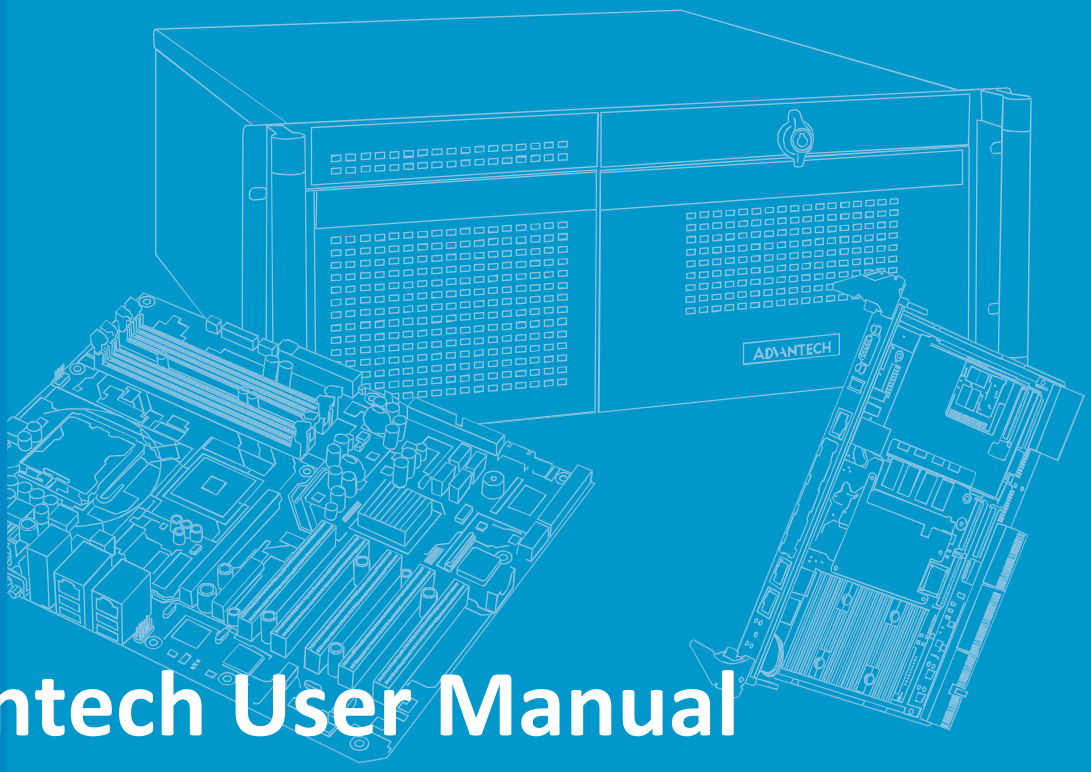

Advantech User Manual

Server iManager

Platform Device Identifier

For Network Device

Revision 13.0

Copyright 2024 All rights reserved.

Advantech Corp. Ltd.

Revision History

Date [mm/dd/yyyy]	Revision	Modifications
03/13/2024	13.0	PDI for network device official release - Update -support list based on v2.02
12/06/2023	12.0	Rename Network Device Identifier to Platform Device Identifier for Network device Base on platform_device_identifier_v2.00 (libapdi version: 02.00, apdi version: 02.00) - Update -support list based on v2.00
9/11/2023	11.0	- Update -support list based on v1.44
6/21/2023	10.0	- Update -support list based on v1.42
4/3/2023	9.00	- Update -support list based on v1.40
1/31/2023	8.00	Edition 8 officical release
12/23/2022	7.01	- Update –support list. - Add description in 1.1, 1.2.3 and 2.1.
10/10/2022	7.00	Edition 7 officical release
09/01/2022	6.02	- Update –support list.
03/01/2022	6.01	- Add Advanced LAN Bypass (I2C) support
08/07/2021	6.00	Edition 6 officical release
07/05/2021	5.01	Update model name part in 1.2.3
10/07/2020	5.00	Edition 5 official release.
30/6/2020	4.01	Added max support NIC number. Added 0./apdi <command> -novlan in 1.2.16 Update header file.
24/9/2019	4.00	Official release
19/7/2019	3.01	Updated API return values.
18/6/2019	3.0	Official release.
17/05/2019	2.01	Updated 1.1 Support for Eth Devices. Updated 1.2.8 ./apdi -lbpcu.cfg command to support new format of LBPCU version 1.30. Update 1.1 Support for Eth Devices.
19/03/2018	2.00	Official release on related product website.
11/05/2018	1.09	Onboard segment number will be assigned from 100 to 255. Add -sort_slotname command.
09/03/2018	1.08	Changed name to Server iManager Network Device Identifier
08/14/2018	1.07	Changed config format from PCIe_BUS/NMC_SLOT="05-06/Slot 5" To PCIe_SW_DEV/NMC_SLOT="05-06/Slot 5" and added example in Appendix3.1.
08/09/2018	1.06	Added -smbios_cfg command which can be used to input the correct smbios type41/9 records.

Advantech Copyright 2024. All rights reserved.

		Added -sysinfo command and apdi_print_system_info() API
07/23/2018	1.05	Changed figure format.
07/19/2018	1.04	Modified description of -support command.
07/09/2018	1.02	Added -sys_NMC_cfg command for -legacy command. Added -support command to list to all supported Eth devices.
06/13/2018	1.00	Initial version

Advantech Copyright 2024. All rights reserved.

Table of Contents

1. SERVER IMANAGER PLATFORM DEVICE IDENTIFIER 7
1.1 SUPPORTED ETH DEVICES 7
1.2 COMMANDS 8
1.2.1 ./apdi 8
1.2.2 ./apdi –V 10
1.2.3 ./apdi -a 10
1.2.4 ./apdi -d <eth_name> 13
1.2.5 ./apdi -f <feature> 13
1.2.6 ./apdi -p <bus:dev.fun> 13
1.2.7 ./apdi -o <file name> 14
1.2.8 ./apdi -lbpcu.cfg 15
1.2.9 ./apdi <command> -legacy 15
1.2.10 ./apdi <command> -config <file name> 16
1.2.11 ./apdi <command> -legacy -sys_NMC_cfg <file name> 17
1.2.12 ./apdi <command> -smbios_cfg <file name> 18
1.2.13 ./apdi <command> -sort_slotname 19
1.2.14 ./apdi -support 20
1.2.15 ./apdi -sysinfo 23
1.2.16 ./apdi <command> -novlan 23
2. LIBAPDI –SERVER IMANAGER PLATFORM DEVICE IDENTIFIER LIBRARY 24
2.1 SUPPORTED DISTRIBUTIONS 24
2.2 HEADER FILE 24
2.3 API COMMAND LIST 27
2.4 ERROR CODES 27
2.5 HOW TO USE THE API 28
2.6 API FUNCTIONS 28
2.6.1 Message Output Level 28
2.6.1.1 apdi_err apdi_set_message_level(int msg_level) 28
2.6.2 Open/Close 28
2.6.2.1 apdi_err apdi_open(void) 28
2.6.2.2 apdi_err apdi_close (void) 30
2.6.3 Information 30
2.6.3.1 apdi_err apdi_get_lib_version(unsigned char *major, unsigned char *minor) 30
2.6.4 Output results 30
2.6.4.1 apdi_print_scan_result(char* eth_name, char* feature, char* bus_dev_fun) 30
2.6.4.2 apdi_fprint_scan_result (char* file_name) 31
2.6.4.3 apdi_output_lbpcu_cfg(void) 31
2.6.4.4 apdi_print_supported_NMC(void) 31
2.6.4.5 apdi_print_system_info (void) 31
2.6.4.6 apdi_print_storage_scan_result(char* storage_name, char flag) 32
2.6.4.7 apdi_dump_efi_var_to_file(char *efi_var, char* file_name) 32
2.6.4.8 apdi_print_storage_mapping_string(char *storage_name, char *slot_name) 32
3. APPENDIX 33
3.1 CONFIG FILE EXAMPLE FOR "–CONFIG" PARAMETER 33
3.2 SYS_NMC_CFG CONFIG FILE EXAMPLE 37

Advantech Copyright 2024. All rights reserved.

3.3 SMBIOS_CFG CONFIG FILE EXAMPLE 38

List of Figures

Figure 1: Usage of 'apdi' Command 10
Figure 2: Usage of 'apdi -V' Command 10
Figure 3: Usage of 'apdi -a' Command 11
Figure 4: Usage of 'apdi -d' Command 13
Figure 5: Usage of 'apdi -f' Command 13
Figure 6: Usage of 'apdi -p' Command 13
Figure 7: Usage of 'apdi -o' Command 14
Figure 8: Usage of 'apdi -lbpcu.cfg' Command 15
Figure 9: Usage of 'apdi -a -legacy' Command 15
Figure 10: Usage of 'apdi -a -config apdi.cfg' Command 16
Figure 11: error message of 'apdi -a -config apdi.cfg' Command 16
Figure 12: Usage of 'apdi -a -legacy -sys_NMC_cfg' Command 17
Figure 13: Usage of 'apdi -a -smbios_cfg smbios.cfg' Command 18
Figure 14: Usage of 'apdi -a -sort_slotname' Command 19
Figure 15: Usage of 'apdi -support' Command 22
Figure 16: Usage of 'apdi -sysinfo' Command 23

Advantech Copyright 2024. All rights reserved.

List of Tables

Table 1: API Function 'apdi_set_message_level' 28
Table 2: API Function 'apdi_open' 29
Table 3: API Function 'apdi_close' 30
Table 4: API Function 'apdi_get_lib_version' 30
Table 5: API Function 'apdi_print_scan_result' 30
Table 6: API Function 'apdi_fprint_scan_result' 31
Table 7: API Function 'apdi_output_lbpcu_cfg' 31
Table 8: API Function 'apdi_print_supported_NMC' 31
Table 9: API Function 'apdi_print_system_info' 31
Table 10: API Function 'apdi_print_storage_scan_result' 32
Table 11: API Function 'apdi_dump_efi_var_to_file' 32
Table 12: API Function 'apdi_print_storage_mapping_string' 32

Advantech Copyright 2024. All rights reserved.

1. SERVER IMANAGER PLATFORM DEVICE IDENTIFIER

Server iManager Platform Device Identifier is designed to solve a common problem on x86 platforms about how to link the “eth device” in Linux systems with the actual physical port in the system. Server iManager Platform Device Identifier shows the info which the user is most interested in (eth devices’ name, bus, id, subsystem id, slot name, panel name, feature, and model name of Advantech). Advantech provides a Linux command line utility that uses the libapdi library. The utility can also be used in scripts, rather than requiring full integration of the libapdi library into applications.

NOTE:

- Utility and library supports maximum 512 physical NICs and an unlimited number of virtual NIC ports.
- Platform device identifier also support storage device, to get more information please check “Server iManager Platform Device Identifier (Storage Device)” User manual.

1.1 Supported Eth Devices

The latest support list is always released with Server iManager Package as below

https://www.advantech.com/en/support/details/software-utility?id=1-1YC9KKF

You can find the download link in Advantech Software portal.
https://campaign.advantech.online/en/Cloud-IoT/software/

Advantech Copyright 2024. All rights reserved.

1.2 Commands

1.2.1 ./apdi

When Server iManager Platform Device Identifier is started without any parameters, usage information is displayed as shown below:

```
# ./apdi
Usage:  apdi {op}
{op}:   The operation you want to do.
===== ethdev related =====
-a      List all eth devices information (Onboard MGMT, LAN, SFP+ -> NMC Slots -> PCIe Slots).
        e.g. apdi -a
        e.g. apdi -a -config apdi.cfg -legacy
-d <eth name> Output specific eth device's bus:dev.fun, slot, front panel name and feature.
        e.g. apdi -d eth0
        e.g. apdi -d eth0 -config apdi.cfg -legacy
-f <feature name> List all eth devices those have this feature.
        e.g. apdi -f POE
             apdi -f A_BYPASS
             apdi -f L_BYPASS
             apdi -f N/A
             apdi -f NCSI
             apdi -f 1PPS
             apdi -f NCSI/1PPS
             apdi -f A_BYPASS_I2C
             apdi -f N/A -config apdi.cfg -legacy
-p <bus:dev.fun> Output specific eth device's slot, front panel name and feature.
        e.g. apdi -p 5e:00.3
        e.g. apdi -p 5e:00.3 -config apdi.cfg -legacy
-o <file_path> Output identified result to <file_path>.
        e.g. apdi -o output
        e.g. apdi -o output -config apdi.cfg -legacy
-lbpcu.cfg Output lbpcu.cfg for lan bypass control utility(lbpcu).
        e.g. apdi -lbpcu.cfg
-legacy Identify old NMC card without correct subsystem id, this function may show wrong info
when using new cards.
        This command could be added after any ethdev related command.
        e.g. apdi -a -legacy
        e.g. apdi -a -legacy -config apdi.cfg
-config <file_path> Read file to get more info about the platform, like internal PCIe slot,
PCIe switch or onboard NIC's feature based on HW design.
        This command could be added after any ethdev related command.
        e.g. apdi -a -config apdi.cfg
        e.g. apdi -a -config apdi.cfg -legacy
```

Advantech Copyright 2024. All rights reserved.

```
-sys_NMC_cfg <file_path> Read file to tell apdi what NMC cards are in the platform when using
-legacy command.

        -legacy command could show NMC-0807/0803, if user only use NMC-0803, then the user can
use NMC config file to tell

        apdi there is only NMC-0803, then apdi will only show NMC-0803 without NMC-0807.

        This command should be added only after -legacy command.

        e.g. apdi -a -config apdi.cfg -legacy -sys_NMC_cfg nmc_name.cfg

-smbios_cfg <file_path> Read file to get correct smbios type 9/41 records of the platform,

        apdi will use the records from file and ignore the records which were read from real
BIOS.

        This command is useful when the BIOS contains incorrect type 9/41 records.

        This command could be added after any ethdev related command.

        e.g. apdi -a -smbios_cfg smbios.cfg

-support Display all supported devices.

        e.g. apdi -support

-sysinfo Display system info.

        e.g. apdi -sysinfo

        e.g. apdi -sysinfo -smbios_cfg smbios.cfg -config apdi.cfg -legacy -sys_NMC_cfg
nmc_name.cfg

-sort_slotname Sort the eth devices by the slot name.

        e.g. apdi -a -sort_slotname

        e.g. apdi -a -sort_slotname -smbios_cfg smbios.cfg -config apdi.cfg -legacy -
sys_NMC_cfg nmc_name.cfg

-diag <smbios_entry_point_file_path> <dmi_file_path> specify smbios_entry_point file and DMI
file

        and ignore real SMBIOS data in memory and sysfs

        (/sys/firmware/dmi/tables/smbios_entry_point and /sys/firmware/dmi/tables/DMI).

        This command is only for debug and diag.

        This command could be added after any ethdev related command.

        e.g. apdi -a -diag /root/a /root/b

-novlan Ignore virtual NIC port in system.

        This command could be added after any ethdev related command.

        e.g. apdi -a -novlan

===== storage related =====

WARNING: USB Storage type not supported.

-s      List all root storage device information (nvme/sdx)

        e.g. apdi -s

        e.g. apdi -s -storage_mapping_cfg apdi_storage_mapping.cfg

-sb     List all storage device information under /sys/class/block

        e.g. apdi -sb

        e.g. apdi -sb -storage_mapping_cfg apdi_storage_mapping.cfg

-sd <storage name> List specific storage device's information.

        e.g. apdi -sd nvme1

        e.g. apdi -sd nvme1 -storage_mapping_cfg apdi_storage_mapping.cfg

-sc     List storage mapping.
```

Advantech Copyright 2024. All rights reserved.

```
        e.g. apdi -sc
        e.g. apdi -sc -storage_mapping_cfg apdi_storage_mapping.cfg
-psms <storage name> <slot name> Print storage mapping string based on input data.
        e.g. apdi -psms nvme1 "slot 8"
        e.g. apdi -psms nvme1 "slot 8" -storage_mapping_cfg apdi_storage_mapping.cfg
-storage_mapping_cfg <file_path> Read file to get onboard port and external storage slot name
mapping.
        apdi will use this information to map the nvmex/sdx to external slot.
        This command could be added after -s, -sb, -sd and -sc command.
        e.g. apdi -s -storage_mapping_cfg apdi_storage_mapping.cfg
===== general =====
-V      Display the version
-defi <EFI Variable> <file_path> Dump EFI Variable to specific file. -storage_mapping_cfg will
be ignored.
        e.g. apdi -defi APDIStorageMapping-475d23f7-32b3-4803-8745-f8ce46d08d5a dump_cfg_file
NOTE: <file_path> means full file path which includes the file name
        e.g. apdi -a -config apdi.cfg
        e.g. apdi -a -config /root/configs/apdi.cfg
```

Figure 1: Usage of 'apdi' Command

1.2.2 ./apdi –V

Display version information for libapdi and utility:

```
root@localhost:~# ./apdi -V
libapdi version: 02.02
apdi version: 02.02
```

Figure 2: Usage of 'apdi -V' Command

1.2.3 ./apdi -a

Lists all eth device information (Onboard MGMT, LAN, SFP+, NMC Slots, PCIe Slots):

This command will print eth devices' name, bus, id, subsystem id, slot name, panel name, features, bypass segment number, and model name of Advantech. This command will show all eth devices in the system sorted by bus number (from low to high). All Virtual NIC ports will be shown after physical NIC ports. Note that this example add *–Legacy* and *–sys_NMC_cfg* parameters to show legacy NMCs on Legacy platform.

```
root@localhost:~# ./apdi -a -legacy -sys_NMC_cfg nmc_name.cfg
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg     N/A Onboard
enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       Onboard I210_1 (*)   Need cfg     N/A Onboard
enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard
enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard
enp94s0f0       5e:00.0     8086 1521 ffff 0000  00:0b:ab:bd:ab:aa  Slot 1               1 (*)                L_BYPASS     4   NMC-0806
enp94s0f1       5e:00.1     8086 1521 ffff 0000  00:0b:ab:bd:ab:ab  Slot 1               2 (*)                L_BYPASS     4   NMC-0806
enp94s0f2       5e:00.2     8086 1521 ffff 0000  00:0b:ab:bd:ab:ac  Slot 1               3 (*)                L_BYPASS     5   NMC-0806
enp94s0f3       5e:00.3     8086 1521 ffff 0000  00:0b:ab:bd:ab:ad  Slot 1               4 (*)                L_BYPASS     5   NMC-0806
```

Advantech Copyright 2024. All rights reserved.

```
enp96s0f0      60:00.0     8086 1521 ffff 0000  00:0b:ab:bd:aa:2a  Slot 1          5 (*)          L_BYPASS     6   NMC-0806
enp96s0f1      60:00.1     8086 1521 ffff 0000  00:0b:ab:bd:aa:2b  Slot 1          6 (*)          L_BYPASS     6   NMC-0806
enp96s0f2      60:00.2     8086 1521 ffff 0000  00:0b:ab:bd:aa:2c  Slot 1          7 (*)          L_BYPASS     7   NMC-0806
enp96s0f3      60:00.3     8086 1521 ffff 0000  00:0b:ab:bd:aa:2d  Slot 1          8 (*)          L_BYPASS     7   NMC-0806
enp100s0f0     64:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  Slot 2          1 (*)          N/A          N/A NMC-4001
enp100s0f1     64:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  Slot 2          2 (*)          N/A          N/A NMC-4001
enp102s0f0     66:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  Slot 2          3 (*)          N/A          N/A NMC-4001
enp102s0f1     66:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  Slot 2          4 (*)          N/A          N/A NMC-4001
enp216s0f0     d8:00.0     8086 1583 13fe 002e  00:0b:ab:f6:8e:c2  Slot 3          1 (*)          N/A          N/A NMC-4006
enp216s0f1     d8:00.1     8086 1583 13fe 0000  00:0b:ab:f6:8e:c3  Slot 3          2 (*)          N/A          N/A NMC-4006
enp218s0f0     da:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  Slot 4          1 (*)          L_BYPASS     16  NMC-0806
enp218s0f1     da:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  Slot 4          2 (*)          L_BYPASS     16  NMC-0806
enp218s0f2     da:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  Slot 4          3 (*)          L_BYPASS     17  NMC-0806
enp218s0f3     da:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  Slot 4          4 (*)          L_BYPASS     17  NMC-0806
enp220s0f0     dc:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  Slot 4          5 (*)          L_BYPASS     18  NMC-0806
enp220s0f1     dc:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  Slot 4          6 (*)          L_BYPASS     18  NMC-0806
enp220s0f2     dc:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  Slot 4          7 (*)          L_BYPASS     19  NMC-0806
enp220s0f3     dc:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  Slot 4          8 (*)          L_BYPASS     19  NMC-0806
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

Figure 3: Usage of 'apdi -a' Command

The panel name means the location of each eth device. For onboard devices, the panel name and slot name are the same by default. For NMC cards, the panel name means the location (port number) in the NMC card. The "*" character means the panel name is assigned by the utility automatically, which is sorted by bus and function number. Users can alter the config file to change the panel name as they want.

There are seven kinds of features supported in utility. If the "!" character is shown after a feature, this means there is conflict between the feature in the config file and the default for this eth device, please check the config file or use the -F command. This command will display the features in the config file. If the conflict is in advanced LAN bypass, then a segment number will be assigned from 200 to 255 (in the bus number order from low to high).

A_BYPASS	Advanced LAN Bypass (NIC)
A_BYPASS-C	Advanced LAN Bypass (NIC) used to control the bypass segment.
A_BYPASS_I2C	Advanced LAN Bypass (I2C)
L_BYPASS	Legacy LAN bypass.
POE	Power over Ethernet.
N/A	Not available.
Unknown	The eth device is not supported (ID not found).
Need cfg	This is an onboard eth device, users need to

Advantech Copyright 2024. All rights reserved.

	use the config file to see the utility features.

Bypass segment number will follow this table.

NICs	Segment number for legacy/advanced bypass
Onboard	100~255 (max)
Slot 1	4~7
Slot 2	8~11
Slot 3	12~15
Slot 4	16~19
Slot 5	20~23
Slot 6	24~27
Slot 7	28~31
Slot 8	32~35

Even if the slot/onboard NICs don't have a bypass feature, the seg numbers will be reserved (this means the numbers will not be used and skipped).

The seg numbers are assigned according to the slot number. If the user changes the slot names in apdi.cfg. Please make sure the last 2 characters are the original numbers. The segment numbers for onboard LANs or for a single slot are assigned by bus:dev.fun.

Different kinds of model name:

Onboard	This is a motherboard device. But no valid subsystem ID.
<NIC_ID>_FI <NIC_ID>_CO	This is a motherboard device. FI means fiber. CO means copper. e.g. I210_CO/I211_CO/I350_CO/X553_CO/ X553_FI/X710_FI/X722_FI
NMC-XXXX PCIE-XXXX	This is a NMC-XXXX/PCIE-XXXX device.
ID not found	Utility can't identify this device.

Examples when the utility can't identify the devices:

For these devices, the utility will show "unknown" in the features, and show "ID not found" in the model name. Notice that there are LTE eth devices which don't have an ID and subsystem ID.

Advantech Copyright 2024. All rights reserved.

```
root@localhost:~# ./apdi -a
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg     N/A Onboard
enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       Onboard I210_1 (*)   Need cfg     N/A Onboard
enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard
enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard
enp96s0f0       60:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  cfg Slot 1           1 (*)                N/A          N/A NMC-4001
enp96s0f1       60:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  cfg Slot 1           2 (*)                N/A          N/A NMC-4001
enp97s0f0       61:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  cfg Slot 1           3 (*)                N/A          N/A NMC-4001
enp97s0f1       61:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  cfg Slot 1           4 (*)                N/A          N/A NMC-4001
wwan0                                                                J6B2                 6 (*)                unknown      N/A ID not found
```

1.2.4 ./apdi -d <eth_name>

Output specific eth device's bus:dev.fun, slot, front panel name and feature:

```
root@localhost:~# ./apdi -d enp1s0
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg     N/A Onboard
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

Figure 4: Usage of 'apdi -d' Command

1.2.5 ./apdi -f <feature>

List all eth devices that have this feature:

Feature:/POE/A_BYPASS/L_BYPASS

```
root@localhost:~# ./apdi -f A_BYPASS -config apdi.cfg
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       LAN 1                A_BYPASS-C    0  Onboard
enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       LAN 2                A_BYPASS      0  Onboard
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

Figure 5: Usage of 'apdi -f' Command

1.2.6 ./apdi -p <bus:dev.fun>

Output specific (bus:dev.fun) eth device slots, front panel name and features:

```
root@localhost:~# ./apdi -p 01:00.0
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg     N/A Onboard
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

Figure 6: Usage of 'apdi -p' Command

Advantech Copyright 2024. All rights reserved.

1.2.7 ./apdi -o <file name>

Output identified result to <file name>:

```
root@localhost:~# ./apdi -o output -legacy -sys_NMC_cfg nmc_name.cfg
root@localhost:~# cat output
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg      Need cfg Onboard
enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       Onboard I210_1 (*)   Need cfg      Need cfg Onboard
enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        Onboard 10GBe (*)    Need cfg      Need cfg Onboard
enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        Onboard 10GBe (*)    Need cfg      Need cfg Onboard
enp94s0f0       5e:00.0     8086 1521 ffff 0000  00:0b:ab:bd:ab:aa  Slot 1               1 (*)                L_BYPASS      4   NMC-0806
enp94s0f1       5e:00.1     8086 1521 ffff 0000  00:0b:ab:bd:ab:ab  Slot 1               2 (*)                L_BYPASS      4   NMC-0806
enp94s0f2       5e:00.2     8086 1521 ffff 0000  00:0b:ab:bd:ab:ac  Slot 1               3 (*)                L_BYPASS      5   NMC-0806
enp94s0f3       5e:00.3     8086 1521 ffff 0000  00:0b:ab:bd:ab:ad  Slot 1               4 (*)                L_BYPASS      5   NMC-0806
enp96s0f0       60:00.0     8086 1521 ffff 0000  00:0b:ab:bd:aa:2a  Slot 1               5 (*)                L_BYPASS      6   NMC-0806
enp96s0f1       60:00.1     8086 1521 ffff 0000  00:0b:ab:bd:aa:2b  Slot 1               6 (*)                L_BYPASS      6   NMC-0806
enp96s0f2       60:00.2     8086 1521 ffff 0000  00:0b:ab:bd:aa:2c  Slot 1               7 (*)                L_BYPASS      7   NMC-0806
enp96s0f3       60:00.3     8086 1521 ffff 0000  00:0b:ab:bd:aa:2d  Slot 1               8 (*)                L_BYPASS      7   NMC-0806
enp100s0f0      64:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  Slot 2               1 (*)                N/A           N/A NMC-4001
enp100s0f1      64:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  Slot 2               2 (*)                N/A           N/A NMC-4001
enp102s0f0      66:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  Slot 2               3 (*)                N/A           N/A NMC-4001
enp102s0f1      66:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  Slot 2               4 (*)                N/A           N/A NMC-4001
enp216s0f0      d8:00.0     8086 1583 13fe 002e  00:0b:ab:f6:8e:c2  Slot 3               1 (*)                N/A           N/A NMC-4006
enp216s0f1      d8:00.1     8086 1583 13fe 0000  00:0b:ab:f6:8e:c3  Slot 3               2 (*)                N/A           N/A NMC-4006
enp218s0f0      da:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  Slot 4               1 (*)                L_BYPASS      16  NMC-0806
enp218s0f1      da:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  Slot 4               2 (*)                L_BYPASS      16  NMC-0806
enp218s0f2      da:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  Slot 4               3 (*)                L_BYPASS      17  NMC-0806
enp218s0f3      da:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  Slot 4               4 (*)                L_BYPASS      17  NMC-0806
enp220s0f0      dc:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  Slot 4               5 (*)                L_BYPASS      18  NMC-0806
enp220s0f1      dc:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  Slot 4               6 (*)                L_BYPASS      18  NMC-0806
enp220s0f2      dc:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  Slot 4               7 (*)                L_BYPASS      19  NMC-0806
enp220s0f3      dc:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  Slot 4               8 (*)                L_BYPASS      19  NMC-0806
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
(?) means CFG file changed the slot number of slot name or NO matched SMBIOS type9/41 record.
    APDI can't identify the slot number to output corresponding seg number.
```

Figure 7: Usage of 'apdi -o' Command

Advantech Copyright 2024. All rights reserved.

1.2.8 ./apdi -lbpcu.cfg

Output lbpcu.cfg for LAN bypass control utility (lbpcu):

This command will output an lbpcu.cfg file with all eth devices which are identified with advanced LAN bypass control features.

Note: supports output eth device name format since apdi version 1.20.

```
root localhost:~# ./apdi -lbpcu.cfg -config apdi.cfg
root@localhost:~# cat lbpcu.cfg
eth_name: eth0
SEGMENT: 1
BUS: 01
DEV: 00
FUN: 0

TYPE:PCIE
```

Figure 8: Usage of 'apdi -lbpcu.cfg' Command

1.2.9 ./apdi <command> -legacy

Identifies old NMC cards (not implemented Advantech subsystem ID), this function may show wrong info when using new cards:

This command could be added after any other command.

```
root@localhost:~# ./apdi -a -legacy
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg     N/A Onboard
enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       Onboard I210_1 (*)   Need cfg     N/A Onboard
enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard
enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard
enp96s0f0       60:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  Slot 1               1 (*)                N/A          N/A NMC-4001
enp96s0f1       60:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  Slot 1               2 (*)                N/A          N/A NMC-4001
enp97s0f0       61:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  Slot 1               3 (*)                N/A          N/A NMC-4001
enp97s0f1       61:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  Slot 1               4 (*)                N/A          N/A NMC-4001
enp99s0f0       63:00.0     8086 1522 ffff 0000  00:0b:ab:f1:2f:6e  Slot 2               1 (*)                unknown      N/A NMC-0108/0120
enp99s0f1       63:00.1     8086 1522 ffff 0000  00:0b:ab:f1:2f:6f  Slot 2               2 (*)                unknown      N/A NMC-0108/0120
enp99s0f2       63:00.2     8086 1522 ffff 0000  00:0b:ab:f1:2f:70  Slot 2               3 (*)                unknown      N/A NMC-0108/0120
enp99s0f3       63:00.3     8086 1522 ffff 0000  00:0b:ab:f1:2f:71  Slot 2               4 (*)                unknown      N/A NMC-0108/0120
enp217s0f0      d9:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  Slot 4               1 (*)                unknown      N/A NMC-0803/0806/0807
enp217s0f1      d9:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  Slot 4               2 (*)                unknown      N/A NMC-0803/0806/0807
enp217s0f2      d9:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  Slot 4               3 (*)                unknown      N/A NMC-0803/0806/0807
enp217s0f3      d9:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  Slot 4               4 (*)                unknown      N/A NMC-0803/0806/0807
enp218s0f0      da:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  Slot 4               5 (*)                unknown      N/A NMC-0803/0806/0807
enp218s0f1      da:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  Slot 4               6 (*)                unknown      N/A NMC-0803/0806/0807
enp218s0f2      da:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  Slot 4               7 (*)                unknown      N/A NMC-0803/0806/0807
enp218s0f3      da:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  Slot 4               8 (*)                unknown      N/A NMC-0803/0806/0807
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

Figure 9: Usage of 'apdi -a -legacy' Command

Advantech Copyright 2024. All rights reserved.

1.2.10 ./apdi <command> -config <file name>

Read file to get more info about the platform, PCIe switch, or onboard NIC features based on the HW design:

This command could be added after any other command.

The user can use this config file to assign a panel name for the onboard NIC and NMC cards.

```
root@localhost:~# ./apdi -a -legacy -config apdi.cfg
root@steve-MIC-6330:~/apdi_source# ./apdi -a -legacy -config apdi.cfg
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model
enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       LAN 1                A_BYPASS-C     0  Onboard
enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       LAN 2                A_BYPASS       0  Onboard
enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        SFP+1                N/A           N/A Onboard
enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        SFP+2                N/A           N/A Onboard
enp96s0f0       60:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  PCIE SLOT 1          p1                   N/A           N/A NMC-4001
enp96s0f1       60:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  PCIE SLOT 1          p2                   N/A           N/A NMC-4001
enp97s0f0       61:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  PCIE SLOT 1          1 (*)                N/A           N/A NMC-4001
enp97s0f1       61:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  PCIE SLOT 1          2 (*)                N/A           N/A NMC-4001
enp99s0f0       63:00.0     8086 1522 ffff 0000  00:0b:ab:f1:2f:6e  PCIE SLOT 2          1 (*)                unknown       N/A NMC-0108/0120
enp99s0f1       63:00.1     8086 1522 ffff 0000  00:0b:ab:f1:2f:6f  PCIE SLOT 2          2 (*)                unknown       N/A NMC-0108/0120
enp99s0f2       63:00.2     8086 1522 ffff 0000  00:0b:ab:f1:2f:70  PCIE SLOT 2          3 (*)                unknown       N/A NMC-0108/0120
enp99s0f3       63:00.3     8086 1522 ffff 0000  00:0b:ab:f1:2f:71  PCIE SLOT 2          4 (*)                unknown       N/A NMC-0108/0120
enp217s0f0      d9:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  PCIE SLOT 4          1 (*)                unknown       N/A NMC-0803/0806/0807
enp217s0f1      d9:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  PCIE SLOT 4          2 (*)                unknown       N/A NMC-0803/0806/0807
enp217s0f2      d9:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  PCIE SLOT 4          3 (*)                unknown       N/A NMC-0803/0806/0807
enp217s0f3      d9:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  PCIE SLOT 4          4 (*)                unknown       N/A NMC-0803/0806/0807
enp218s0f0      da:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  PCIE SLOT 4          5 (*)                unknown       N/A NMC-0803/0806/0807
enp218s0f1      da:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  PCIE SLOT 4          6 (*)                unknown       N/A NMC-0803/0806/0807
enp218s0f2      da:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  PCIE SLOT 4          7 (*)                unknown       N/A NMC-0803/0806/0807
enp218s0f3      da:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  PCIE SLOT 4          8 (*)                unknown       N/A NMC-0803/0806/0807
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

Figure 10: Usage of 'apdi -a -config apdi.cfg' Command

Please check the config file in Appendix 3.1. Note that the config file needs to be modified for each platform.

If there are some errors in the config file, then the apdi utility will tell you the line number and output the following error message.

```
root@localhost:~# ./apdi -a -legacy -config apdi.cfg
Can't find SLOT_NAME in SMBIOS type9/41 record! ./apdi.cfg line: 2
Error: APDI_ERR_CONF_FILE_FORMAT (0x81)
```

Figure 11: error message of 'apdi -a -config apdi.cfg' Command

Advantech Copyright 2024. All rights reserved.

1.2.11 ./apdi <command> -legacy -sys_NMC_cfg <file name>

Read the file to tell utility what NMC cards are in the platform when using -legacy command:

-legacy command could show NMC-0803/0807 if the user only uses NMC-0803, then the user can use NMC config file to tell apdi there is only NMC-0803, then apdi will only show NMC-0803 without NMC-0807. Please Check Appendix 3.2 to see the example.

This command should be added only after -legacy command.

```
root@localhost:~# ./apdi -a -config apdi.cfg -legacy -sys_NMC_cfg nmc_name.cfg

eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model

enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       LAN 1                A_BYPASS-C     0  Onboard

enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       LAN 2                A_BYPASS       0  Onboard

enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        SFP+1                N/A          N/A Onboard

enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        SFP+2                N/A          N/A Onboard

enp96s0f0       60:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  PCIE SLOT 1          p1                   N/A          N/A NMC-4001

enp96s0f1       60:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  PCIE SLOT 1          p2                   N/A          N/A NMC-4001

enp97s0f0       61:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  PCIE SLOT 1          1 (*)                N/A          N/A NMC-4001

enp97s0f1       61:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  PCIE SLOT 1          2 (*)                N/A          N/A NMC-4001

enp99s0f0       63:00.0     8086 1522 ffff 0000  00:0b:ab:f1:2f:6e  PCIE SLOT 2          1 (*)                A_BYPASS-C     8  NMC-0120

enp99s0f1       63:00.1     8086 1522 ffff 0000  00:0b:ab:f1:2f:6f  PCIE SLOT 2          2 (*)                A_BYPASS       8  NMC-0120

enp99s0f2       63:00.2     8086 1522 ffff 0000  00:0b:ab:f1:2f:70  PCIE SLOT 2          3 (*)                A_BYPASS-C     9  NMC-0120

enp99s0f3       63:00.3     8086 1522 ffff 0000  00:0b:ab:f1:2f:71  PCIE SLOT 2          4 (*)                A_BYPASS       9  NMC-0120

enp217s0f0      d9:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  PCIE SLOT 4          1 (*)                L_BYPASS      16  NMC-0803

enp217s0f1      d9:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  PCIE SLOT 4          2 (*)                L_BYPASS      16  NMC-0803

enp217s0f2      d9:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  PCIE SLOT 4          3 (*)                L_BYPASS      17  NMC-0803

enp217s0f3      d9:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  PCIE SLOT 4          4 (*)                L_BYPASS      17  NMC-0803

enp218s0f0      da:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  PCIE SLOT 4          5 (*)                L_BYPASS      18  NMC-0803

enp218s0f1      da:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  PCIE SLOT 4          6 (*)                L_BYPASS      18  NMC-0803

enp218s0f2      da:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  PCIE SLOT 4          7 (*)                L_BYPASS      19  NMC-0803

enp218s0f3      da:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  PCIE SLOT 4          8 (*)                L_BYPASS      19  NMC-0803

(*) means default, user can use config file to overwrite the panel name.

(!) means CFG file feature is not matched with Adv default feature.
```

Figure 12: Usage of 'apdi -a -legacy -sys_NMC_cfg' Command

Advantech Copyright 2024. All rights reserved.

1.2.12 ./apdi <command> -smbios_cfg <file name>

Read the file to get the correct smbios type 9/41 records of the platform:

-smbios_cfg command will use the records from file and ignore the records which were read from the real BIOS. This command is useful when the BIOS contains incorrect SMBIOS type 9/41 records.

Please check Appendix 3.3 to see the example.

```
root@localhost:~# ./apdi -a -config apdi.cfg -legacy -sys_NMC_cfg nmc_name.cfg -smbios_cfg smbios.cfg

eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature      seg model

enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg     N/A Onboard

enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       Onboard I210_1 (*)   Need cfg     N/A Onboard

enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard

enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        Onboard 10GBe (*)    Need cfg     N/A Onboard

enp96s0f0       60:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  cfg Slot 1           1 (*)                N/A          N/A NMC-4001

enp96s0f1       60:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  cfg Slot 1           2 (*)                N/A          N/A NMC-4001

enp97s0f0       61:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  cfg Slot 1           3 (*)                N/A          N/A NMC-4001

enp97s0f1       61:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  cfg Slot 1           4 (*)                N/A          N/A NMC-4001

enp99s0f0       63:00.0     8086 1522 ffff 0000  00:0b:ab:f1:2f:6e  cfg Slot 2           1 (*)                A_BYPASS-C    8  NMC-0120

enp99s0f1       63:00.1     8086 1522 ffff 0000  00:0b:ab:f1:2f:6f  cfg Slot 2           2 (*)                A_BYPASS      8  NMC-0120

enp99s0f2       63:00.2     8086 1522 ffff 0000  00:0b:ab:f1:2f:70  cfg Slot 2           3 (*)                A_BYPASS-C    9  NMC-0120

enp99s0f3       63:00.3     8086 1522 ffff 0000  00:0b:ab:f1:2f:71  cfg Slot 2           4 (*)                A_BYPASS      9  NMC-0120

enp217s0f0      d9:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  cfg Slot 4           1 (*)                L_BYPASS     16  NMC-0803

enp217s0f1      d9:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  cfg Slot 4           2 (*)                L_BYPASS     16  NMC-0803

enp217s0f2      d9:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  cfg Slot 4           3 (*)                L_BYPASS     17  NMC-0803

enp217s0f3      d9:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  cfg Slot 4           4 (*)                L_BYPASS     17  NMC-0803

enp218s0f0      da:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  cfg Slot 4           5 (*)                L_BYPASS     18  NMC-0803

enp218s0f1      da:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  cfg Slot 4           6 (*)                L_BYPASS     18  NMC-0803

enp218s0f2      da:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  cfg Slot 4           7 (*)                L_BYPASS     19  NMC-0803

enp218s0f3      da:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  cfg Slot 4           8 (*)                L_BYPASS     19  NMC-0803

(*) means default, user can use config file to overwrite the panel name.

(!) means CFG file feature is not matched with Adv default feature.
```

Figure 13: Usage of 'apdi -a -smbios_cfg smbios.cfg' Command

Advantech Copyright 2024. All rights reserved.

1.2.13 ./apdi <command> -sort_slotname

Sort the eth devices by slot name:

This command could be added after any other command.

```
root@localhost:~# ./apdi -a -sort_slotname

eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model

enp61s0f0       3d:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  Onboard 10GBe        Onboard 10GBe (*)    Need cfg      N/A Onboard

enp61s0f1       3d:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  Onboard 10GBe        Onboard 10GBe (*)    Need cfg      N/A Onboard

enp1s0          01:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2e  Onboard I210_0       Onboard I210_0 (*)   Need cfg      N/A Onboard

enp2s0          02:00.0     8086 1533 ffff 0000  00:0b:ab:f4:ef:2f  Onboard I210_1       Onboard I210_1 (*)   Need cfg      N/A Onboard

enp94s0f0       5e:00.0     8086 1521 ffff 0000  00:0b:ab:bd:ab:aa  Slot 1               1 (*)                unknown       N/A ID not found

enp94s0f1       5e:00.1     8086 1521 ffff 0000  00:0b:ab:bd:ab:ab  Slot 1               2 (*)                unknown       N/A ID not found

enp94s0f2       5e:00.2     8086 1521 ffff 0000  00:0b:ab:bd:ab:ac  Slot 1               3 (*)                unknown       N/A ID not found

enp94s0f3       5e:00.3     8086 1521 ffff 0000  00:0b:ab:bd:ab:ad  Slot 1               4 (*)                unknown       N/A ID not found

enp96s0f0       60:00.0     8086 1521 ffff 0000  00:0b:ab:bd:aa:2a  Slot 1               5 (*)                unknown       N/A ID not found

enp96s0f1       60:00.1     8086 1521 ffff 0000  00:0b:ab:bd:aa:2b  Slot 1               6 (*)                unknown       N/A ID not found

enp96s0f2       60:00.2     8086 1521 ffff 0000  00:0b:ab:bd:aa:2c  Slot 1               7 (*)                unknown       N/A ID not found

enp96s0f3       60:00.3     8086 1521 ffff 0000  00:0b:ab:bd:aa:2d  Slot 1               8 (*)                unknown       N/A ID not found

enp100s0f0      64:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:78  Slot 2               1 (*)                unknown       N/A ID not found

enp100s0f1      64:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:79  Slot 2               2 (*)                unknown       N/A ID not found

enp102s0f0      66:00.0     8086 10fb ffff ffff  00:0b:ab:f3:35:7a  Slot 2               3 (*)                unknown       N/A ID not found

enp102s0f1      66:00.1     8086 10fb ffff ffff  00:0b:ab:f3:35:7b  Slot 2               4 (*)                unknown       N/A ID not found

enp216s0f0      d8:00.0     8086 1583 13fe 002e  00:0b:ab:f6:8e:c2  Slot 3               1 (*)                N/A           N/A NMC-4006

enp216s0f1      d8:00.1     8086 1583 13fe 0000  00:0b:ab:f6:8e:c3  Slot 3               2 (*)                N/A           N/A NMC-4006

enp218s0f0      da:00.0     8086 1521 ffff 0000  00:0b:ab:3d:b6:41  Slot 4               1 (*)                unknown       N/A ID not found

enp218s0f1      da:00.1     8086 1521 ffff 0000  00:0b:ab:3d:b6:42  Slot 4               2 (*)                unknown       N/A ID not found

enp218s0f2      da:00.2     8086 1521 ffff 0000  00:0b:ab:3d:b6:43  Slot 4               3 (*)                unknown       N/A ID not found

enp218s0f3      da:00.3     8086 1521 ffff 0000  00:0b:ab:3d:b6:44  Slot 4               4 (*)                unknown       N/A ID not found

enp220s0f0      dc:00.0     8086 1521 ffff 0000  00:0b:ab:5c:1d:9a  Slot 4               5 (*)                unknown       N/A ID not found

enp220s0f1      dc:00.1     8086 1521 ffff 0000  00:0b:ab:5c:1d:9b  Slot 4               6 (*)                unknown       N/A ID not found

enp220s0f2      dc:00.2     8086 1521 ffff 0000  00:0b:ab:5c:1d:9c  Slot 4               7 (*)                unknown       N/A ID not found

enp220s0f3      dc:00.3     8086 1521 ffff 0000  00:0b:ab:5c:1d:9d  Slot 4               8 (*)                unknown       N/A ID not found

(*) means default, user can use config file to overwrite the panel name.

(!) means CFG file feature is not matched with Adv default feature.

(?) means CFG file changed the slot number of slot name or NO matched SMBIOS type9/41 record.

    APDI can't identify the slot number to output corresponding seg number.
```

Figure 14: Usage of 'apdi -a -sort_slotname' Command

Advantech Copyright 2024. All rights reserved.

1.2.14 ./apdi -support

List all supported devices.

```
root@localhost:~# ./apdi -support
model              feature        svid sdid index
NMC-0107           L_BYPASS       13fe 0038 0
NMC-0108           N/A            13fe 0037 1
NMC-0111           L_BYPASS       13fe 2042 2
NMC-0112           N/A            13fe 2043 3
NMC-0114           L_BYPASS       13fe 2044 4
NMC-0115           L_BYPASS       13fe 2045 5
NMC-0116           N/A            13fe 2046 6
NMC-0120           A_BYPASS       13fe 002a 7
NMC-0120           A_BYPASS       13fe 002b 8
NMC-0121           A_BYPASS       13fe 0035 9
NMC-0121           N/A            13fe 0036 10
NMC-0122           A_BYPASS       13fe 201f 11
NMC-0122           L_BYPASS       13fe 203f 12
NMC-0122           N/A            13fe 203a 13
NMC-0803           L_BYPASS       13fe 0034 14
NMC-0804           N/A            13fe 2041 15
NMC-0805           N/A            13fe 2040 16
NMC-0806           A_BYPASS       13fe 0032 17
NMC-0806           N/A            13fe 0033 18
NMC-0807           A_BYPASS       13fe 201a 19
NMC-0807           L_BYPASS       13fe 203d 20
NMC-0807           N/A            13fe 201d 21
NMC-0807           A_BYPASS       13fe 204b 22
NMC-0807           L_BYPASS       13fe 204c 23
NMC-0807           N/A            13fe 204d 24
NMC-0808           A_BYPASS       13fe 201b 25
NMC-0808           L_BYPASS       13fe 203e 26
NMC-0808           N/A            13fe 201e 27
NMC-0808           A_BYPASS       13fe 204e 28
NMC-0808           L_BYPASS       13fe 204f 29
NMC-0808           N/A            13fe 2050 30
NMC-0809           N/A            13fe 201c 31
NMC-0809           N/A            13fe 2051 32
NMC-1001           N/A            13fe 2017 33
NMC-1004           N/A            13fe 0031 34
NMC-1008           A_BYPASS       13fe 0028 35
NMC-1008           A_BYPASS       13fe 0029 36
NMC-1009           N/A            13fe 200c 37
```

Advantech Copyright 2024. All rights reserved.

NMC-1010	A_BYPASS	13fe 200d	38
NMC-1011	N/A	13fe 2019	39
NMC-1012FN	N/A	13fe 205a	40
NMC-2501	N/A	13fe 2018	41
NMC-4001	N/A	13fe 0030	42
NMC-4005	N/A	13fe 002f	43
NMC-4006	N/A	13fe 002e	44
NMC-4006	N/A	13fe 2058	45
NMC-4007	A_BYPASS	13fe 0027	46
NMC-4007	A_BYPASS	13fe 0026	47
NMC-4008	A_BYPASS	13fe 2013	48
NMC-4008	A_BYPASS	13fe 2056	49
NMC-4009	N/A	13fe 2014	50
NMC-6002	N/A	15b3 2015	51
NMC-6002	N/A	13fe 2015	52
NMC-6003	N/A	13fe 2048	53
NMC-6003L	N/A	13fe 2059	54
NMC-E01	POE	13fe 2016	55
NMC-E02	POE	13fe 2047	56
NMC-UW01	N/A	13fe 204a	57
NMC-01N0	A_BYPASS_I2C	13fe 205d	58
NMC-01N1	N/A	13fe 205e	59
NMC-10N0	N/A	13fe 205f	60
NMC-10N1	N/A	13fe 2067	61
NMC-10N2	A_BYPASS_I2C	13fe 2066	62
NMC-08N1	A_BYPASS_I2C	13fe 2061	63
NMC-08N1	N/A	13fe 2065	64
NMC-0808-SF00C	A_BYPASS_I2C	13fe 2069	65
NMC-0807-SF00C	A_BYPASS_I2C	13fe 206a	66
NMC-0122-SF00C	A_BYPASS_I2C	13fe 206b	67
NMC-0808-SF01E	N/A	13fe 206c	68
NMC-0807-SF01E	N/A	13fe 206d	69
NMC-0122-SF01E	N/A	13fe 206e	70
NMC-4006-SF02E	N/A	13fe 2072	71
NMC-10M0FN-04A1	N/A	13fe 2070	72
NMC-10M0FN-04A2	N/A	13fe 2071	73
NMC-RFKIT	N/A	1912 0015	74
PCIE-2130NP	N/A	13fe 205b	75
PCIE-2131BP	A_BYPASS	13fe 0011	76
PCIE-2221BP	A_BYPASS	13fe 3008	77
PCIE-2221NP	N/A	13fe 3009	78
PCIE-2221NP	N/A	13fe 300a	79
PCIE-2232BP	A_BYPASS	13fe 2052	80

Advantech Copyright 2024. All rights reserved.

```
PCIE-2420          N/A            13fe 3007 81
I210_CO            A_BYPASS       13fe 302b 82
I210_CO            A_BYPASS       13fe 302e 83
I210_CO            N/A            13fe 301c 84
I210_CO            N/A            13fe 301a 85
I210_CO            NCSI           13fe 3016 86
I210_CO            POE            13fe 3023 87
I211_CO            N/A            13fe 3022 88
I211_CO            POE            13fe 3027 89
I350_CO            A_BYPASS       13fe 301e 90
I350_CO            N/A            13fe 3017 91
X553_CO            A_BYPASS       13fe 301b 92
X553_CO            A_BYPASS       13fe 3026 93
X553_CO            N/A            13fe 3024 94
X553_CO            POE            13fe 3018 95
X553_FI            N/A            13fe 3025 96
X553_FI            N/A            13fe 302c 97
X710_FI            1PPS           13fe 3031 98
X710_FI            N/A            13fe 3029 99
X722_FI            1PPS           13fe 302f 100
X722_FI            N/A            13fe 301f 101
X722_FI            N/A            13fe 3028 102
X722_FI            NCSI           13fe 301d 103
X722_FI            NCSI/1PPS      13fe 3030 104
WX1860A2_CO        N/A            13fe 3032 105
WX1860A4_CO        A_BYPASS_I2C   13fe 3033 106
WX1860A4_FI        N/A            13fe 3034 107
I226_CO            N/A            13fe 3035 108
E822_FI_CO         A_BYPASS       13fe 3037 109
I210_FI            N/A            13fe 303c 110
I226_CO            A_BYPASS_I2C   13fe 3043 111
WX1860A4_CO        L_BYPASS       13fe 3045 112
X710_FI            N/A            13fe 303d 113
WX1860A4_FI        N/A            13fe 3038 114
WX1860A4_CO        A_BYPASS_I2C   13fe 3039 115
WX1860A4_CO        N/A            13fe 303a 116
E823_FI_CO         A_BYPASS       13fe 303b 117
AMD_FI             N/A            13fe 3041 118
I350_CO            POE            13fe 3042 119
```

Figure 15: Usage of 'apdi -support' Command

Advantech Copyright 2024. All rights reserved.

1.2.15 ./apdi -sysinfo

Display system info.

```
root@localhost:~# ./apdi -sysinfo
Onboard_cnt: 1
Slot_cnt: 4
Onboard I210_0: Onboard
cfg Slot 1:
cfg Slot 2: NMC-0805
cfg Slot 3: NMC-1004
cfg Slot 4:
```

Figure 16: Usage of 'apdi -sysinfo' Command

1.2.16 ./apdi <command> -novlan

Ignore virtual NIC port in system.

This command could be added after any other command.

```
root@localhost:~# ./apdi -a -novlan
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model
b03p0           03:00.0     8086 1533 ffff 0000  c4:00:ad:07:07:1d  MGMT 1               MGMT 1 (*)           Need cfg      N/A Onboard
b04p0           04:00.0     8086 1533 ffff 0000  c4:00:ad:07:07:1e  MGMT 2               MGMT 2 (*)           Need cfg      N/A Onboard
b17p0           17:00.0     8086 1521 13fe 2016  74:fe:48:2f:95:5a  Slot 4               1 (*)                POE           N/A NMC-E01
b17p1           17:00.1     8086 1521 13fe 2016  74:fe:48:2f:95:5b  Slot 4               2 (*)                POE           N/A NMC-E01
b17p2           17:00.2     8086 1521 13fe 2016  74:fe:48:2f:95:5e  Slot 4               3 (*)                POE           N/A NMC-E01
b17p3           17:00.3     8086 1521 13fe 2016  74:fe:48:09:3d:d9  Slot 4               4 (*)                POE           N/A NMC-E01
b19p0           19:00.0     8086 1521 13fe 0032  00:0b:ab:dd:6c:dc  Slot 3               1 (*)                A_BYPASS-C    12  NMC-0806
b19p1           19:00.1     8086 1521 13fe 0032  00:0b:ab:dd:6c:dd  Slot 3               2 (*)                A_BYPASS      12  NMC-0806
b19p2           19:00.2     8086 1521 13fe 0032  00:0b:ab:dd:6c:de  Slot 3               3 (*)                A_BYPASS-C    13  NMC-0806
b19p3           19:00.3     8086 1521 13fe 0032  00:0b:ab:dd:6c:df  Slot 3               4 (*)                A_BYPASS      13  NMC-0806
b6bp0           6b:00.0     8086 37d3 8086 0000  00:00:00:00:03:14  SFP+ 1               SFP+ 1 (*)           Need cfg      N/A Onboard
b6bp1           6b:00.1     8086 37d3 8086 0000  00:00:00:00:03:15  SFP+ 2               SFP+ 2 (*)           Need cfg      N/A Onboard
bb9p0           b9:00.0     8086 10fb 13fe 0028  00:0b:ab:f1:e5:a8  Slot 1               1 (*)                A_BYPASS-C    4   NMC-1008
bb9p1           b9:00.1     8086 10fb 13fe 0028  00:0b:ab:f1:e5:a9  Slot 1               2 (*)                A_BYPASS      4   NMC-1008
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
(?) means CFG file changed the slot number of slot name or NO matched SMBIOS type9/41 record.
    APDI can't identify the slot number to output corresponding seg number.
(u) means this NIC device is from USB. The vid and did are from USB.
(!!) means there is no valid type 9/41 SMBIOS record. Please contact Advantech.
N/A in slot name means it's a USB network device or a virtual network device. It's NOT a PCIE network device.
```

Figure 17: Usage of 'apdi -a -novlan' Command

Advantech Copyright 2024. All rights reserved.

2. LIBAPDI –SERVER IMANAGER PLATFORM DEVICE IDENTIFIER LIBRARY

Advantech provides a static Linux library (libapdi.a) and corresponding header file (libapdi.h) for integration of Server iManager Platform Device Identifier into customer applications.

2.1 Supported Distributions

Libapdi is a generic Linux user space library and thus is expected to work with most Linux distributions.

Advantech uses the latest up-to-date Ubuntu distributions (4.15.0-22-generic) to perform software quality assurance and release testing. For more details, please refer to the release notes delivered as part of the utility.

You can also refer to Server_iManager support list.

https://www.advantech.com/en/support/details/software-utility?id=1-1YC9KKF

2.2 Header File

The header file exports/ provides common definitions for the API commands. A brief introduction is provided here for reference only. Please check the header file for more details:

P.S. The definition of error codes and struct can be found in Section 2.4

```
/**
 *
 * Copyright (c) 2017 Advantech Co. Ltd.
 * All Rights Reserved
 *
 */



#ifndef _LIBAPDI_H
#define _LIBAPDI_H

#include "apdi_cmd_reg_var.h"

#define APDI_LIB_MAJOR_VERSION 1
#define APDI_LIB_MINOR_VERSION 34

/**
 * @brief Get apdi library version.
 */
extern apdi_err apdi_get_lib_version(unsigned char *major, unsigned char *minor);

/**
```

```
 * @brief Set message output level
 */
extern apdi_err apdi_set_message_level(int msg_level);

/**
 * @brief Provide system mapping
 *
 * -# Check each eth devices, get its PCIe bus:dev.fun, get subsystem
 * Vendor/Device ID, mac address, slot name, panel name, feature, seg number, model name.
 * And save these data in
 * struct apdi_system_environment_info_hdr apdi_system_environment_info;
 * struct apdi_eth_data_hdr apdi_eth_device_mapping[MAXIMUM_PORT_NUMBER];
 */
extern apdi_err apdi_open(void);

/**
 * @brief print all apdi interface
 if eth_name is not NULL
 then this function will print the eth_name device only.
 if the function_name is not NULL
 then this function will print the devices which have this function.
 if the bus_dev_fun is not NULL
 then this function will print the device which has this bus_dev_fun.
 */
extern apdi_err apdi_print_scan_result(char* eth_name, char* feature, char* bus_dev_fun);

/**
 * @brief fprint all apdi interface
 */
extern apdi_err apdi_fprint_scan_result(char* file_name);

/**
 * @brief This function prints the result of apdi_open(). This function will printf devices'
name, bus, slot name, panel name.
 * If storage_name is not NULL, then this function will print the storage_name device only.
(flag don't care)
 * flag =
APDI_STORAGE_PRINT_DO_NOT_CARE/APDI_STORAGE_PRINT_ROOT_DEV/APDI_STORAGE_PRINT_BLOCK_DEV/APDI_S
TORAGE_PRINT_CONFIG
 */
extern apdi_err apdi_print_storage_scan_result(char* storage_name, char flag);

/**
 * @brief This function dump EFI Variable to specific file.
```

```
 */
extern apdi_err apdi_dump_efi_var_to_file(char *efi_var, char* file_name);


/**
 * @brief This function prints string for the storage_mapping_cfg file based on storage_name
and slot_name.
 */
extern apdi_err apdi_print_storage_mapping_string(char *storage_name, char *slot_name);


/**
 * @brief output lbpcu.cfg file
 */
extern apdi_err apdi_output_lbpcu_cfg(void);


/**
 * @brief printf all supported NMC names
 */
extern apdi_err apdi_print_supported_NMC(void);


/**
 * @brief printf all system info.
 */
extern apdi_err apdi_print_system_info(void);


/**
 * @brief Close all apdi
 */
extern apdi_err apdi_close(void);


#endif
```

Advantech Copyright 2024. All rights reserved.

2.3 API Command List

The following table lists all available API and highlights required versions for their support:

API	Section	Supported from libapdi version
apdi_set_message_level	2.6.1.1	01.00
apdi_open	2.6.2.1	01.00
apdi_close	0	01.00
apdi_get_lib_version	2.6.3.1	01.00
apdi_print_scan_result	2.6.4.1	01.00
apdi_fprint_scan_result	2.6.4.2	01.00
apdi_output_lbpcu_cfg	2.6.4.3	01.00
apdi_print_supported_NMC	2.6.4.4	01.02
apdi_print_system_info	2.6.4.5	01.08
apdi_print_storage_scan_result	2.6.4.6	02.00
apdi_dump_efi_var_to_file	2.6.4.7	02.00
apdi_print_storage_mapping_string	2.6.4.8	02.00

2.4 Error Codes

```
typedef signed char apdi_err;
/* APP Return code */
#define APDI_NO_ERROR                      (0)
#define APDI_ERR_OPEN_FILE                 (-128)
#define APDI_ERR_CONF_FILE_FORMAT          (-127)
#define APDI_ERR_CLOSE_FILE                (-125)
#define APDI_ERR_CLOSE                     (-120)
#define APDI_ERR_DATA_FORMAT               (-116)
#define APDI_ERR_SMBIOS_NOT_EXIST          (-111)
#define APDI_ERR_SMBIOS_CHECKSUM           (-110)
#define APDI_ERR_PARAM_OUT_OF_BOUND        (-109)
#define APDI_ERR_PARAM_NULL_POINTER        (-108)
#define APDI_ERR_INIT                      (-107)
#define APDI_ERR_COMMAND_NOT_SUPPORT       (-106)
#define APDI_ERR_MMAP                      (-105)
#define APDI_ERR_OPEN_DIR                  (-104)
#define APDI_ERR_NIC_NOT_EXIST             (-103)
#define APDI_ERR_SMBIOS_DATA_FORMAT_WRONG  (-102)
#define FAIL                               (-1)
#define TRUE                               1
#define FALSE                              0
#define SUCCESS                            0
```

Advantech Copyright 2024. All rights reserved.

2.5 How to Use the API

In order to use the apdi library, the following steps need to be followed:

- Call the apdi_open function to allocate and initialize resources.
- Perform functions or operations.
- Call the apdi_close function to properly release allocated resources.

2.6 API Functions

2.6.1 Message Output Level

2.6.1.1 apdi_err apdi_set_message_level(int msg_level)

This function sets the message output level.

Msg_level 0 means no error and debug message.

Msg_level 1 means no debug but error message.

Msg_level 2 means open debug and error message.

Returns 0 on success or the following error codes on error:

Name	Error Code
APDI_ERR_PARAM_OUT_OF_BOUND	0x93

Table 1: API Function 'apdi_set_message_level'

2.6.2 Open/Close

2.6.2.1 apdi_err apdi_open(void)

This function opens a software interface and provides system mapping. All information of eth devices (not including virtual devices) and storage devices will be recorded in global variables in libapdi.h.

```
struct apdi_system_environment_info_hdr apdi_system_environment_info;
struct apdi_eth_data_hdr apdi_eth_device_mapping[MAXIMUM_PORT_NUMBER];
struct apdi_storage_data_hdr apdi_storage_device_mapping[MAXIMUM_PORT_NUMBER];
```

The user need to call this function first before any other operations. This function will also check variables in apdi_cmd_reg_var.h.

If the user sets the *g_config_flag*, then this function will check and use the *g_config_file_name* file.

If the user sets the *g_legacy_flag*, then this function will try to identify the eth device even if this is a legacy card from Advantech.

If the user sets the *g_force_flag*, then this function will overwrite the eth device's feature according to the config file, even if the feature of the config is conflicting with the result of this function.

Advantech Copyright 2024. All rights reserved.

If the user sets the *g_NMC_name_flag*, then this function will check and use the *g_NMC_name_file_name* file.

If the user sets the *g_smbios_flag*, then this function will check and use the *g_smbios_file_name* file.

If the user sets the *g_novlan*, then all virtual NIC ports will be ignored.

If the user set the *g_apdi_storage_mapping_cfg,* then this function will check and use the *g_apdi_storage_mapping_cfg_file_name* file.

```
int g_legacy_flag = 0;
int g_config_flag = 0;
int g_force_flag = 0;
int g_NMC_name_flag = 0;
int g_smbios_flag = 0;
int g_apdi_storage_mapping_cfg = 0;
……
int g_novlan = 0;

char g_config_file_name[NAME_LEN];
char g_NMC_name_file_name[NAME_LEN];
char g_smbios_file_name[NAME_LEN];
char g_apdi_storage_mapping_cfg_file_name[NAME_LEN];
```

Returns 0 on success or the following error codes on error:

Name	Error Code
APDI_ERR_OPEN_DIR	-104
APDI_ERR_MMAP	-105
APDI_ERR_SMBIOS_CHECKSUM	-110
APDI_ERR_CONF_FILE_FORMAT	-127
APDI_ERR_PARAM_NULL_POINTER	-108

Table 2: API Function 'apdi_open'

Advantech Copyright 2024. All rights reserved.

2.6.2.2 apdi_err apdi_close (void)

This function closes all software interface.

Returns 0 on success or the following error codes on error:

Name	Error Code
APDI_ERR_INIT	-107

Table 3: API Function 'apdi_close'

2.6.3 Information

2.6.3.1 apdi_err apdi_get_lib_version(unsigned char *major, unsigned char *minor)

This function returns the major and minor versions of libapdi.

Please refer to the release notes for more information on libapdi versions.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_PARAM_NULL_POINTER	-108

Table 4: API Function 'apdi_get_lib_version'

2.6.4 Output results

2.6.4.1 apdi_print_scan_result(char* eth_name, char* feature, char* bus_dev_fun)

This function prints the result of apdi_open(). This function will printf eth devices' name, bus, id, subsystem id, slot name, panel name, feature, segment number, and model name of Advantech.

If eth_name is not NULL, then this function will print the eth_name device only. If the feature is not NULL, then this function will print the devices which have this feature. If the bus_dev_fun is not NULL, then this function will print the device which has this bus_dev_fun.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107

Table 5: API Function 'apdi_print_scan_result'

Advantech Copyright 2024. All rights reserved.

2.6.4.2 apdi_fprint_scan_result (char* file_name)

This function prints the result of apdi_open() to a file. This function will printf eth devices' name, bus, id, subsystem id, slot name, panel name, feature, segment number, and model name of Advantech into the file_name file.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107

Table 6: API Function 'apdi_fprint_scan_result'

2.6.4.3 apdi_output_lbpcu_cfg(void)

This function will output the lbpcu config file which is based on the result of apdi_open(). The config file will contain all eth devices which have the advanced LAN bypass feature.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107

Table 7: API Function 'apdi_output_lbpcu_cfg'

2.6.4.4 apdi_print_supported_NMC(void)

This function will print all supported eth devices.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107

Table 8: API Function 'apdi_print_supported_NMC'

2.6.4.5 apdi_print_system_info (void)

This function will output number of onboard NICs and number of slots on the platform.

Also outputs the NMC name of each slot and returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107

Table 9: API Function 'apdi_print_system_info'

Advantech Copyright 2024. All rights reserved.

2.6.4.6 apdi_print_storage_scan_result(char* storage_name, char flag)

This function prints the result of apdi_open(). This function will printf devices' name, bus, slot name, panel name.

If storage_name is not NULL, then this function will print the storage_name device only. (flag don't care)

Flag: APDI_STORAGE_PRINT_DO_NOT_CARE, APDI_STORAGE_PRINT_ROOT_DEV, APDI_STORAGE_PRINT_BLOCK_DEV, APDI_STORAGE_PRINT_CONFIG

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107
APDI_ERR_PARAM_OUT_OF_BOUND	-109

Table 10: API Function 'apdi_print_storage_scan_result'

2.6.4.7 apdi_dump_efi_var_to_file(char *efi_var, char* file_name)

This function dump EFI Variable to specific file.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107
APDI_ERR_OPEN_FILE	-128

Table 11: API Function 'apdi_dump_efi_var_to_file'

2.6.4.8 apdi_print_storage_mapping_string(char *storage_name, char *slot_name)

This function prints string for the storage_mapping_cfg file based on storage_name and slot_name.

Returns 0 on success or the following error code on error:

Name	Error Code
APDI_ERR_INIT	-107
APDI_ERR_PARAM_OUT_OF_BOUND	-109

Table 12: API Function 'apdi_print_storage_mapping_string'

Advantech Copyright 2024. All rights reserved.

3. APPENDIX

3.1 Config File example for "–config" parameter

Here is an example in which Slot 5 is connected to a PEX 8724 PCIe switch.

Slot 5 => Slot 5, 6, 7, 8.

```
SLOT_NAME="Slot 5"
PCIe_SW_DEV/NMC_SLOT="01-01/Slot 5"
PCIe_SW_DEV/NMC_SLOT="02-02/Slot 6"
PCIe_SW_DEV/NMC_SLOT="09-09/Slot 7"
PCIe_SW_DEV/NMC_SLOT="0a-0a/Slot 8"
```

```
[root@localhost source]# lspci -t
-[0000:00]-+-00.0
           +-01.0-[01]--+-00.0
           |            +-00.1
           |            +-00.2
           |            \-00.3
           +-01.1-[02]--+-00.0
           |            +-00.1
           |            +-00.2
           |            \-00.3
           +-02.0
           +-14.0
           +-14.2
           +-16.0
           +-17.0
           +-1c.0-[03]--+-00.0
           |            \-00.1
           +-1c.4-[04]--+-00.0
           |            +-00.1
           |            +-00.2
           |            \-00.3
           +-1d.0-[05-0b]----00.0-[06-0b]--+-01.0-[07]--+-00.0
           |                               |            +-00.1
           |                               |            +-00.2
           |                               |            \-00.3
           |                               +-02.0-[08]--
           |                               +-08.0-[09]--
           |                               +-09.0-[0a]--+-00.0
           |                               |            +-00.1
           |                               |            +-00.2
           |                               |            \-00.3
           |                               \-0a.0-[0b]--+-00.0
           |                                            \-00.1
```

```
[root@localhost source]# lspci
00:00.0 Host bridge: Intel Corporation Skylake Host Bridge/
00:01.0 PCI bridge: Intel Corporation Skylake PCIe Controll
00:01.1 PCI bridge: Intel Corporation Skylake PCIe Controll
00:02.0 VGA compatible controller: Intel Corporation HD Gra
00:14.0 USB controller: Intel Corporation Sunrise Point-H U
00:14.2 Signal processing controller: Intel Corporation Sun
00:16.0 Communication controller: Intel Corporation Sunrise
00:17.0 SATA controller: Intel Corporation Sunrise Point-H
00:1c.0 PCI bridge: Intel Corporation Sunrise Point-H PCI E
00:1c.4 PCI bridge: Intel Corporation Sunrise Point-H PCI E
00:1d.0 PCI bridge: Intel Corporation Sunrise Point-H PCI E
00:1d.6 PCI bridge: Intel Corporation Sunrise Point-H PCI E
00:1d.7 PCI bridge: Intel Corporation Sunrise Point-H PCI E
00:1f.0 ISA bridge: Intel Corporation Sunrise Point-H LPC C
00:1f.2 Memory controller: Intel Corporation Sunrise Point-
00:1f.3 Audio device: Intel Corporation Sunrise Point-H HD
00:1f.4 SMBus: Intel Corporation Sunrise Point-H SMBus (rev
01:00.0 Ethernet controller: Intel Corporation I350 Gigabit
01:00.1 Ethernet controller: Intel Corporation I350 Gigabit
01:00.2 Ethernet controller: Intel Corporation I350 Gigabit
01:00.3 Ethernet controller: Intel Corporation I350 Gigabit
02:00.0 Ethernet controller: Intel Corporation I350 Gigabit
02:00.1 Ethernet controller: Intel Corporation I350 Gigabit
02:00.2 Ethernet controller: Intel Corporation I350 Gigabit
02:00.3 Ethernet controller: Intel Corporation I350 Gigabit
03:00.0 Ethernet controller: Intel Corporation 82599ES 10-G
03:00.1 Ethernet controller: Intel Corporation 82599ES 10-G
04:00.0 Ethernet controller: Intel Corporation I350 Gigabit
04:00.1 Ethernet controller: Intel Corporation I350 Gigabit
04:00.2 Ethernet controller: Intel Corporation I350 Gigabit
04:00.3 Ethernet controller: Intel Corporation I350 Gigabit
05:00.0 PCI bridge: PLX Technology, Inc. PEX 8724 24-Lane,
06:01.0 PCI bridge: PLX Technology, Inc. PEX 8724 24-Lane,
06:02.0 PCI bridge: PLX Technology, Inc. PEX 8724 24-Lane,
06:08.0 PCI bridge: PLX Technology, Inc. PEX 8724 24-Lane,
06:09.0 PCI bridge: PLX Technology, Inc. PEX 8724 24-Lane,
06:0a.0 PCI bridge: PLX Technology, Inc. PEX 8724 24-Lane,
```

```
#apdi_FWA-4130_8_slot.cfg
SLOT_NAME="MgmtLAN1"
FUNCTION/NAME/FEATURE="0/Mgmt LAN 1/NA"

SLOT_NAME="MgmtLAN2"
FUNCTION/NAME/FEATURE="0/Mgmt LAN 2/NA"

SLOT_NAME="Slot 5"
PCIe_SW_DEV/NMC_SLOT="01-01/Slot 5"
PCIe_SW_DEV/NMC_SLOT="02-02/Slot 6"
PCIe_SW_DEV/NMC_SLOT="09-09/Slot 7"
PCIe_SW_DEV/NMC_SLOT="0a-0a/Slot 8"

# ===== DO NOT CHANGE THE FORMAT IN THIS FILE =====
# This configuration file is for Advantech Platform Device Identifier.
# The purpose of this file is to porvide more information about the platform
# specific information like internal PCIe slot, PCIe switch or onboard NIC's
# feature based on HW design.
#
# The Bypass seg numbers in APDI are assigned according to the Slot number.
```

Advantech Copyright 2024. All rights reserved.

```
# If the user change the Slot names in apdi.cfg. Please make sure the last 2 characters are
the original numbers.
#
# SLOT_NAME: Identify which slot you want to define.
# CHANGE_NAME: Change the slot name as you want.
#
# For the middle board or riser board with PCIe switch, please identify the
# slots behind the PCIe switch. (Based on "device" number)
# PCIe_SW_DEV/NMC_SLOT: Which PCIe switch device is for which NMC slot.
# ex: PCIe switch device 01 ~ 03 is for NMC slot 2, then:
# PCIe_SW_DEV/NMC_SLOT="01-03/Slot 2"
# PCIe_SW_DEV/NMC_SLOT: Which PCIe switch device is for which PCIe slot.
# ex: PCIe switch device 02 ~ 04 is for riser's PCIe slot 1
# PCIe_SW_DEV/NMC_SLOT="02-04/Slot 1"
#
# For the onboard PCIe devices, please identify the name and features for each
# ports. (Based on "function" number)
# Features that need to be highlight:
# A_BYPASS, L_BYPSS, POE, NA
# A_BYPASS means advanced lan BYPASS
# A_BYPASS-C means advanced lan BYPASS which controls a segment by this BUS:DEV.FUNC
# L_BYPASS means legacy lan BYPASS
# POE means Power Over Ethernet
# NA means no features
# FUNCTION/NAME/FEATURE: Function for this port/The name of this port/What
#                        feature it can support.
#
# NOTE: 80:01.3 Means Bus = 80, Device = 01, Function = 3 (Hex)
#
# Examples:
# A. One platform has:
#   - 10 PCIe slots,
#     8 of them is for NMC slot 1 ~ 8, slot 9 and slot 10 is for Internal PCIe
#     cards.
#   - 2 Management ports.
#   - 2 SFP+ ports from PCH. But it use same PCIe root ports.
# In this example, we don't need a config file for it because 10 PCIe slots, 2
# management ports and 2 SFP+ ports will be identified already. We don't need
# to provide more information. If you want, you can highlight the SFP+ port
# name.
# config example:
# SLOT_NAME="SFP+"
# FUNCTION/NAME/FEATURE="0/SFP+ 1/NA"
```

```
# FUNCTION/NAME/FEATURE="1/SFP+ 2/NA"
#
# B. One platform has:
#   - 6 PCIe slots,
#     2 of them are x8 for NMC 1 and NMC 2 (Slot 1, Slot 2)
#     2 of them are x4 for NMC 3 and NMC 4 (Slot 3, Slot 4)
#     2 of them are x4 PCIe slots
#       Sku A with 4 NMCs, no Middle board with PCIe switch.
#       Sku B.1 with 6 NMCs, 1 middle board without PCIe switch.
#           B.2 with 6 NMCs, 1 middle board with PCIe switch (Slot 6 connected
#           to PCIe switch.)
#       Sku C with 8 NMCs, 1 middle board with 1 PCIe switch, it use 2 x4 NMC
#       slots.
#   - 2 management ports.
# In this example, there are 2 x4 Slots that may or may not have middle boards.
# So each sku (excpet Sku A and Sku B.1) will need one specific config files.
#
# Sku B.2 config example:
# (NMC slot 5 is on PCIe switch device 01 and 02, and NMC slot 6 is on PCIe
# switch device 09 and 0a.)
# SLOT_NAME="Slot 5"
# PCIe_SW_DEV/NMC_SLOT="01-02/Slot 5"
# PCIe_SW_DEV/NMC_SLOT="09-0a/Slot 6"
#
# Sku C config example:
# (NMC slot 5 is on PCIe switch device 01 and 02, NMC slot 6 is on PCIe switch
# device 09 and 0a, NMC slot 7 is on PCIe switch device 07 and 08, NMC slot 8 is
# on PCIe switch device 10 and 11.)
# SLOT_NAME="Slot 5"
# PCIe_SW_DEV/NMC_SLOT="01-02/Slot 5"
# PCIe_SW_DEV/NMC_SLOT="09-0a/Slot 6"
# PCIe_SW_DEV/NMC_SLOT="07-08/Slot 7"
# PCIe_SW_DEV/NMC_SLOT="10-11/Slot 8"
#
# C. One platform has:
#   - 2 PCIe slots
#     1 x16 slots for 2 x8 NMC slots, 1 x8 for internal PCIe cards.
#   - 2 Management ports.
#   - 1 NIC device with 4 ports and with advanced bypass function.
# In this example, 1 x16 slot and 1 x8 slot will be identified already. But
# we need to define functions for onboard NIC device.
# config example:
# SLOT_NAME="Onboard NIC"
```

Advantech Copyright 2024. All rights reserved.

```
# FUNCTION/NAME/FEATURE="0/LAN 1/A_BYPASS-C"
# FUNCTION/NAME/FEATURE="1/LAN 2/A_BYPASS"
# FUNCTION/NAME/FEATURE="2/LAN 3/A_BYPASS-C"
# FUNCTION/NAME/FEATURE="3/LAN 4/A_BYPASS"
#
# D. One platform has:
#   - 2 PCIe slots
#     1 x8, 1 x4 for internal PCIe cards.
#   -2 Management ports.
#  - 2 NIC devices, both with 4 ports, one NIC device is 4 RJ45 ports, another
#    is 2 RJ45 ports and 2 SFP ports.
# config example:
# SLOT_NAME="Onboard NIC 1"
# FUNCTION/NAME/FEATURE="0/LAN 1/NA"
# FUNCTION/NAME/FEATURE="1/LAN 2/NA"
# FUNCTION/NAME/FEATURE="2/LAN 3/NA"
# FUNCTION/NAME/FEATURE="3/LAN 4/NA"
#
# SLOT_NAME="Onboard NIC 2"
# FUNCTION/NAME/FEATURE="0/LAN 5/NA"
# FUNCTION/NAME/FEATURE="1/LAN 6/NA"
# FUNCTION/NAME/FEATURE="2/SFP 1/NA"
# FUNCTION/NAME/FEATURE="3/SFP 2/NA"
#
# E. One platform has:
#  - 2 NMC cards, one NMC device is 8 RJ45 ports, another
#    is 4 SFP ports.
#  - This part is the special case for the 8 port NMC card
#    which need to reassign the bus order of top board and bottom board.
# config example:
#SLOT_NAME="PCIE SLOT  1"
#FUNCTION/NAME/BUS_ORDER="0/p1/L"
#FUNCTION/NAME/BUS_ORDER="1/p2/L"
#FUNCTION/NAME/BUS_ORDER="2/p3/L"
#FUNCTION/NAME/BUS_ORDER="3/p4/L"
#FUNCTION/NAME/BUS_ORDER="0/p5/H"
#FUNCTION/NAME/BUS_ORDER="1/p6/H"
#FUNCTION/NAME/BUS_ORDER="2/p7/H"
#FUNCTION/NAME/BUS_ORDER="3/p8/H"
#
#SLOT_NAME="PCIE SLOT 3"
#FUNCTION/NAME/BUS_ORDER="0/p1/L"
#FUNCTION/NAME/BUS_ORDER="1/p2/L"
```

```
#FUNCTION/NAME/BUS_ORDER="2/p3/L"
#FUNCTION/NAME/BUS_ORDER="3/p4/L"
```

3.2 sys_NMC_cfg Config File example

nmc_name.cfg:

```
NMC-1004 NA
NAEM-0805 NA

#This NMC name cfg file is to solve the ambiguous when using -legacy command.
#Please use -legacy -sys_NMC_cfg <file name>.
#You can use apdi -support to see all supported NMC card list.
#Please add NMC name and feature which are on your platform.
#Use this format(NMC-XXXX featrue)
#NMC-0807 A_BYPASS
#NMC-0121 NA
#NMC-1004 NA
#NAEM-1001 NA
#NMC-0120 A_BYPASS
```

Here is an example for when using an old card on a platform.

1. Only -a command
   Apdi can't identify the model from slot 2 to slot 3. (ID not found)

```
root@localhost:~# ./apdi -a
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model
virbr0                                             52:54:00:05:0b:13                                           unknown       N/A ID not found
virbr0-nic                                         52:54:00:05:0b:13                                           unknown       N/A ID not found
eth159          06:00.0     8086 1521 ffff 0000  00:0b:ab:55:f8:8a  cfg Slot 2           1 (*)                unknown       N/A ID not found
eth162          06:00.1     8086 1521 ffff 0000  00:0b:ab:55:f8:8b  cfg Slot 2           2 (*)                unknown       N/A ID not found
eth160          06:00.2     8086 1521 ffff 0000  00:0b:ab:55:f8:8c  cfg Slot 2           3 (*)                unknown       N/A ID not found
eth161          06:00.3     8086 1521 ffff 0000  00:0b:ab:55:f8:8d  cfg Slot 2           4 (*)                unknown       N/A ID not found
eth157          07:00.0     8086 1522 ffff 0000  00:0b:ab:55:f8:8e  cfg Slot 2           5 (*)                unknown       N/A ID not found
eth163          07:00.1     8086 1522 ffff 0000  00:0b:ab:55:f8:8f  cfg Slot 2           6 (*)                unknown       N/A ID not found
eth156          07:00.2     8086 1522 ffff 0000  00:0b:ab:55:f8:90  cfg Slot 2           7 (*)                unknown       N/A ID not found
eth158          07:00.3     8086 1522 ffff 0000  00:0b:ab:55:f8:91  cfg Slot 2           8 (*)                unknown       N/A ID not found
eth124          0a:00.0     8086 10fb ffff ffff  00:0b:ab:f4:7c:c2  cfg Slot 3           1 (*)                unknown       N/A ID not found
eth120          0a:00.1     8086 10fb ffff ffff  00:0b:ab:f4:7c:c3  cfg Slot 3           2 (*)                unknown       N/A ID not found
eth135          0f:00.0     8086 1533 ffff 0000  00:0b:ab:b6:b7:b9  Onboard I210_0       Onboard I210_0 (*)   Need cfg      N/A Onboard
(*) means default, user can use config file to overwrite the panel name.
(!) means CFG file feature is not matched with Adv default feature.
```

2. -a -legacy command
   Apdi can identify the model from slot 2 to slot 3. (NMC-0805/0808)

```
root@localhost:~# ./apdi -a -legacy
eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name            panel name           feature       seg model
virbr0                                             52:54:00:05:0b:13                                           unknown       N/A ID not found
virbr0-nic                                         52:54:00:05:0b:13                                           unknown       N/A ID not found
eth159          06:00.0     8086 1521 ffff 0000  00:0b:ab:55:f8:8a  cfg Slot 2           1 (*)                unknown       N/A NMC-0805/0808
eth162          06:00.1     8086 1521 ffff 0000  00:0b:ab:55:f8:8b  cfg Slot 2           2 (*)                unknown       N/A NMC-0805/0808
eth160          06:00.2     8086 1521 ffff 0000  00:0b:ab:55:f8:8c  cfg Slot 2           3 (*)                unknown       N/A NMC-0805/0808
```

```
eth161          06:00.3     8086 1521 ffff 0000  00:0b:ab:55:f8:8d  cfg Slot 2          4 (*)                unknown       N/A NMC-0805/0808

eth157          07:00.0     8086 1522 ffff 0000  00:0b:ab:55:f8:8e  cfg Slot 2          5 (*)                unknown       N/A NMC-0805/0808

eth163          07:00.1     8086 1522 ffff 0000  00:0b:ab:55:f8:8f  cfg Slot 2          6 (*)                unknown       N/A NMC-0805/0808

eth156          07:00.2     8086 1522 ffff 0000  00:0b:ab:55:f8:90  cfg Slot 2          7 (*)                unknown       N/A NMC-0805/0808

eth158          07:00.3     8086 1522 ffff 0000  00:0b:ab:55:f8:91  cfg Slot 2          8 (*)                unknown       N/A NMC-0805/0808

eth124          0a:00.0     8086 10fb ffff ffff  00:0b:ab:f4:7c:c2  cfg Slot 3          1 (*)                unknown       N/A NAEM-1001/NMC-
1004/1008

eth120          0a:00.1     8086 10fb ffff ffff  00:0b:ab:f4:7c:c3  cfg Slot 3          2 (*)                unknown       N/A NAEM-1001/NMC-
1004/1008

eth135          0f:00.0     8086 1533 ffff 0000  00:0b:ab:b6:b7:b9  Onboard I210_0      Onboard I210_0 (*)   Need cfg      N/A Onboard

(*) means default, user can use config file to overwrite the panel name.

(!) means CFG file feature is not matched with Adv default feature.
```

3. -a -legacy -sys_NMC_cfg nmc_name.cfg
   Apdi uses nmc_name.cfg which is from the user and can identify the NMC cards successfully.

```
root@localhost:~# ./apdi -a -legacy -sys_NMC_cfg nmc_name.cfg

eth name        bus:dev.fun vid  did  svid sdid  mac address        slot name           panel name           feature       seg model

virbr0                                          52:54:00:05:0b:13                                               unknown       N/A ID not found

virbr0-nic                                      52:54:00:05:0b:13                                               unknown       N/A ID not found

eth159          06:00.0     8086 1521 ffff 0000  00:0b:ab:55:f8:8a  cfg Slot 2          1 (*)                N/A           N/A NMC-0805

eth162          06:00.1     8086 1521 ffff 0000  00:0b:ab:55:f8:8b  cfg Slot 2          2 (*)                N/A           N/A NMC-0805

eth160          06:00.2     8086 1521 ffff 0000  00:0b:ab:55:f8:8c  cfg Slot 2          3 (*)                N/A           N/A NMC-0805

eth161          06:00.3     8086 1521 ffff 0000  00:0b:ab:55:f8:8d  cfg Slot 2          4 (*)                N/A           N/A NMC-0805

eth157          07:00.0     8086 1522 ffff 0000  00:0b:ab:55:f8:8e  cfg Slot 2          5 (*)                N/A           N/A NMC-0805

eth163          07:00.1     8086 1522 ffff 0000  00:0b:ab:55:f8:8f  cfg Slot 2          6 (*)                N/A           N/A NMC-0805

eth156          07:00.2     8086 1522 ffff 0000  00:0b:ab:55:f8:90  cfg Slot 2          7 (*)                N/A           N/A NMC-0805

eth158          07:00.3     8086 1522 ffff 0000  00:0b:ab:55:f8:91  cfg Slot 2          8 (*)                N/A           N/A NMC-0805

eth124          0a:00.0     8086 10fb ffff ffff  00:0b:ab:f4:7c:c2  cfg Slot 3          1 (*)                N/A           N/A NMC-1004

eth120          0a:00.1     8086 10fb ffff ffff  00:0b:ab:f4:7c:c3  cfg Slot 3          2 (*)                N/A           N/A NMC-1004

eth135          0f:00.0     8086 1533 ffff 0000  00:0b:ab:b6:b7:b9  Onboard I210_0      Onboard I210_0 (*)   Need cfg      N/A Onboard

(*) means default, user can use config file to overwrite the panel name.

(!) means CFG file feature is not matched with Adv default feature.
```

3.3 smbios_cfg Config File example

smbios.cfg:

```
TYPE="41"

NAME="Onboard I210_0"

BUS:DEV.FUN="00:1c.0"


TYPE="41"

NAME="Onboard I210_1"

BUS:DEV.FUN="00:1c.1"


TYPE="41"

NAME="Onboard 10GBe"

BUS:DEV.FUN="3c:00.0"


TYPE="9"

NAME="cfg Slot 1"

BUS:DEV.FUN="5d:00.0"
```

```
TYPE="9"
NAME="cfg Slot 2"
BUS:DEV.FUN="5d:02.0"

TYPE="9"
NAME="cfg Slot 3"
BUS:DEV.FUN="d7:00.0"

TYPE="9"
NAME="cfg Slot 4"
BUS:DEV.FUN="d7:02.0"

TYPE="9"
NAME="cfg Slot 5"
BUS:DEV.FUN="17:00.0"

TYPE="9"
NAME="cfg Slot 6"
BUS:DEV.FUN="17:02.0"

TYPE="9"
NAME="cfg Slot 7"
BUS:DEV.FUN="85:00.0"

#TYPE="9"
#NAME="cfg Slot 8"
#BUS:DEV.FUN="85:00.2"

#This smbios cfg file is to solve the problem of NO correct type 9/41 records in BIOS.
#Please use -smbios_cfg <file name> command.
#Please add correct type 9/41 records which are on your platform.
#Use this format
#
#TYPE="9"
#NAME="cfg Slot 8"
#BUS:DEV.FUN="85:00.2"
#
#TYPE="41"
#NAME="cfg on10GBe"
#BUS:DEV.FUN="3c:00.0"
```

Advantech Copyright 2024. All rights reserved.