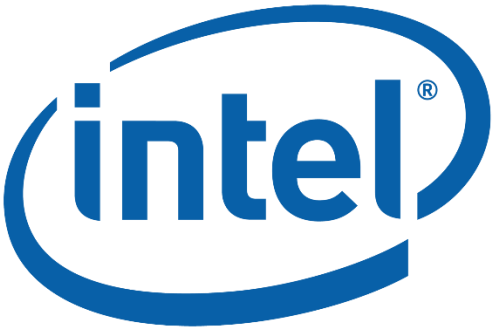


VCUST

User Guide

Version 1.0.0

June, 27, 2019

Document Number: XXXXXX

This product specification applies to only the standard Intel NUC Software Product.

INFORMATION IN THIS DOCUMENT IS PROVIDED IN CONNECTION WITH INTEL® PRODUCTS. NO LICENSE, EXPRESS OR IMPLIED, BY ESTOPPEL OR OTHERWISE, TO ANY INTELLECTUAL PROPERTY RIGHTS IS GRANTED BY THIS DOCUMENT. EXCEPT AS PROVIDED IN INTEL'S TERMS AND CONDITIONS OF SALE FOR SUCH PRODUCTS, INTEL ASSUMES NO LIABILITY WHATSOEVER, AND INTEL DISCLAIMS ANY EXPRESS OR IMPLIED WARRANTY, RELATING TO SALE AND/OR USE OF INTEL PRODUCTS INCLUDING LIABILITY OR WARRANTIES RELATING TO FITNESS FOR A PARTICULAR PURPOSE, MERCHANTABILITY, OR INFRINGEMENT OF ANY PATENT, COPYRIGHT OR OTHER INTELLECTUAL PROPERTY RIGHT. UNLESS OTHERWISE AGREED IN WRITING BY INTEL, THE INTEL PRODUCTS ARE NOT DESIGNED NOR INTENDED FOR ANY APPLICATION IN WHICH THE FAILURE OF THE INTEL PRODUCT COULD CREATE A SITUATION WHERE PERSONAL INJURY OR DEATH MAY OCCUR.

All Intel NUC Boards are evaluated as Information Technology Equipment (I.T.E.) for use in personal computers (PC) for installation in homes, offices, schools, computer rooms, and similar locations. The suitability of this product for other PC or embedded non-PC applications or other environments, such as medical, industrial, alarm systems, test equipment, etc. may not be supported without further evaluation by Intel.

Intel Corporation may have patents or pending patent applications, trademarks, copyrights, or other intellectual property rights that relate to the presented subject matter. The furnishing of documents and other materials and information does not provide any license, express or implied, by estoppel or otherwise, to any such patents, trademarks, copyrights, or other intellectual property rights.

Intel may make changes to specifications and product descriptions at any time, without notice.

Designers must not rely on the absence or characteristics of any features or instructions marked “reserved” or “undefined.” Intel reserves these for future definition and shall have no responsibility whatsoever for conflicts or incompatibilities arising from future changes to them.

Intel, the Intel logo, Intel NUC and Intel Core are trademarks of Intel Corporation in the U.S. and/or other countries.

* Other names and brands may be claimed as the property of others.

Contents

Revision History

Revision Number	Description	Revision Date
1.0.0	Initial Release	6/27/2019

1 Introduction

1.1 Overview

VCUST is a command line tool that is designed to assist integrators (who usually work in manufacturing or enterprise environment) with the process of BIOS/SMBIOS customization, OEM Windows* product key injection (OEM Activation 3.0, OA3) of Intel® NUC products.

Section 2 gives information on how to display help messages and current settings. Section 3 gives detailed information on how to use certain commands. Section 4 gives general description of three switches. Section 5 provides a table, listing every combination of commands and switches and its meaning. Section 6 illustrates how to run VCUST. In the end, section 7 gives an example.

1.2 Terminology

Term	Description
BIOS	Basic Input/Output System
OA3	OEM Activation 3.0
OEM	Original Equipment Manufacturer
OS	Operating System
SMBIOS	System Management BIOS

2 How to run VCUST

NOTE: VCUST must be run in administrator privilege.

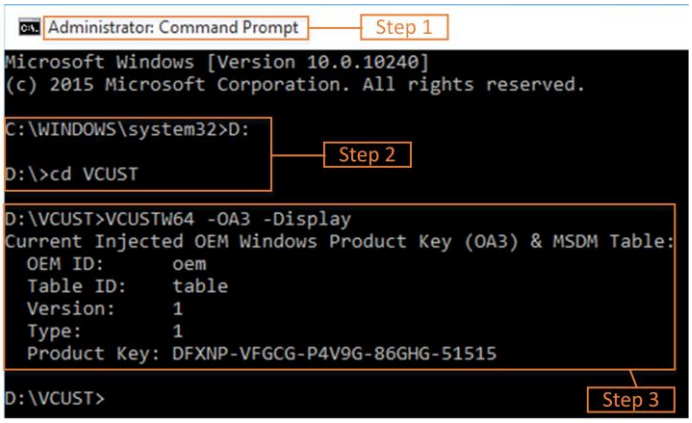
It is intended for 64-bit Windows systems.

2.1 How to run VCUST in Windows 10

Step 1. Open a command prompt as administrator
If you run from a batch file, right click and select “Run as administrator”.

Step 2. Change directory to the folder that contains VCUSTW64.exe

Step 3. Run a command, such as “VCUSTW64 -OA3 -Display”
If there OA3 data on the BIOS, it will be displayed.



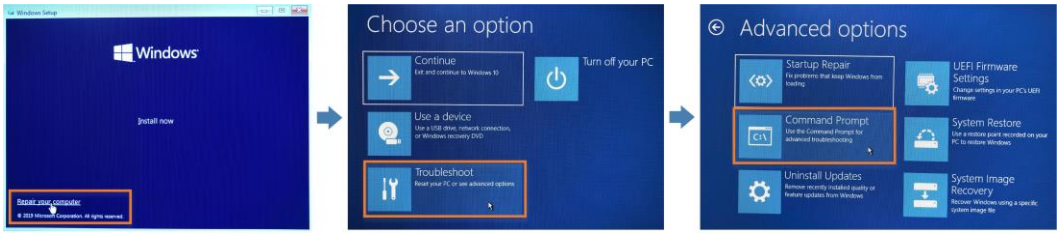
2.2 How to run VCUST in WinPE

Step 1. Create a Windows 10 installation media (bootable USB thumb drive) following Microsoft's Media Creation Tool website.

Step 2. Create a folder on the USB drive (e.g., named VCUST) and copy VCUSTW64.exe to it.

Step 3. Boot from the USB drive.

Step 4. In Windows Setup, select “Repair your computer”. Choose “Troubleshoot” and then “Command Prompt”.

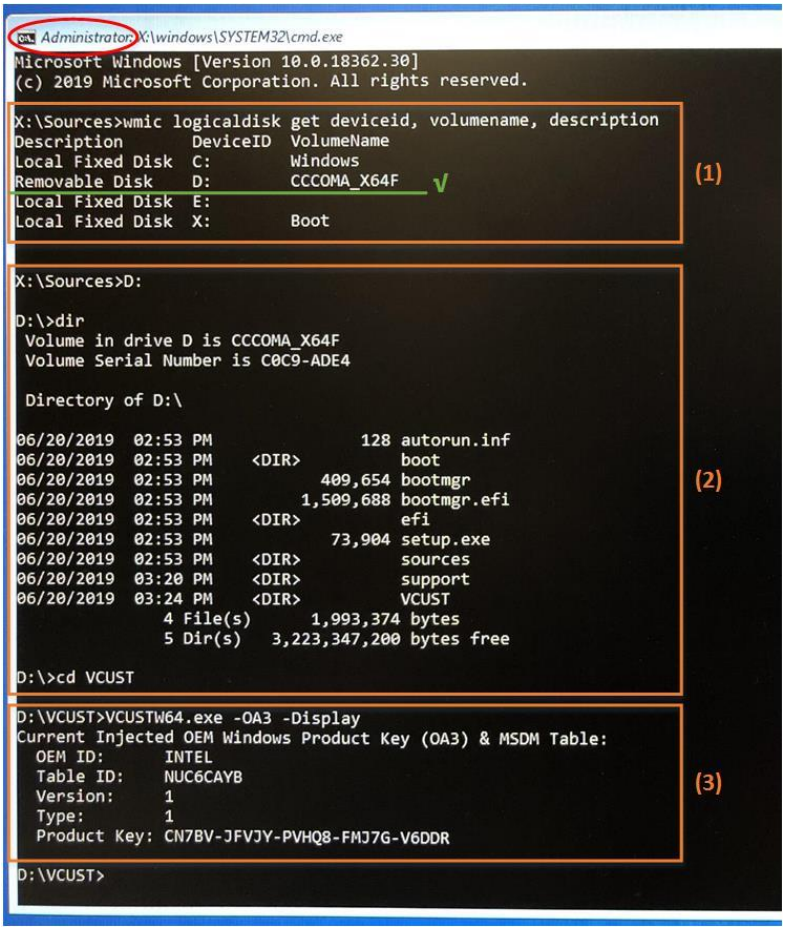


Step 5. Run VCUST in the Command Prompt as administrator.

(1) Find the USB drive.

(2) Change directory to the folder that contains VCUSTW64.exe.

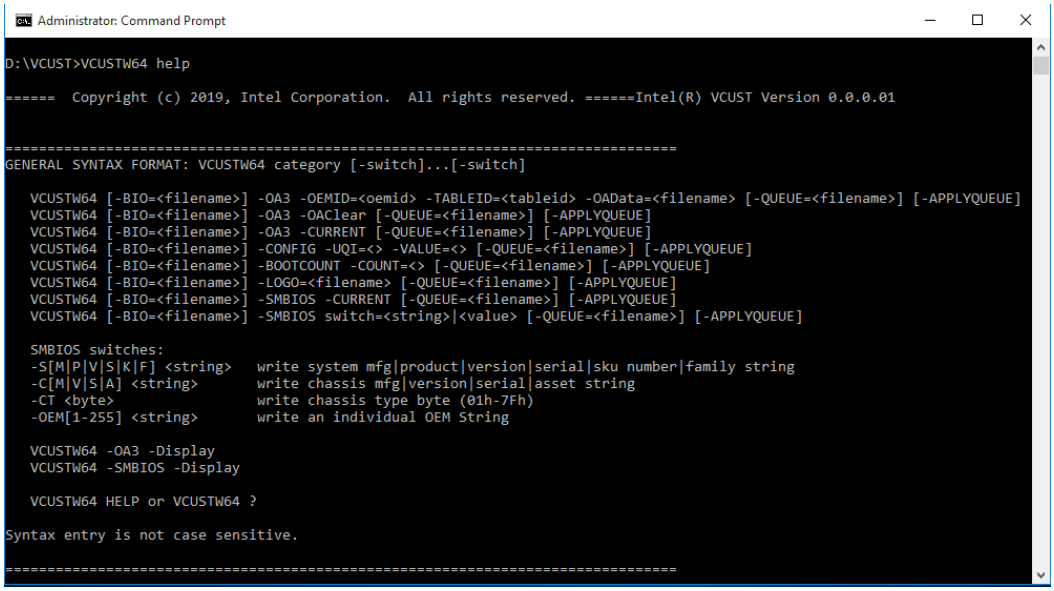
(3) Run a command, such as “VCUSTW64 -OA3 -Display”.
If there OA3 data on the BIOS, it will be displayed.



3 Help Function

The help splash screen below can be displayed by one of the following commands:

- *VCUSTW64 HELP*
- *VCUSTW64 ?*



Current OA3 data and current SMBIOS settings can be displayed by:

- *VCUSTW64 -OA3 -Display*
- *VCUSTW64 -SMBIOS -Display*

4 Commands

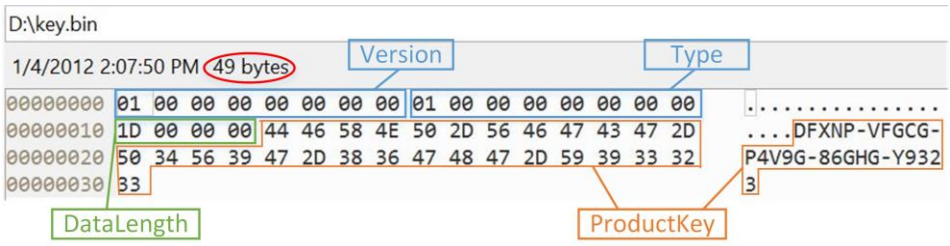
Denote **-COMMAND** as a command that can be used together with -BIO, -QUEUE, and -APPLYQUEUE.
-COMMAND can be one the following.

Note: Only one -COMMAND is allowed at a time. Do not use multiple -COMMANDs together.
Example: Instead of "VCUSTW64 -OA3 -CURRENT -SMBIOS -SM=Intel" use this:

- *VCUSTW64 -OA3 -CURRENT -QUEUE=queue.bio*
- *VCUSTW64 -SMBIOS -SM=Intel -QUEUE=queue.bio -APPLYQUEUE*

4.1 -OA3 -OEMID=<oemid> -TABLEID=<tableid> -OAData=<filename>

1) Meaning: Update OA3 data.
2) Requirements:
   - oemid must be 1-6 ASCII characters.
   - tableid must be 1-8 ASCII characters.
   - OAData file must exist and
     - The file extension must be ".bin".
     - The size must be 49 Bytes (8 Bytes for Version, 8 for Type, 4 for DataLength, and 29 for ProductKey).
     - DataLength field must be 29 (0x1D).
     - ProductKey filed must be XXXXX-XXXXX-XXXXX-XXXXX-XXXXX, where X is a letter (a-z, A-Z) or a digit (0-9).
3) Example:
   VCUSTW64 -OA3 -OEMID=oem12 -TABLEID=table12 -OAData=key.bin
   where, key.bin is like this:



4.2 -OA3 -OAClear

1) Meaning: Clear OA3 data.
2) Requirements: None.
3) Example:
   VCUSTW64 -OA3 -OAClear

4.3 -OA3 -CURRENT

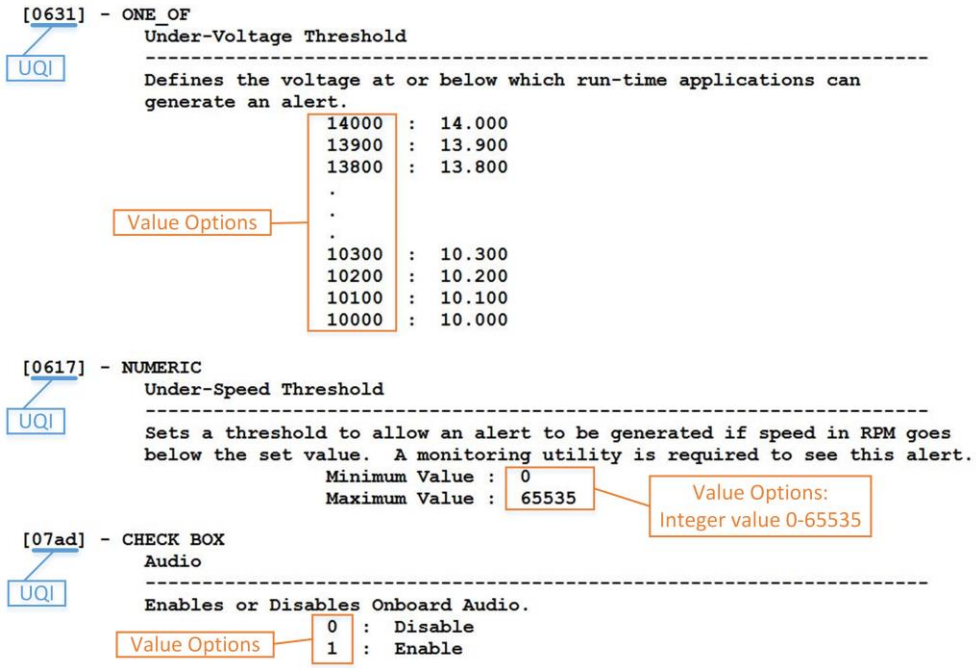
1) Meaning: Get the current OA3 data.
2) Requirements: None.
3) Example:

   VCUSTW64 -OA3 -CURRENT

4.4 -CONFIG -UQI=<> -VALUE=<>

1) Meaning: Update a BIOS setting.
2) Requirements:
   - UQI and value options should be from a report of the current BIOS.
   - UQI must be xxxx, where x is a hex digit (0-9, a-f, or A-F).
   - Value must be a combination of digits (0-9).
3) Examples:

   VCUST supports three types of BIOS settings.

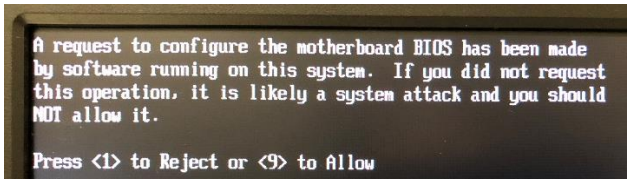


   Based on the UQI and value options above, here are corresponding command examples:

   VCUSTW64 -CONFIG -UQI=0631 -VALUE=13800
   VCUSTW64 -CONFIG -UQI=0617 -VALUE=255
   VCUSTW64 -CONFIG -UQI=07ad -VALUE=0

4.5 -BOOTCOUNT -COUNT=<>

1) Meaning: Set boot count value to enable unattended BIOS update procedure. If the boot counter has not decremented to 0, the message below won't prompt while applying the customization.



2) Requirements:
   - Decimal integer value 0-50.
3) Example:
   VCUSTW64 -BOOTCOUNT -COUNT=30

4.6 -LOGO=<filename>

1) Meaning: Change the logo image.
2) Requirements:
   - Must be a jpeg image and ".jpg" must be appended to the end of the filename.
   - The max size of the file (.JPG) is 60KB.
   - Resolution:
     - The X dimension resolution must >= 120 and <= 1920 pixels.
     - The Y dimension resolution must >= 120 and <= 1080 pixels.
     - The minimum resolution is 120 x 120 pixels.
     - The maximum resolution is 1920 x 1080 pixels.
   - The file must have been opened with MS Paint and saved. Otherwise, the jpeg image can be compressed in a way that cannot be shown during booting.
3) Example:
   VCUSTW64 -LOGO=D:\Logo\Intel_NUC_Default_307x230.jpg

NOTE: On the NUCs listed below, it requires a platform .BIO file as an input for VCUST to change the logo image successfully. A platform .BIO file for a specific NUC can be downloaded from Intel Download Center for Mini PCs. E.g., KY0063.bio is a platform .BIO file for Skull Canyon.

Project	Product code
Hades Canyon	NUC8i7HNK
Skull Canyon	NUC6i7KYK

So, on the above NUCs, run the following command to change the logo image:

VCUSTW64 -BIO=NUC_specific_platform.bio -LOGO=D:\Logo\Intel_NUC_Default_307x230.jpg

4.7 -SMBIOS -CURRENT

1) Meaning: Get current SMBIOS settings.
2) Requirements: None.
3) Example:
   VCUSTW64 -SMBIOS -CURRENT

4.8 -SMBIOS switch=<string>|<value>

1) Meaning: Update or clear a SMBIOS setting.
2) Requirements:
   - String must be 1-40 ASCII characters. If user wants to delete a current string setting, set it as a space in double quotation marks (“ ”).
   - The Chassis type: x, xx, xh, xxh, where x is a hex digit (0-9, a-f, or A-F). Its value must >=00h and <=7Fh. If user wants to delete the current Chassis type, set it to 1h which means other or none.
   - The OEM filed number must be consecutive with BIOS. E.g., if OEM1 already exists in BIOS,
     - We can update OEM1 or set OEM2:
       - *VCUSTW64 -SMBIOS -OEM1=new_oem1*
       - *VCUSTW64 -SMBIOS -OEM2=oem2*
     - If we want to set OEM3, we must set OEM2 first.
       (1) Of course we can:
       Step1. *VCUSTW64 -SMBIOS -OEM2=oem2*
       Step2. After OEM2 string exists in BIOS, set OEM3:
       VCUSTW64 -SMBIOS -OEM3=oem3
       (2) We can also:
       Step1. Create a queue BIO file that contains the OEM2 string.
       VCUSTW64 -SMBIOS -OEM2=oem2 -QUEUE=queue.bio
       Step2. Add the OEM3 string to the queue BIO file. Since queue.bio already contains the OEM2 string, OEM3 is considered consecutive.
       VCUSTW64 -SMBIOS -OEM3=oem3 -QUEUE=queue.bio
       Step3. So far, queue.bio contains both OEM2 and OEM3 strings. Then update BIOS with queue.bio.
       VCUSTW64 -BIO=queue.bio
3) Examples:
   VCUSTW64 -SMBIOS -SM=Intel
   VCUSTW64 -SMBIOS -CT=23
   VCUSTW64 -SMBIOS -CT=5h
   VCUSTW64 -SMBIOS -OEM1=Intel_board

5 Switches

5.1 -BIO=<filename>

1) Meaning: -BIO switch is used to input an existing .bio file.
2) Requirements:
   - The .bio file specified must exist.
   - The file extension ".bio" must be appended to the end of the filename.
   - The .bio file must follow the format of a capsule file. Otherwise, it will be considered invalid.
3) Example:
   - Update BIOS with a .bio file downloaded from Intel Download Center.
     VCUSTW64 -BIO=KY0055.bio
   - Update BIOS with a .bio file created in previous steps.
     VCUSTW64 -OA3 -CURRENT -QUEUE=queue.bio
     VCUSTW64 -BIO=queue.bio
   - Update BIOS with the existingAndValid.bio and a custom SMBIOS setting: system manufacturer.
     VCUSTW64 -BIO=existingAndValid.bio -SMBIOS -SM=Intel

5.2 -QUEUE=<filename>

1) Meaning: -QUEUE switch is used to save user custom settings and/or an existing .bio file to a new .bio file. If there's no -APPLYQUEUE behind, it will NOT update BIOS immediately by rebooting right away.
   One useful usage of -QUEUE is that it appends changes to a .bio which you can choose to flash at a later time.
4) Requirements:
   - The file extension ".bio" must be appended to the end of the filename.
   - If -QUEUE is used together with -BIO like one of the following, no further requirements for the file specified after -QUEUE. It can be non-existing or invalid. Because it will be overwritten anyway.
     - *VCUSTW64 -BIO=existing.bio -QUEUE=queue.bio*
     - *VCUSTW64 -BIO=existing.bio -QUEUE=queue.bio -APPLYQUEUE*
     - *VCUSTW64 -BIO=existing.bio -COMMAND -QUEUE=queue.bio*
     - *VCUSTW64 -BIO=existing.bio -COMMAND -QUEUE=queue.bio -APPLYQUEUE*
   - If -QUEUE is used without -BIO and with a -COMMAND like.
     - *VCUSTW64 -COMMAND -QUEUE=queue.bio*
     - *VCUSTW64 -COMMAND -QUEUE=queue.bio -APPLYQUEUE*

     queue.bio can be non-existing. If queue.bio exists, it must be valid and the new custom setting will append to the end of queue.bio.
   - If -QUEUE is used only with -APPLYQUEUE, the file specified after -QUEUE must exist and valid.
     - *VCUSTW64 -QUEUE=queue.bio -APPLYQUEUE*
2) Examples:
   - Add a custom setting to an existing BIO file and save it as a new BIO file (queue.bio will be overwritten if already exists).

VCUSTW64 -BIO=existing.bio -SMBIOS -SM=Intel -QUEUE=queue.bio

- Add a custom setting to the end of an existing BIO file.
  VCUSTW64 -SMBIOS -SM=Intel -QUEUE=existing.bio

5.3 -APPLYQUEUE

1) Meaning: -APPLYQUEUE switch is intended to be used after -QUEUE. Its purpose is to apply the newly created .bio file, i.e., to update BIOS with the newly created .bio file immediately. It can also be used without -QUEUE. In this case, it acts as if a temp .bio file is created and used to update BIOS immediately. But the temp .bio file won't be saved.
2) Requirements: None.
3) Examples:
   - Add a custom setting to an existing BIO file and save it as a new BIO file (queue.bio) and apply the newly created BIO file immediately.
     VCUSTW64 -BIO=existing.bio -SMBIOS -SM=Intel -QUEUE=queue.bio -APPLYQUEUE

Note: -QUEUE and -APPLYQUEUE are not stand-alone commands.

6 Examples

6.1 How to save multiple custom settings to one .bio file without an immediate reboot and apply all the changes later in just one reboot

1) We can use -QUEUE to save multiple custom settings one by one to a .bio file without an immediate reboot. The .bio file can be used to flash BIOS at a later stage.

   Step 1. *VCUSTW64 -SMBIOS -SM=Intel* ***-QUEUE=queue1.bio***
   Step 2. *VCUSTW64 -OA3 -CURRENT* ***-QUEUE= queue1.bio***
   Step 3. *VCUSTW64 -BOOTCOUNT -COUNT=25* ***-QUEUE= queue1.bio***
   Step 4. *VCUSTW64 -LOGO=nuc.jpg* ***-QUEUE= queue1.bio***

   Here is the explanation of each step above.

   Step 1. Create a new queue1.bio to save the custom SMBIOS system manufacturer information.
   Step 2. Append the current OA3 data in the system to queue1.bio.
   Step 3. Append a custom boot count value to queue1.bio.
   Step 4. Append a custom logo image to queue1.bio.

   So far, queue1.bio contains four custom settings and no reboot occurs, which allows us to use queue1.bio to flash BIOS when it's the right time.

2) Basically, there are two ways to apply an existing .bio file with an immediate reboot.
   We can either use -BIO:

   Step 5. *VCUSTW64* ***-BIO=queue1.bio***

   Or use -QUEUE together with -APPLYQUEUE:

   Step 5. *VCUSTW64* ***-QUEUE=queue1.bio -APPLYQUEUE***

   Either way above will trigger a reboot right away to flash BIOS with queue1.bio.

6.2 How to add a custom setting to an existing .bio and save it as a new .bio

We can use -BIO to input an existing .bio file and use -QUEUE to save it as a new one.

Step 1. *VCUSTW64* ***-BIO=existing.bio*** *-CONFIG -UQI=07ad -VALUE=0* ***-QUEUE=new.bio***

This command does the following:

- Append a custom BIOS setting to the end of the existing.bio.

- Save it as the new.bio.

Please note that:

- The existing.bio is left unchanged.
- No reboot is triggered.

We can flash BIOS with the new.bio immediately (reboot right away) by specifying -APPLYQUEUE after -QUEUE:

Step 1. *VCUSTW64* ***-BIO=existing.bio*** *-CONFIG -UQI=07ad -VALUE=0* ***-QUEUE=new.bio -APPLYQUEUE***

7 Meaning of each combination of command and switches

① -BIO ② -COMMAND ③ -QUEUE ④ -APPLYQUEUE

No.	Combination	Example	Description	A new .bio file created & saved?	Trigger reboot right away?
1 ①	-BIO	-BIO=bio.bio	Update BIOS with an existing .bio file. If bio.bio exists and is valid, then BIOS will be updated with it. Otherwise, error message will be shown.	No	Yes
2 ②	-COMMAND	-SMBIOS -SM=Intel	Change a setting. Change the manufacturer of the SMBIOS system information to “Intel”.	No	Yes
3 ③	-QUEUE	-QUEUE=myBIO.bio	Invalid command. -queue is not a stand-alone command		
4 ④	-APPLYQUEUE	-APPLYQUEUE	Invalid command. -applyqueue is not a stand-alone command		
5 ① ②	-BIO -COMMAND	-BIO=bio.bio -SMBIOS -SM=Intel	Update BIOS with an existing .bio file & change a setting. If bio.bio exists and is valid, BIOS will be updated with it and the setting	No	Yes

			specified by the SMBIOS SM will also be changed. Otherwise, error message will be shown.		
6 ① ③	-BIO -QUEUE	-BIO=bio.bio -QUEUE=queue.bio	Copy an existing .bio file. If bio.bio exists and is valid, it will be copied and saved to queue.bio. If queue.bio does not exist, it will be created. If already exists, it will be overwritten.	Yes	No
7 ① ④	-BIO -APPLYQUEUE	-BIO=bio.bio -APPLYQUEUE	Same as #1.	No	Yes
8 ② ③	-COMMAND -QUEUE	-SMBIOS -SM=Intel -QUEUE=queue.bio	Save a custom setting to a .bio file. If queue.bio does not exist, it is created to save a custom SMBIOS system manufacturer information “Intel”. If queue.bio already exists and is valid, the existing content will still be there, and the custom setting is appended. If queue.bio exists but is invalid, error message will be shown.	Yes	No
9 ② ④	-COMMAND -APPLYQUEUE	-SMBIOS -SM=Intel -APPLYQUEUE	Same as #2.	No	Yes
10	-QUEUE	-QUEUE=bio.bio	Same as #1.	No	Yes

③ ④	-APPLYQUEUE	-APPLYQUEUE			
11 ① ② ③	-BIO -COMMAND -QUEUE	-BIO=bio.bio -SMBIOS -SM=Intel -QUEUE=queue.bio	Add a custom setting to an existing .bio file and save it as a new one. If bio.bio exists and is valid, queue.bio is created (overwritten if already exists) based on bio.bio. queue.bio reserves everything within bio.bio and has a custom SMBIOS system manufacturer info appended at the end. Otherwise, error message will be shown.	Yes	No
12 ① ② ④	-BIO -COMMAND -APPLYQUEUE	-BIO=bio.bio -SMBIOS -SM=Intel -APPLYQUEUE	Same as #5.	No	Yes
13 ① ③ ④	-BIO -QUEUE -APPLYQUEUE	-BIO=bio.bio -QUEUE=queue.bio -APPLYQUEUE	Same as #6 and update BIOS immediately with queue.bio.	Yes	Yes
14 ② ③ ④	-COMMAND -QUEUE -APPLYQUEUE	-SMBIOS -SM=Intel -QUEUE=queue.bio -APPLYQUEUE	Same as #8 & update BIOS immediately with queue.bio.	Yes	Yes
15 ① ② ③ ④	-BIO -COMMAND -QUEUE -APPLYQUEUE	-BIO=bio.bio -SMBIOS -SM=Intel -QUEUE=queue.bio -APPLYQUEUE	Same as #11 & update BIOS immediately with queue.bio.	Yes	Yes