

Aptio 5.x DMIEDIT User Guide

AMI SOFTWARE UTILITY USER GUIDE

REVISION 1.20 – DECEMBER 27, 2023

NDA REQUIRED

Revision History

Date	Revision	Description
2013-11-01	1.00	Initial document created and update content to the latest released of DmiEdit.
2014-01-24	1.01	Add a command to read/write the offset 04h of type 0.
2017-02-24	1.02	Remove DOS support. Add Windows 10 support into OS list. Add a command to read/write the offset 12h of type 3.
2017-04-28	1.03	Add answer for Windows digitally signed driver.
2017-08-03	1.04	Add note for DOS not support. Correct execution file name in OS list.
2017-12-26	1.05	Add command /CH to read/write the offset 11h of Type3 for Height data field.
2018-05-17	1.06	Add Linux Driver chapter for the description of Pre-Requisites and Signing Driver. Add answer for Linux error when secure boot is enabled.
2019-02-18	1.07	Add requirement of admin privileges for Windows. Add requirement of root access for Linux.
2019-03-05	1.08	Add firmware requirement of SMBIOS module.
2019-05-09	1.09	Add answer for Linux Segmentation Fault or Windows Stopped Working. Add answer for Windows driver information to support WSMT. Updated the steps in Signing Driver for Linux.
2019-06-28	1.10	AMI copyright message updated.
2020-06-08	1.11	Update Supported OS list.
2020-09-22	1.12	Update firmware requirement of SMBIOS module. Add command /PSNH /PATH /PPNH for Type 4 items with handle. Add additional Windows driver description.
2021-05-13	1.13	Add requirement for make and libelf-dev in Linux prerequisites. Add Linux driver case in Linux prerequisites.
2022-05-24	1.14	Update Supported OS list. Update firmware requirement of SMBIOS module. Update Linux Pre-Requisites. Update Linux signature driver reference link.
2022-06-16	1.15	Revise Windows driver description.
2022-06-17	1.16	Remove the Driver information for Windows 7.
2022-09-16	1.17	Formatted document as per the AMI template.
2023-05-15	1.18	Update firmware requirement of SMBIOS module. Reviewed the document format.
2023-12-20	1.19	Added a method for generating signing keys during the Linux driver signing process. Update firmware requirement of SMBIOS module for 3.6/3.7 Spec.
2023-12-27	1.20	Reviewed the document format

Disclaimer

This publication contains proprietary information which is protected by copyright. No part of this publication may be reproduced, transcribed, stored in a retrieval system, translated into any language or computer language, or transmitted in any form whatsoever without the prior written consent of the publisher, American Megatrends International LLC. American Megatrends International LLC. retains the right to update, change, modify this publication at any time, without notice.

For Additional Information

Call American Megatrends International LLC. at 1-800-828-9264 for additional information.

Limitations of Liability

In no event shall American Megatrends be held liable for any loss, expenses, or damages of any kind whatsoever, whether direct, indirect, incidental, or consequential, arising from the design or use of this product or the support materials provided with the product.

Limited Warranty

No warranties are made, either expressed or implied, with regard to the contents of this work, its merchantability, or fitness for a particular use. American Megatrends assumes no responsibility for errors and omissions or for the uses made of the material contained herein or reader decisions based on such use.

Trademark and Copyright Acknowledgments

Copyright © 2023 American Megatrends International LLC. All Rights Reserved.

American Megatrends International LLC.

5555 Oakbrook Parkway

Suite 200

Norcross, GA 30093 (USA)

All product names used in this publication are for identification purposes only and are trademarks of their respective companies.

Table of Contents

Chapter 1 Introduction 1

Overview 1

Requirements 1

Supported Operation System 1

Firmware Requirements 2

Input File 3

Getting Started 3

Installation 3

Command line switches 3

Creating Input File (SET.DMS) 9

Running DMIEdit Program 10

Chapter 2 Features 11

Edit SMBIOS Strings 11

Edit SMBIOS Strings Process 11

Chapter 3 Linux Driver 12

Linux Pre-Requisites 12

Signing Driver and Enrolling Public Key to the System 14

Chapter 4 FAQ 20

Linux shows error when secure boot is enabled 20

Windows shows driver error when WSMT support is enabled 20

Segmentation Fault when kernel 3.14.40 with XEN 4.2.4 20

"XXX has stopped working" or "Segmentation Fault" error 21

Document Information

Technical Support

AMI provides technical support only for AMI products licensed directly from AMI.

Web Site

We invite you to visit our website at ami.com.

Purpose

This document provides information to use the AptioV DMIEDIT tool to update the SMBIOS.

Audience

The intended audiences are Generic BIOS Engineers, OEM Engineers, and Aptio Customers.

Chapter 1 Introduction

Overview

DMIEdit stands for Desktop Management Interface Edit. It allows you to modify strings associated with SMBIOS tables. This utility works with Aptio firmware with SMBIOS support.

You can modify the following SMBIOS tables with DMIEdit:

- Bios Information (Type 0)
- System (Type 1)
- Base Board (Type 2)
- Chassis (Type 3)
- Processor Information (Type 4)
- OEM String (Type 11)
- System Configuration Options (Type 12)
- Portable Battery (Type 22)
- System Power Supply (Type 39)

***Note**: The System Firmware must support the SMBIOS specification.*

Requirements

Supported Operation System

DMIEdit for Windows

AMIDEWIN (AMIDEWIN.EXE/AMIDEWINx64.EXE) support by the following Windows:

- Microsoft® Windows® 8
- Microsoft® Windows® 8.1
- Microsoft® Windows® Server 2016
- Microsoft® Windows® Server 2019
- Microsoft® Windows® Server 2022
- Microsoft® Windows® 10
- Microsoft® Windows® 11
- Microsoft® Windows® PE

It is also likely to run normally under unlisted Windows environments.

AMIDEWIN requires the specific Windows driver.

***Due to System IO access, Windows version requires administrator privileges and executes with "Run as Administrator" option.**

DMIEdit for EFI

AMIDEEFI (AMIDEEFI.EFI/AMIDEEFIx64.EFI) supports EFI Shell Environment.

DMIEdit for Linux

AMIDELNX (amidelnx_32/amidelnx_64) support Linux operation system.

The following versions of Linux distribution listed which version AMI testing:

- Ubuntu (20.04)
- Red Hat
- Fedora (29)
- openSUSE
- Debian (9.6)
- CentOS (7.3)

It is also likely to run normally under unlisted Linux environments.

***Due to System IO access, Linux version requires root authority.**

Note*: DOS version stops supporting in DMIEDIT 5.18 or later version.*

Firmware Requirements

DMI Editor (Desktop Management Interface Edit) program requires that the host system Firmware is an AMI AptioV and above.

SMBIOS module

- Smbios_07 label and above for SMBIOS 3.0 Spec.
- Smbios_12 label and above for WSMT support.
- Smbios_14 label and above for SMBIOS 3.1 and SMBIOS 3.1.1 Spec.
- Smbios_19 label and above for SMBIOS 3.2 Spec.
- Smbios_23 label and above for SMBIOS 3.3 Spec.
- Smbios_26 label and above for Non-register based SW SMI support.
- Smbios_30 label and above for SMBIOS 3.4 Spec.
- Smbios_31 label and above for SMBIOS 3.5 Spec.
- Smbios_33 label and above for SMBIOS 3.6 Spec.
- Smbios_37 label and above for SMBIOS 3.7 Spec.

Input File

DMI Editor (Desktop Management Interface Edit) requires that an input file used.

***Note**: Creating an input file is explained in more details in Chapter Getting Started.*

Getting Started

Installation

To install the DMIEdit for EFI (Desktop Management Interface Edit), copy the DMIDEEFI.EFI/ DMIDEEFIx64.EFI executable file to the hard disk drive or any other storage location of the system that will use it. In the case of MS Windows platform, make sure all the files (tool and driver) are in the same directory.

Command line switches

Command line switches give the users the flexibility to change individual SMBIOS table fields. If the users want to change in multiple groups at a time, then input script file (shown below) is the best method. The different command line switches are as following:

AMIDEWIN <Command 1>

or

AMIDEWIN [Option 1] [Option 2]…

Command

/ALL [FileName]	Output SMBIOS string to screen or file.
/DMS [FileName]	Create configuration file.
/DUMPALL [FileName]	Output all SMBIOS data to screen or file.
/DUMP # [#] …	Read Type # data.

Options

Part 0. System (Type 0)

/IVN ["String"] Read/Write BIOS vendor name.

/IV ["String"] Read/Write BIOS version.

/ID ["String"] Read/Write BIOS release date.

Part 1. System (Type 1)

/SM ["String"] Read/Write system manufacturer.

/SP ["String"] Read/Write system product.

/SV ["String"] Read/Write system version.

/SS ["String"] Read/Write system serial number.

/SU [16 Bytes] Read/Write system UUID.

/SU AUTO Generate system UUID and update automatically.

/SK ["String"] Read/Write SKU number.

/SF ["String"] Read/Write family name.

Part 2-1 Base Board (Type 2)

/BM ["String"] Read/Write baseboard manufacturer.

/BP ["String"] Read/Write baseboard product.

/BV ["String"] Read/Write baseboard version.

/BS ["String"] Read/Write baseboard serial number.

/BT ["String"] Read/Write Asset Tag.

/BLC ["String"] Read/Write location in Chassis.

Part 2-2 Base Board (Type 2)

/BMH [device handle#] ["String"]

Read/Write baseboard manufacturer with device handle number.

/BPH [device handle#] ["String"]

Read/Write baseboard product with device handle number.

/BVH [device handle#] ["String"]

Read/Write baseboard version with device handle number.

/BSH [device handle#] ["String"]

Read/Write baseboard serial number with device handle number.

/BTH [device handle#] ["String"]

Read/Write Asset Tag with device handle number.

/BLCH [device handle#] ["String"]

Read/Write location in Chassis with device handle number.

Part 3-1 Chassis (Type 3)

/CM ["String"]	Read/Write chassis manufacturer.
/CT [8-Bits value]	Read/Write chassis type.
/CV ["String"]	Read/Write chassis version.
/CS ["String"]	Read/Write chassis serial number.
/CA ["String"]	Read/Write chassis tag.
/CO [32-Bits value]	Read/Write chassis OEM-defined value.
/CH [8-Bits value]	Read/Write chassis Height.
/CPC [8-Bits value]	Read/Write chassis Power Cords number.
/CSK ["String"]	Read/Write chassis SKU Number.

Part 3-2 Chassis (Type 3)

/CMH [device handle#] ["String"]

Read/Write chassis manufacturer. with device handle number.

/CTH [device handle#] ["String"]

Read/Write chassis type. with device handle number.

/CVH [device handle#] ["String"]

Read/Write chassis version. with device handle number.

/CSH [device handle#] ["String"]

Read/Write chassis serial number with device handle number.

/CAH [device handle#] ["String"]

Read/Write chassis tag with device handle number.

/COH [device handle#] [32-Bits value]

Read/Write chassis OEM-defined value with device handle number.

/CHH [device handle#] [8-Bits value]

Read/Write chassis Height with device handle number.

/CPCH [device handle#] [8-Bits value]

Read/Write chassis Power Cords number with device handle number.

/CSKH [device handle#] ["String"]

Read/Write chassis SKU Number with device handle number.

Part 4-1 Processor Information (Type 4)

/PSN ["String"] Read/Write Processor Information serial number.

/PAT ["String"] Read/Write Processor Information asset tag.

/PPN ["String"] Read/Write Processor Information part number.

Part 4-2 Processor Information (Type 4)

/PSNH [device handle#] ["String"]

Read/Write Processor Information serial number with device handle number.

/PATH [device handle#] ["String"]

Read/Write Processor Information asset tag with device handle number.

/PPNH [device handle#] ["String"]

Read/Write Processor Information part number with device handle number.

Part 5. OEM String (Type 11)

/OS [<Number><"String">] Read/Write #th OEM string.

Part 6. OEM String (Type 12)

/SCO [<Number><"String">] Read/Write #th OEM string.

Part 7. Portable Battery (Type 22)

/PBL [device handle#] ["String"]

Read/Write Portable Battery Location.

/PBM [device handle#] ["String"]

Read/Write Portable Battery Manufacturer.

/PBD [device handle#] ["String"]

Read/Write Portable Battery Manufacturer Date.

/PBS [device handle#] ["String"]

Read/Write Portable Battery Serial Number.

/PBN [device handle#] ["String"]

Read/Write Portable Battery Device Name.

/PBCH [device handle#] [8-Bits value]

Read/Write Portable Battery Device Chemistry.

/PBCA [device handle#] [16-Bits value]

Read/Write Portable Battery Design Capacity.

/PBV [device handle#] [16-Bits value]

Read/Write Portable Battery Design Voltage.

/PBSV [device handle#] ["String"]

Read/Write Portable Battery SBDS Version Number.

/PBE [device handle#] [8-Bits value]

Read/Write Portable Battery Maximum Error.

/PBSN [device handle#] [16-Bits value]

Read/Write Portable Battery SBDS Serial Number.

/PBSD [device handle#] [16-Bits value]

Read/Write Portable Battery SBDS Manufacturer Date.

/PBSC [device handle#] ["String"]

Read/Write Portable Battery SBDS Device Chemistry.

/PBCM [device handle#] [8-Bits value]

Read/Write Portable Battery Design Capacity Multiplier.

/PBO [device handle#] [32-Bits value]

Read/Write Portable Battery OEM-Specific.

Part 8. System Power Supply (Type 39)

/PU [device handle#] [8-Bits value]

Read/Write Power supply unit group.

/PL [device handle#] ["String"]

Read/Write Power supply location.

/PD [device handle#] ["String"]

Read/Write Power supply device name.

/PM [device handle#] ["String"]

Read/Write Power supply manufacturer.

/PS [device handle#] ["String"]

Read/Write Power supply serial number.

/PT [device handle#] ["String"]

Read/Write Power supply asset tag number.

/PN [device handle#] ["String"]

Read/Write Power supply model part number.

/PR [device handle#] ["String"]

Read/Write Power supply revision level.

/PP [device handle#] [4-Bits value]

Read/Write Power supply max power capacity

/PC [device handle#] [4-Bits value]

Read/Write Power supply characteristics.

/PVH [device handle#] [4-Bits value]

Read/Write Power supply voltage probe handle.

/PDH [device handle#] [4-Bits value]

Read/Write Power supply cooling dev. handle.

/PCH [device handle#] [4-Bits value]

Read/Write Power supply current probe handle.

Creating Input File (SET.DMS)

Use a text editor to create the input file. The file name must be SET.DMS. The SET.DMS input file must include at least one SMBIOS table entry. Each SMBIOS table entry contains the SMBIOS table type name followed by the strings to be edited, which is separated by <space>=<space>.

The following is an example of a SET.DMS input file:

```
[System]
Manufacturer = AMI
Product = Dummy
Version = 6.22
SerialNum = 123455
SKU = SKU12345
Family = Fam12345
UUID = 0123456789ABCDEF0123456789ABCDEF

[BaseBoard]
Manufacturer = AMI
Product = Dummy
Version = 1.22
SerialNum = 122333

[Chassis]
Manufacturer = AMI
Version = 1.22
SerialNum = 12222
TagNum = 122212
ChassisType = 83
ChassisOEM = FFFF0000

[OemString]
String = AMI
String = http://www.ami.com
```



```
String = xxxxx
```

***Note**:*

The DMIEdit program can fail if it encounters an error in the SET.DMS input file.

For variable length block fields of SMBIOS tables, the value of this field should be in hexadecimal numbers, which helps in determining the length of the block. The hexadecimal digits representation is exactly the twice the actual size required to store that value. For example, the UUID has block length of 16 bytes hence the number of hexadecimal digits in the script/command line required should be 32.

Running DMIEdit Program

For EFI shell, the argument should be as follows:

```
AMIDEEFI SET.DMS <ENTER>
```

For Windows, copy all the four files in the same folder as the input script file, open the command console prompt in the folder and type the following convention to run:

```
C:\>AMIDEWIN SET.DMS <ENTER>
```

For using Command line switches, one command line or multiple options can be used at a time:

```
C:\>AMIDEWIN /IVN AMI

C:\>AMIDEWIN /BS 1234567890
```

or

```
C:\>AMIDEWIN /IVN AMI /BS 1234567890
```

Chapter 2 Features

Edit SMBIOS Strings

DMI Editor allows users to modify SMBIOS strings that are associated with the SMBIOS tables.

Edit SMBIOS Strings Process

To modify SMBIOS strings, follow the steps outlined in the table below:

Step	Description
Step 1	Flash the new Firmware.
Step 2	Boot the system in EFI shell, Linux or Windows.
Step 3	Create a SET.DMS input file as described in Chapter Getting Started.
Step 4	Run the DMIEdit program as described in Chapter Getting Started.

Chapter 3 Linux Driver

Linux Pre-Requisites

1. Log in Linux as root otherwise use sudo (if permitted).
2. The compiler suite (gcc, make, libelf-dev) must be installed. If these packages are not installed, the driver CANNOT be built.
3. For most of the distributions, tool will generate driver without any notification, if it doesn't exist you need to install kernel sources. Also if Initmem fails, Please follow point 4.
4. Kernel sources must be installed, *CONFIGURED*, and then compiled. Following are steps to do this:

    a. Find Running Kernel's Configuration File:

    To configure the sources, simply change to the kernel source directory (typically **/lib/modules/$(uname -r**)/build). If it doesn't exist, you need to install kernel source.

    Typically, the reference configuration for the kernel can be found in the /boot directory with filename '.**config', 'kernel.config**', or **'vmlinux-2.4.18-3.config'**. Type **'uname -a'** and use the configuration filename that best matches the output from **'uname -a'**. Also, check for /dev/mem directory existence. If it doesn't exist, you need to install kernel sources.

    Normally it comes with the installation unless if the option is deselected.

    On some distributions Red Hat for instance, there is a config directory under **/lib/modules/$(uname -r)/build**.

    Copy this configuration file into the root of the Linux kernel source tree (usually it is **/lib/modules/$(uname -r)/build**). This file must be renamed to ".config"(dot config).

    b. Make Your Driver (amifldrv_mod.o):

    For most distribution, the command /MAKEDRV to build driver is:

    ```
    amidelnx_32 /MAKEDRV
    ```

or

amidelnx_64 /MAKEDRV

If your Linux's kernel source tree is under /lib/modules/$(uname -r)/build, instead of being in the default path '/lib/modules/$(uname -r)/build', then add a KERNEL flag:

amidelnx_32 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build

or

amidelnx_64 /MAKEDRV KERNEL=/lib/modules/$(uname -r)/build

If KERNEL is omitted, the default path is /lib/modules/$(uname -r)/build.

This should work for MOST distributions.

c. Make Your Driver from driver source files (amifldrv_mod.o):

Using command /GENDRV, it will generate driver source files to specific directory.

amidelnx_32 /GENDRV [Option]

or

amidelnx_64 /GENDRV [Option]

Where,

[Option]: Specific kernel source 'KERNEL=XXXX' same as the /MAKEDRV

Generate files as outlined below:

File Name	Description
Ver. 5.24 or previous versions	
amiwrap.c	Driver source code.
amiwrap.h	Driver header.
amifldrv.o_shipped	Object file for driver. (Or amifldrv.o)
Makefile	Make file
Ver. 5.25 or later versions	

---

amiwrap.c	Driver source code.
amifldrv.h	Driver header.
amifldrvdefs.h	Driver struct define.
amifldrv.o_shipped	Object file for driver. (Or amifldrv.o)
Makefile	Make file

For most distribution, the command to build the driver is:

```
make
```

If your Linux's kernel source tree is under /lib/modules/$(uname -r)/build, instead of being in the default path '/lib/modules/$(uname -r)/build', then add a KERNEL flag:

```
make KERNEL=/lib/modules/$(uname -r)/build
```

If KERNEL is omitted, the default is /lib/modules/$(uname -r)/build.

This should work for MOST distributions.

d. Check Your Build:

Check the version of running Linux kernel with 'uname -r'.

Check the version of amifldrv_mod.o with 'modinfo amifdrv_mod.o'.

If they mismatch, you will need to select the correct configuration file (.config), rebuild your kernel, and then rebuild your driver as described in steps a, b, c and d.

5. The case of Linux driver:

	Secure Boot Enabled	**Secure Boot Disabled**
WSMT is supported	Driver required	Driver no required
Can access file path:/dev/mem	Driver required	Driver no required
Run Time Memory Hole support	Driver required	Driver no required

Signing Driver and Enrolling Public Key to the System

The following prerequisites are needed on the build system to sign the driver:

1. Login to Linux OS as root otherwise use sudo.
2. The compiler suite (gcc) must be installed. If it's not installed, the driver cannot be built.
3. OpenSSL: Needed to generate cryptographic keys. OpenSSL tool can be downloaded from

   https://www.openssl.org
4. Perl interpreter: Needed to run the signing script. Perl tool can be downloaded from

   https://www.perl.org

Follow the below steps to sign the driver:

1. Boot to Linux OS.
2. Generate a Public and Private key pair using below openssl command: >

   openssl req -x509 -new -nodes -utf8 -sha256 -days 36500 -batch -config configuration_file.config -outform DER -out public_key.der -keyout private_key.priv

   OR

   openssl req -x509 -new -nodes -utf8 -sha256 -days 36500 -batch -config configuration_file.config -outform DER -out public_key.der -keyout private_key.priv -addext "extendedKeyUsage=codeSigning"

***Note**: The configuration file configuration_file.config must be created with the required information before running the command. A sample configuration file is shown below. The values in <> must be filled with actual values.*

configuration_file.config:

```
 [ req ]
default_bits = 4096
distinguished_name = req_distinguished_name
prompt = no
string_mask = utf8only
x509_extensions = myexts

[ req_distinguished_name ]
O = <organization_name>
```

```
CN = <organization_name> Signing Key
emailAddress = <email_address>

[ myexts ]
basicConstraints=critical,CA:FALSE
keyUsage=digitalSignature
subjectKeyIdentifier=hash
authorityKeyIdentifier=keyed
```

3. Build driver using below command. The driver will be generated in the current directory with name amifldrv_mod.o.

```
> amidelnx_64 /MAKEDRV
```

4. Execute below command to sign driver with the key generated in step 2.

```
> perl /usr/src/kernels/$(uname -r)/scripts/sign-file sha256
private_key.priv public_key.der amifldrv_mod.o
```

or

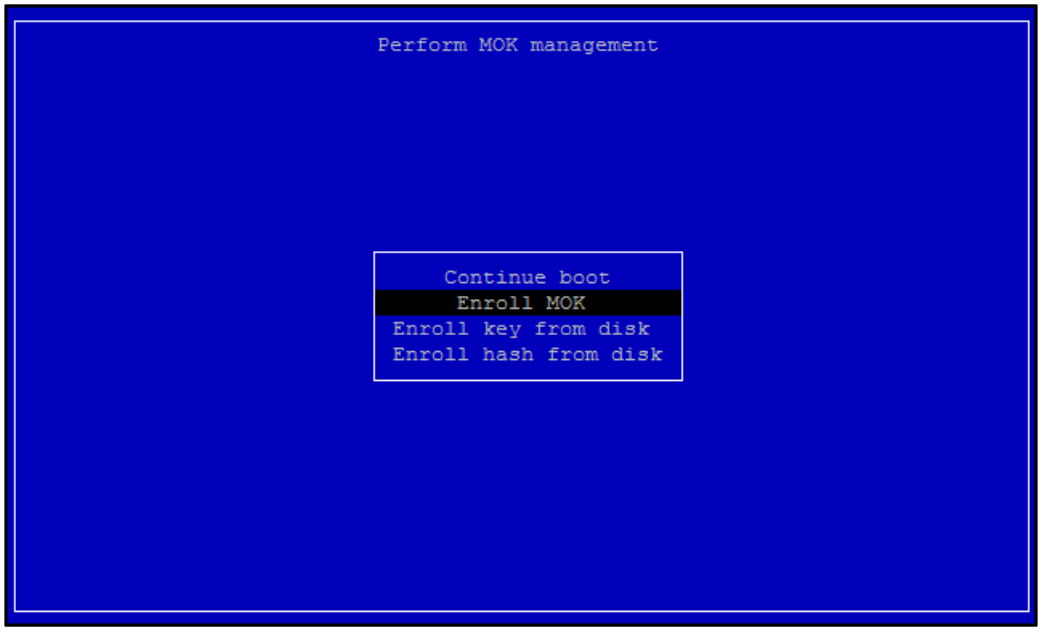
```
> /usr/src/kernels/$(uname -r)/scripts/sign-file sha256
private_key.priv public_key.der amifldrv_mod.o
```

5. Request addition of public key to MOK list using mokutil. The command will prompt a password which will be needed during public key enrollment in next step.

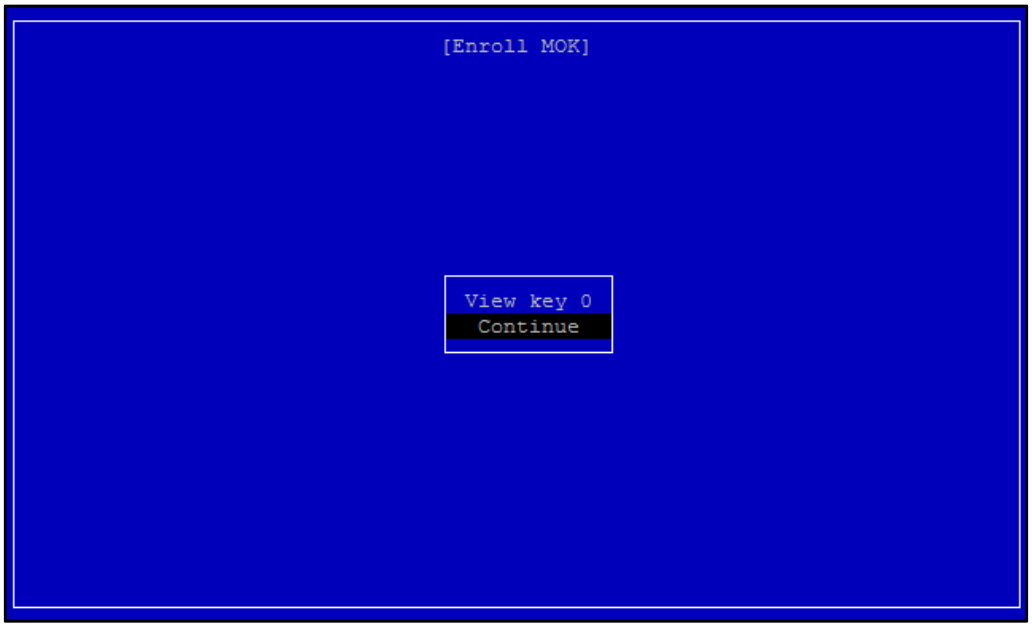
```
> mokutil --import public_key.der
```

6. Reboot the system which will launch MOK manager application to complete public key enrollment.

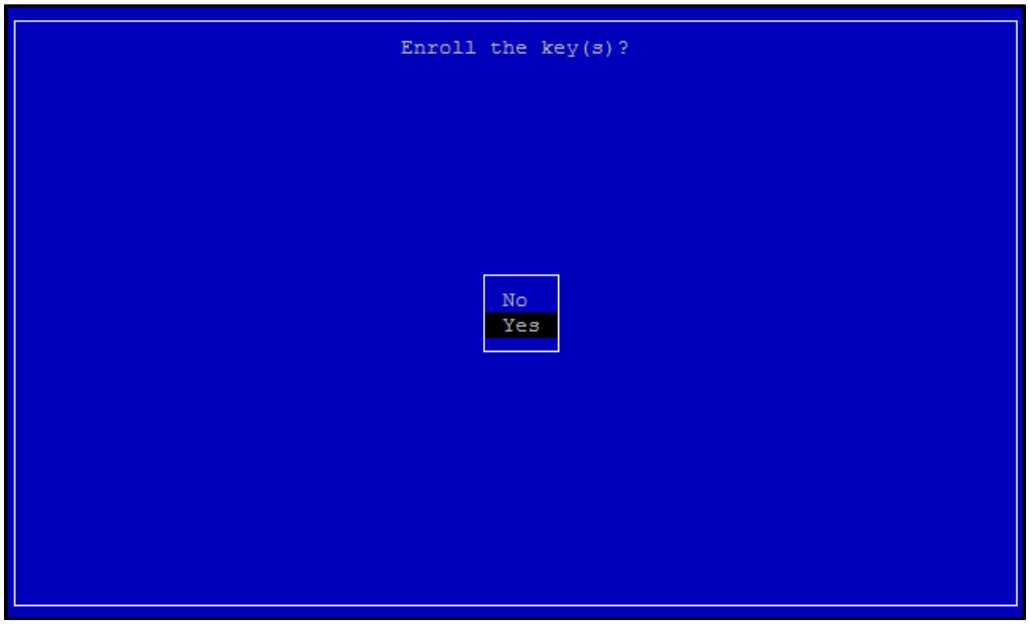
    i. Select Enroll MOK.



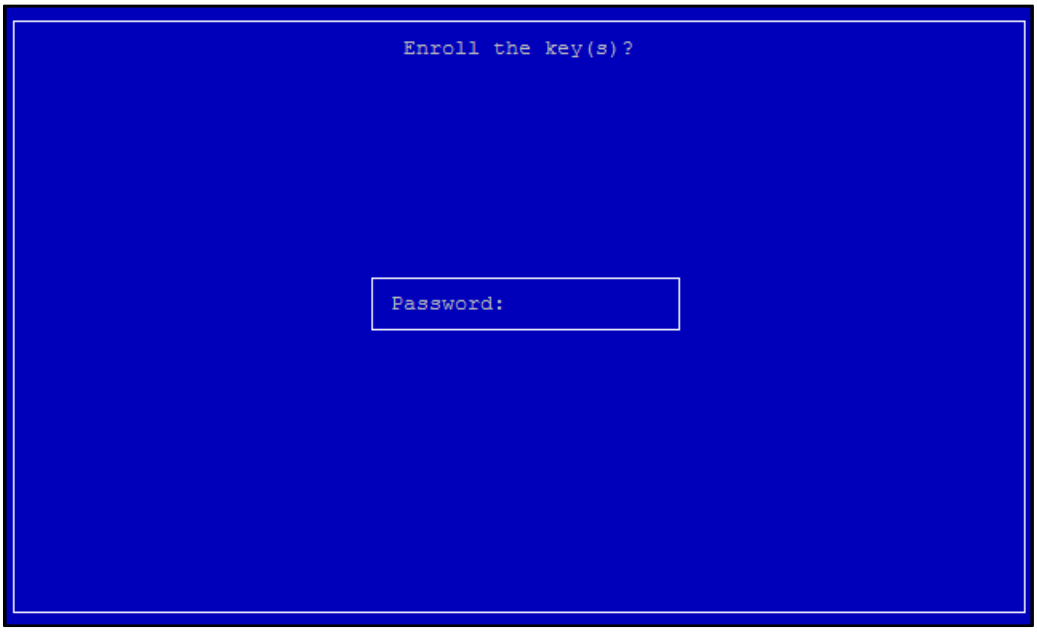
ii. Select Continue.



iii. Select Yes.



iv. .Input step 5 password.



7. Once the public key enrollment is done, Boot to OS and execute below command to ensure the newly added key is available in system key ring.

```
> keyctl list %:.system_keyring
```

or

```
> keyctl list %:.builtin_trusted_keys
```

8. Install signed driver using insmod command.

```
> insmod amifldrv_mod.o
```

9. Ensure it is loaded successfully using lsmod command.

Reference:

https://access.redhat.com/documentation/en-us/red_hat_enterprise_linux/8/html/managing_monitoring_and_updating_the_kernel/signing-kernel-modules-for-secure-boot_managing-monitoring-and-updating-the-kernel

Chapter 4 FAQ

Linux shows error when secure boot is enabled

The following error messages appear because a signed driver is required when secure boot enable.

71 - Error: Linux does not support Auto Build Driver when Secure Boot Enable. Or

Segmentation fault (with an unsigned driver)

***Note**: For the singing driver, please refer to Signing Driver and Enrolling Public Key to the System.*

Windows shows driver error when WSMT support is enabled

If OS is Win 10 build 10240 (version 1507), for WSMT support need to install Windows hot fix, KB3081436.

https://docs.microsoft.com/en-us/windows-hardware/drivers/install/driver-signing

Segmentation Fault when kernel 3.14.40 with XEN 4.2.4

Please follow the steps below to operate the configuration, and try again.

1. Check if the system runs under X11, a.k.a GUI mode. If so, switch to console mode with the following command.

   ```
   # systemctl set-default multi-user.target
   ```

   Remember to restart system after configured.

2. Check if Dom0 has enough free memory.

   ```
   # xl info
   ```

   Check the item "free_memory", make sure it is larger than 1024. If the number is lower than 1024, use the command

   ```
   # xm mem-set Domain-0 1024
   ```



You don't need to restart the system after this step.

3. Set virtual CPU number to 1 of Domain-0 in XEN.

```
# xm vcpu-set Domain-0 1
```

4. Run tool.

"XXX has stopped working" or "Segmentation Fault" error

Common reason which caused the error is that: Tool access to invalid/prohibit memory area.

Since tool uses/needs driver file to access memory, most cases we met that caused the error are:

(1) Don't use administrator/root privilege to execute tool.
(2) Use unmatched version between driver file and tool.
(3) BIOS has enabled "Secure Boot" but doesn't sign driver file for it.
(4) Use some newer/older tool version that is not compatible with some motherboard.

For (1), if users are using:

- Windows version
    - Please open a command line with administrator privilege, then execute tool in it.
- Linux version
    - Please use "sudo" command to execute tool.

For (2), if users are using:

- Windows version
    - Driver file (amigendrvXX.sys) can be found in tool deliverables (Bundled with tool).
    - Please don't use other source of .sys file from other version or other AMI tools.
- Linux version
    - Driver file (amifldrv_mod.o) will be auto-built by tool while executing tool.
    - Driver file will be generated in the same folder of tool.
    - Recommended to create an empty folder and put tool in it before executing, tool will generate the pure/clean driver file in folder.

For (3), please check to Signing Driver and Enrolling Public Key to the System section to sign driver file.

For (4), if checked (1) (2) (3) but still see error:

1. Firstly, please try the latest tool version.
2. Secondary, please try the previous tool version (Which can run well in that motherboard previously):
    - If the previous version also fails, please check BIOS code.
    - If the previous version runs well, and also latest version still fails, please kindly contact AMI.