Aptio* V Integrator Tool - iDmiEdit

User Guide

January 2024



You may not use or facilitate the use of this document in connection with any infringement or other legal analysis concerning ASUS products described herein. You agree to grant ASUS a non-exclusive, royalty-free license to any patent claim thereafter drafted which includes subject matter disclosed herein.

No license (express or implied, by estoppel or otherwise) to any intellectual property rights is granted by this document.

All information provided here is subject to change without notice. Contact your ASUS representative to obtain the latest ASUS product specifications and roadmaps.

The products described may contain design defects or errors known as errata which may cause the product to deviate from published specifications. Current characterized errata are available on request.

ASUS and the ASUS logo are trademarks of ASUSTeK Computer Incorporated.
Copyright © 2023, ASUSTEK Computer Inc. All rights reserved.

Contents

1.0 Introduction 5

2.0 iDmiEdit User Guide 6

2.1 Overview 6

2.2 Requirements 6

2.2.1 Supported Operating Systems 6

2.3 iDmiEdit Usage 7

2.3.1 Command Line Switches 7

2.3.2 Display the Value of a Single SMBIOS Field 8

2.3.3 Change a Single SMBIOS Field with the Command Line 9

2.3.4 Change Multiple SMBIOS Fields with a Script File (SET.DMS) 10

Revision History

Date	Revision	Description
December 2019	1.0	Initial release.
May 2020	2.0	Additional command line examples.
December 2020	3.0	Tiger Lake support
January 2020	4.0	Additional Tiger Lake support
October 2021	5.0	Fixed issue: “49-Error: A Platform condition has prevented executing”, caused by some Crucial memory SO-DIMMs.
January 2024	6.0	Asus rebrand update

1.0 Introduction

ASUS® Aptio* V Integrator Tools are 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), changing the Logo, etc. of ASUS® NUC AptioV based products.

The table describes the purpose of each of the tools.

Tool	Description
iFlashV	iFlashV is a command line tool that allows you to: • Update the system BIOS • Update a logo image • Insert a specific OEM activation key
iDmiEdit	iDmiEdit (Desktop Management Interface Edit) allows you to modify strings associated with SMBIOS tables.
iCHLogo	iCHLogo allows you to replace the default Intel logo image with a custom image.
iSetupCfg	iSetupCfg is a command line tool which provides you an easy way to update NVRAM variables from within the EFI, Linux*, or Windows*-based environment. You can: • Extract variables directly from the BIOS. • Change settings using a text editor and then update the BIOS with the custom settings.

2.0 iDmiEdit User Guide

2.1 Overview

iDmiEdit (Desktop Management Interface Edit) allows you to modify strings associated with SMBIOS tables. You can modify the following SMBIOS tables with iDmiEdit:

- Type 1: System Information
- Type 3: System Enclosures or Chassis
- Type 11: OEM Strings

2.2 Requirements

2.2.1 Supported Operating Systems

	• iDmiEditWin32.exe is supported on Windows 32-bit operating systems. It requires both, amifldrv32.sys and amigendrv32.systo work correctly (both are included in the download). • iDmiEditWin64.exe is supported on Windows 64-bit operating systems. It requires both, amifldrv32.sys and amigendrv64.systo work correctly (both are included in the download). Make sure the following files are in the same directory: • If Windows 64-bit: ▪ amifldrv64.sys ▪ amigendrv64.sys ▪ iDmiEditWin64.exe • If Windows 32-bit: ▪ amifldrv32.sys ▪ amigendrv32.sys ▪ iDmiEditWin32.exe iDmiEdit for Windows is run from a Command Prompt in Administrator mode: 1. Click the Windows button and type **Command Prompt.** 2. Right-click on Command Prompt and select **Run as Administrator.**

iDmiEdit for EFI	▪ iDmiEditEfi32.efi is supported in EFI shell. ▪ iDmiEditEfi64.efi: is supported in EFIx64 shell.
iDmiEdit for Linux	▪ iDmiEditLnx32 is supported in Linux 32-bit operating systems. ▪ iDmiEditLnx64 is supported in Linux 64-bit operating systems.

2.3 iDmiEdit Usage

2.3.1 Command Line Switches

Option	Description
/all	Outputs current Type 1, Type 3, and Type 11 SMBIOS strings to the screen
/all [fileame.txt]	Outputs current Type 1, Type 3, and Type 11 SMBIOS strings to a text file
/dms set.dms	Create a configuration file that can be used to update multiple SMBIOS settings.
set.dms	Writes SMBIOS changes from a text file of modified SMBIOS settings.
/dumpall	Outputs detailed SMBIOS data to the screen
/dumpall [filename.txt]	Outputs detailed SMBIOS data to a text file
/dump 1 /dump 3 /dump 11	Outputs detailed SMBIOS data for the specific Table Type: 1: System Information 3: System Enclosure or Chassis 11: OEM Strings

Options

System Information (Type 1)

Option	Description
/sm	Reads system manufacturer */sm ["string"] - writes a new value for system manufacturer*
/sp	Reads system product */sp [8-bit value] - writes a new value for system product*
/sv	Reads system version */sv ["string"] - writes a new value for system version*
/ss	Reads system serial number */ss ["string"] - writes a new value for system serial number*
/su	Reads system UUID */su [16-byte value] - writes a new value for system UUID*
/su auto	Generate system UUID and update automatically

/sk	Reads SKU number */sk ["string"] - writes a new value for SKU number*
/sf	Reads family name */sf ["string"] - writes a new value for family name*

System Enclosure or Chassis (Type 3)

Option	Description
/cm	Reads chassis manufacturer */cm ["string"] - writes a new value for chassis manufacturer*
/ct	Reads chassis type */ct [8-bit value] - writes a new value for chassis type*
/cv	Reads chassis version */cv ["string"] - writes a new value for chassis version*
/cs	Reads chassis serial number */cs ["string"] - writes a new value for chassis serial number*
/ca	Reads chassis tag */ca ["string"] - writes a new value for chassis tag*
/co	Reads chassis OEM-defined value */co [32-bit value] - writes a new value for chassis OEM-defined value*
/ch	Reads chassis height */ch [8-bit value] - writes a new value for chassis height*
/cpc	Reads chassis power cords number */cpc [8-bit value] - writes a new value for chassis power cords number*
/csk	Reads chassis SKU Number */csk ["string"] - writes a new value for chassis SKU number*

OEM String (Type 11)

Option	Description
/os	Reads #th OEM string */os [<number><"string">] – writes #th OEM string*

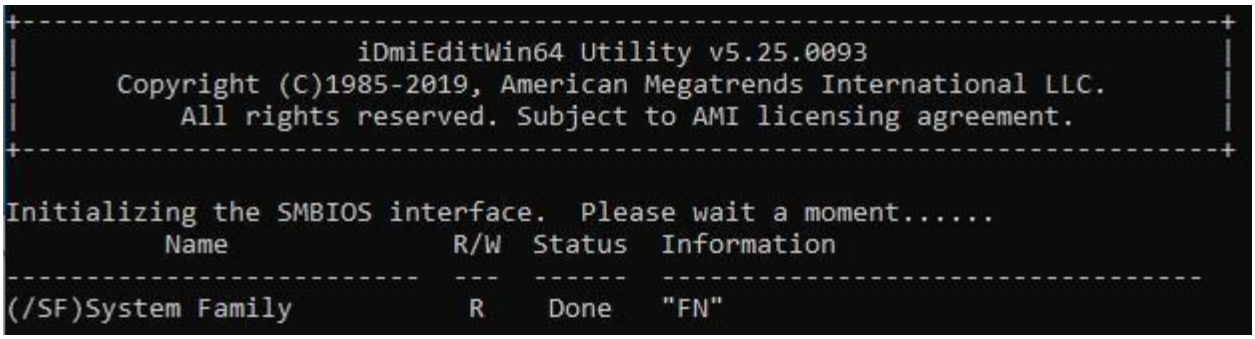
2.3.2 Display the Value of a Single SMBIOS Field

Example command lines to display the System Family:

OS / Environment	Example command syntax
Windows 32-bit	iDmiEditWin32.exe /sf
Windows 64-bit	iDmiEditWin64.exe /sf
UEFI 32-bit	iDmiEditEfi32.efi /sf

UEFI 64-bit	iDmiEditEfi64.efi /sf
Linux 32-bit	iDmiEditLnx32 /sf
Linux 64-bit	iDmiEditLnx64 /sf

Results:

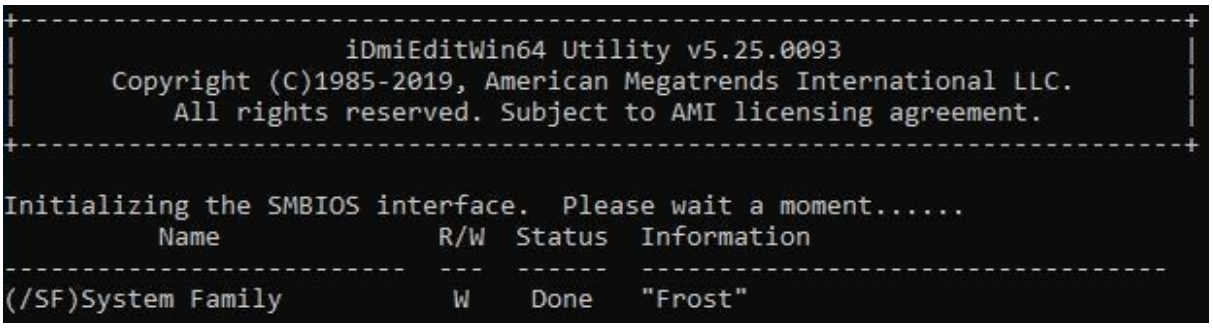


2.3.3 Change a Single SMBIOS Field with the Command Line

Example command lines to change the System Family string to a new value:

OS / Environment	**Example command syntax**
Windows 32-bit	iDmiEditWin32.exe /sf “Frost”
Windows 64-bit	iDmiEditWin64.exe /sf “Frost”
UEFI 32-bit	iDmiEditEfi32.efi /sf “Frost”
UEFI 64-bit	iDmiEditEfi64.efi /sf “Frost”
Linux 32-bit	iDmiEditLnx32 /sf “Frost”
Linux 64-bit	iDmiEditLnx64 /sf “Frost”

Results:



The command line can be used to change one or multiple SMBIOS fields at a time. Examples:

- iDMiEditWin64.exe /sm "MyManufacturer"

- iDMiEditWin64.exe /ss "1234567890"
- iDMiEditWin64.exe /sm "MyManufacturer" /ss "1234567890"

2.3.4 Change Multiple SMBIOS Fields with a Script File (SET.DMS)

If you want to change multiple SMBIOS settings at one time, follow these steps.
Step 1: Create the input script file:

OS / Environment	Example command syntax
Windows 32-bit	iDmiEditWin32.exe /dms set.dms
Windows 64-bit	iDmiEditWin64.exe /dms set.dms
UEFI 32-bit	iDmiEditEfi32.efi /dms set.dms
UEFI 64-bit	iDmiEditEfi64.efi /dms set.dms
Linux 32-bit	iDmiEditLnx32 /dms set.dms
Linux 64-bit	iDmiEditLnx64 /dms set.dms

The file name must be **SET.DMS**. The following is an example of a SET.DMS input script file:

- **[System]**

  Manufacturer = Intel Corporation
  Product = NUC8i7BEK
  Version = 6.22
  SerialNum = 123455
  SKU = SKU12345
  Family = Fam12345
  UUID = 0123456789ABCDEF0123456789ABCDEF

- **[Chassis]**

  Manufacturer = Intel Corporation
  Version = 1.22
  SerialNum = 12222
  TagNum = 122212
  ChassisType = 0F
  ChassisOEM = FFFF0000

- **[OemString]**

  String = Intel Corporation
  String = http://www.intel.com
  String = xxxxx

Step 2: Use a text editor to edit the SET.DMS file with the desired SMBIOS changes. You only need to include the items you want to change.

The SET.DMS file must include at least one SMBIOS table entry – System, Chassis, or OemString. Each SMBIOS table entry contains the SMBIOS table type name followed by the string to be edited, which is separated by <space>=<space>.

Example:

```
[System]
Family = Frost Canyon

[Chassis]
SerialNum = 123456
TagNum = ABCDEF
```

Step 3: Update SMBIOS with the script file. Run the following command:

OS / Environment	**Example command syntax**
Windows 32-bit	iDmiEditWin32.exe set.dms
Windows 64-bit	iDmiEditWin64.exe set.dms
UEFI 32-bit	iDmiEditEfi32.efi set.dms
UEFI 64-bit	iDmiEditEfi64.efi set.dms
Linux 32-bit	iDmiEditLnx32 set.dms
Linux 64-bit	iDmiEditLnx64 set.dms