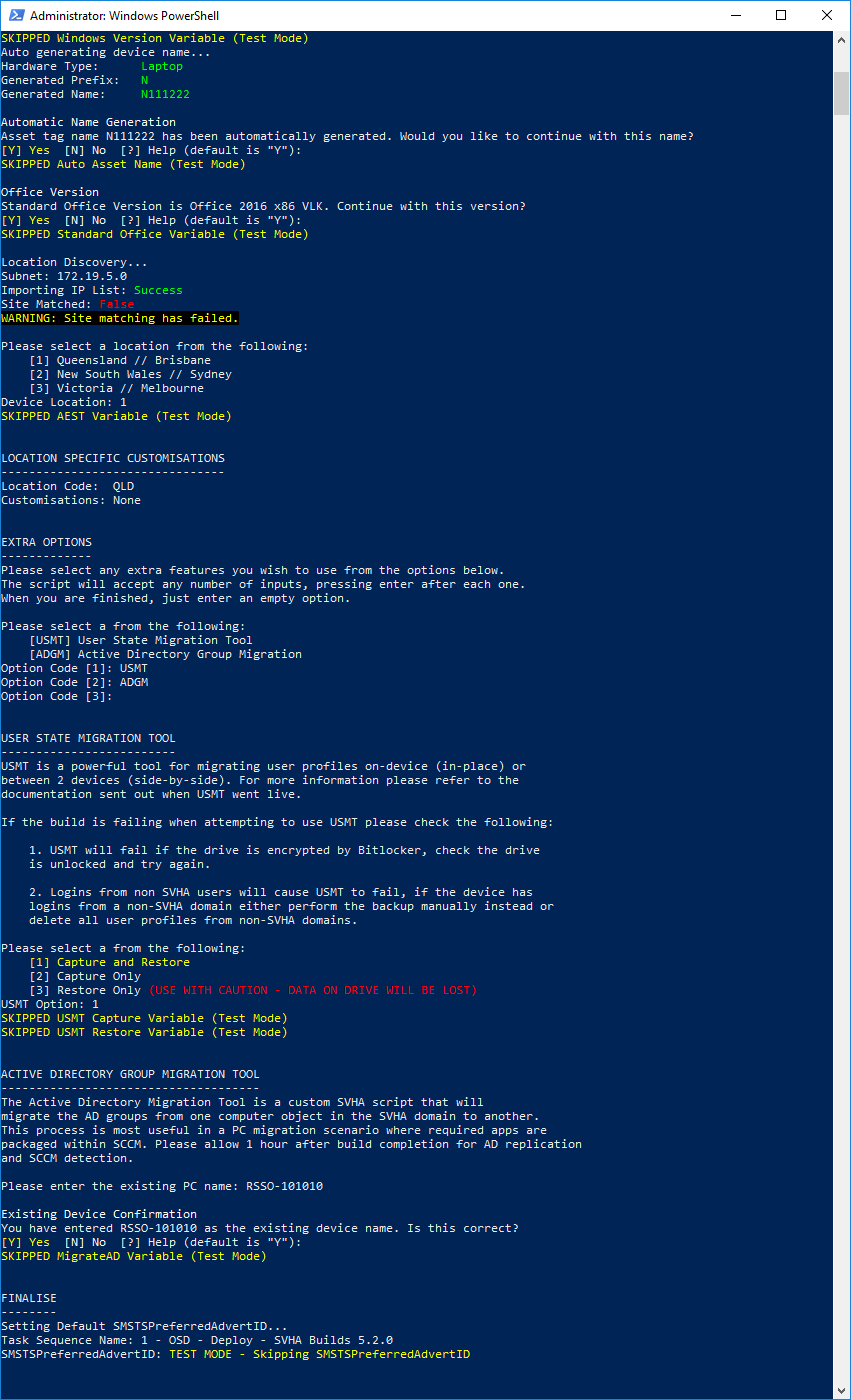
#### Active Directory Group Migration

If you wish to use the Active Directory Group Migration option you will be presented with the following. Please enter the device name of the *existing* device you wish to migrate the AD groups *from*. When prompted, confirm the entry is correct by entering “Y” and pressing enter or by simply pressing “enter”.



### Finalise

The final section sets a variable that allows the device to build. As mentioned earlier, this section means an end to the requirement of putting a device in a collection before it will build. This will be tested initially to ensure no issues arise.



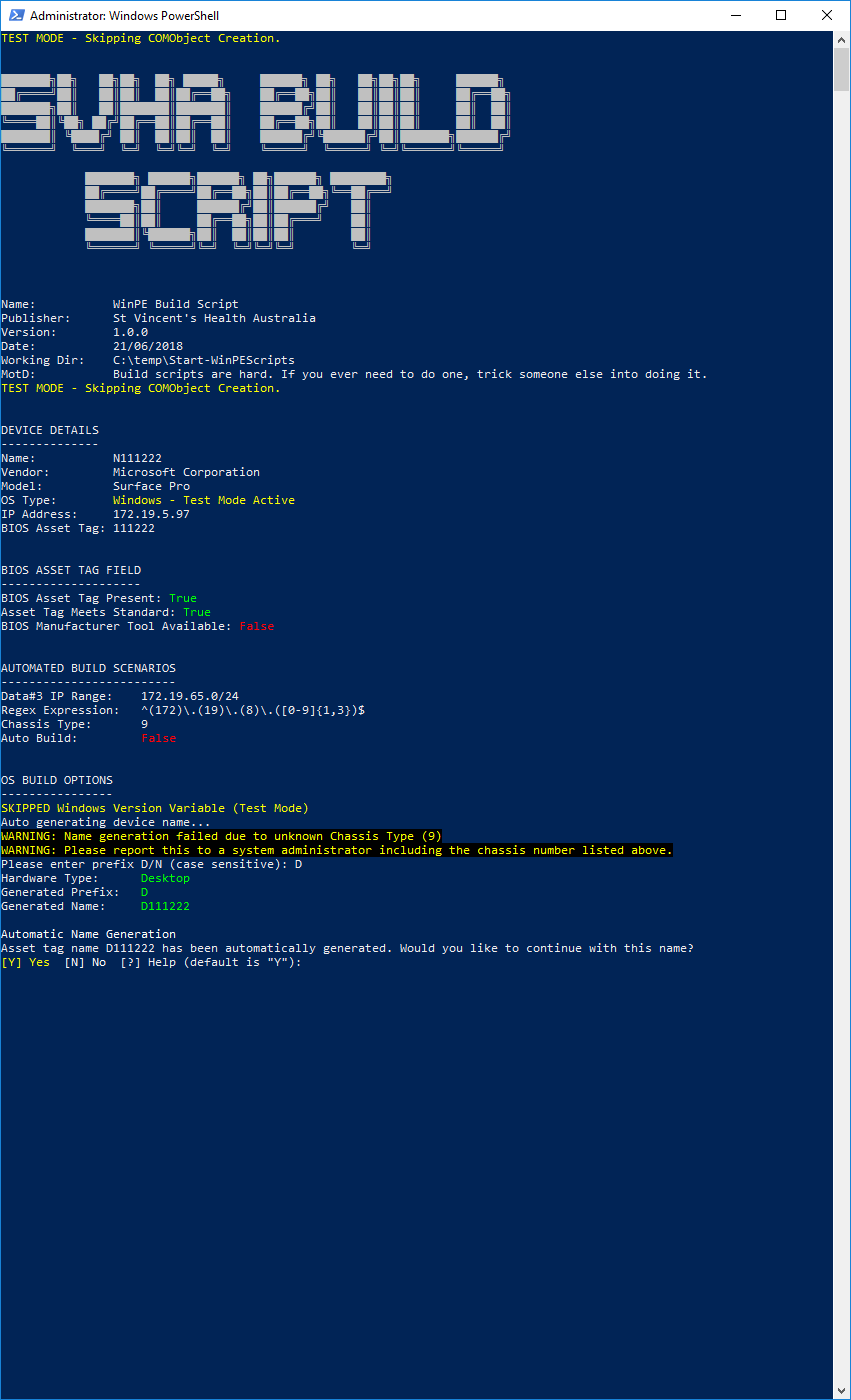
# Troubleshooting & FAQ

**Q.** I’m getting a “Name generation failed due to unknown chassis type” error! Fix please!

**A.** To determine whether the name should be prefixed with an “N” or a “D”, the script uses a WMI entry called “Chassis Type”. These values are part of a standard and can be found on Microsoft’s website. This error means this device has a chassis type not known to the script.

To allow you to build the script will prompt for a prefix and the script will then continue.

Please email though the chassis type and the expected prefix to Adam Stevens to have it updated for future builds.



**Q.** I have a suggestion for an improvement.

**A.** There is no such thing, it is perfection. If you wish though, you’re welcome to send through your idea. If it is good I will steal it and claim it as my own.

**Q.** The script is forcefully exiting and the build doesn’t start!

**A.** Please give me a call and we can work on it together. In the meantime, if you are unable to get in contact and the build is urgent, the existing rebuild options will continue to function for the transition. If you aren’t getting the option to select a task sequence, start the process again but force quit the script before it completes.

**Q.** I don’t like it, the UI was better!

**A.** While I understand a CLI script like this isn’t always as user friendly as a UI, the functionality gains we get from using it warranted the effort. If the script is confusing or slowing you down, please get in contact and I can work on improving or reverting the changes.