H2OFFT™ (Flash Firmware Tool) User Guide for EFI Version

Revision 1.4
January 2020

Disclaimer

Contents

Revision History

Revision Number	Description	Author	Release Date
1.1o	Initial version.	Kevin Liang	May, 2012
1.2d	Update command, platform.ini and iFdPacker	Kevin Liang	August, 2012
1.2g	Update command, platform.ini and iFdPacker	Kevin Liang	September, 2012
1.2h	Update command, iFdPacker and support PFAT image update	Kevin Liang	October, 2012
1.2i	Update command, platform.ini and iFdPacker	Kevin Liang	January, 2013
1.2k	Update iFdPacker	Kevin Liang	February, 2013
1.2l	Update command	Jerry Yen	May, 2014
1.3	Update command and ini settings	Daniel Chang	June, 2018
1.4	Update command and ini settings	Daniel Chang	January, 2020

1 Introduction

1.1   Overview

Insyde H2OFFT Shell is a flash utility program from Insyde Software for EFI Shell environment. It provides a powerful and intelligent tool for updating and maintaining the computer system BIOS under EFI Shell environment. It also features a friendly user interface for saving, loading and updating the BIOS, as well as displaying the system BIOS information. This chapter provides a quick introduction on Insyde H2OFFT Shell and explains what it can do for you.

1.2   Support Functionality

Insyde H2OFFT Shell offers you the following functionality:

- Allows you to easily update the system BIOS under EFI Shell environment.

- Allows you to verify the system BIOS to ensure system reliability.

- Auto detects hardware settings so you will know if Insyde H2OFFT is compatible with your system or not. This gives enhanced security.

1.3   System Requirements

Installing Insyde H2OFFT Shell is quick and easy. However, you need to be aware of these requirements before installing the program:

- EFI Shell environment

- InsydeH2O BIOS-compatible motherboard (Insyde H2OFFT will auto detect the BIOS on motherboard if compatible)

- InsydeH2O BIOS with IHISI

1.4 Member

This Insyde H2OFFT Shell package including following files:

File	Description
Doc\H2OFFT_UserGuide_EFI.pdf	User guide of this utility.
Doc\ReleaseNote.txt	Release history.
Tools\iFdPacker.exe	Single-execution package tool. It can package the H2OFFT-S.efi (or H2OFFT-Sx64.efi) and BIOS binary file into single file to run.
Tools\PlatformIniEditor.exe	The editor for platofrm.ini.
Tools\platform.xml	The configure file of PlatformIniEditor.
H2OFFT-S.efi	The main executable file of Shell flash utility that supports x86 system.
H2OFFT-Sx64.efi	The main executable file of Shell flash utility that supports x64 system.
platform.ini	Flash utility option configuration file.

2 Using Insyde Flash

Insyde H2OFFT Shell (H2OFFT-Sx64.efi / H2OFFT-S.efi) is a tool that allows access to the BIOS ROM. It supports many features in this tool, using –h to view the supported commands.

All options below supports both –option and /option format.

Option	Description
-h	This flash utility help.
-ab	Check battery life percentage.
-ac	Do not check AC power.
-acb	Check AC power and battery life percentage.
-all	Force flash all protected regions.
-b	Force flash protected PEI region.
-bios	Flash BIOS region.
-dc	Disable comparison in normal flash process.
-desc	Flash descriptor region.
-di	Disable ID display.
-e:OFFSET,SIZE,ADDRESS	Update fix size from file offset to physical address. OFFSET, SIZE, ADDRESS are DWORD value in hex.
-ecbp	Update non-share EC and BIOS of EC merged BIOS binary, EC update block by block.
-ecp	Update non-share EC block by block.
-edt#@:"VALUE"	Update customizes data (such as logo with signature).
-eob	Update BIOS only of EC merged BIOS binary.
-extec:EC_FILE	Update Intel external EC region with input file.
-fd	Force flash protected DXE region.
-fe	Force flash protected EC region.
-fl	Force flash protected Logo region.
-fm	Force flash protected CPU Microcode region.
-fn	Force flash protected OEM NVS region.
-fp	Force flash protected Password region.
-ft:TYPE	Force flash specified protect type region.
-fv	Force flash protected Variable region.
-g	Read current ROM and save to file.
-gbe	Flash GbE region.
-mc	Skip model name and BIOS version check.
-me	Flash ME region.
-mfg	Run in manufacture mode.
-n	Do not reboot after flash.

-nativeme:ME_FILE	Update Intel native ME with input file.
-OemCus	Tell BIOS to do OEM customization feature.
-oemid:GUID	OEM ID for update Intel native ME, it only valid when –nativeme: used.
-pbi:TYPE	Flash specified type of BIOS protect region. TYPE is the protected region type BYTE value in hex.
-pdr	Flash Platform Data region.
-pi	Query BVDT protection/private region MAP in input BIOS file.
-pq	Query BIOS protection region MAP in current ROM.
-pr	Query external region MAP in current ROM.
-priv	Query BIOS private region MAP in current ROM.
-pw	Query whole region MAP in current ROM.
-pwd:PASSWORD	Input password for the feature which need password.
-r	Reboot after flash.
-rt:TIMES,DELAY	When SMI error, retry how many times and delay how many milliseconds between each retry.
-s	Shutdown after flash.
-u	Show confirm message.
-vrt:TIMES	When verify error, retry how many times.

2.1 Show help (-h)

H2OFFT-Sx64.efi -h

This feature will list all supported options of this tool on screen, allowing users to search through it.

2.2 Check battery life percentage (-ab)

H2OFFT-Sx64.efi BIOS_FILENAME –ac –ab

This feature needs to use the option "ac" to skip AC check when the AC is not plugged in (only use when AC is not plugged in and with -ac). This feature will check the battery percentage remaining. It must be more than the value that is set in BIOS.

2.3 Do not check AC power (-ac)

H2OFFT-Sx64.efi BIOS_FILENAME -ac

Before flashing the BIOS, the tool will check if the AC power is plugged in. If it is not, the tool will not continue to flash and pop a message to inform user. If you do not want to check it, please use this option.

2.4 Check AC power and battery life percentage (-acb)

H2OFFT-Sx64.efi BIOS_FILENAME -acb

Before flashing BIOS, the tool will check AC power and battery percentage. Both AC power and battery percentage need to pass the check.

2.5 Force flash all protected regions (-all)

H2OFFT-Sx64.efi BIOS_FILENAME -all

BIOS will report to the tool about which areas will be protected in the BIOS region. User can use this command to flash all BIOS protected regions.

2.6 Force Flash BIOS protected area

Since there are many protecting areas in the BIOS region, you can utilize these options to force flash specified protected areas.

H2OFFT-Sx64.efi FILENAME -option

Option	Description
-b	Force flash protected PEI region.
-fd	Force flash protected DXE region.
-fe	Force flash protected EC region.

-fl	Force flash protected Logo region.
-fm	Force flash protected CPU Microcode region.
-fn	Force flash protected OEM NVS region.
-fp	Force flash protected Password region.
-ft:TYPE	Force flash specified protect type region.
-fv	Force flash protected Variable region.

2.7 Flash the region that described in descriptor table (for Intel platform only)

H2OFFT-Sx64.efi BIOS_FILENAME [-desc | -me | -ec | -gbe | -bios | -pdr]

In descriptor mode (Intel platform with descriptor region), there may have more than two regions of the ROM. These commands will provide you the ability to flash the regions you want.

Note: These commands support on Intel platform only.

2.8 Disable comparison in normal flash process (-dc)

H2OFFT-Sx64.efi BIOS_FILENAME –dc

During flash, flash tool will do block comparison before write. The block comparison method is read block of write address from ROM and compare with the block which want to write. If block data is the same, flash tool will skip this block and process next. This option used to disable this compare feature, when it used all blocks will be write to ROM.

2.9 Disable ID display (-di)

H2OFFT-Sx64.efi BIOS_FILENAME -di

Tool will show the platform name and BIOS version during flash, and this option will disable the platform name display.

2.10 Update fix size from file offset to physical address (-e:)

H2OFFT-Sx64.efi BIOS_FILENAME -e:OFFSET,SIZE,ADDRESS

Tool will read file from the OFFSET by SIZE and update to the physical ADDRESS of BIOS.

2.11   Flash non-share EC with BIOS and show percentage (-ecbp)

H2OFFT-Sx64.efi FILENAME -ecbp

This option will update EC after BIOS update success. The FILENAME is the format that merged EC and BIOS binary (EC binary + BIOS binary directly without any header).

This feature will update EC block by block, not update EC in single SMI call. To learn more about this EC update SMI, please read IHISI spec v1.8.0 or later version

2.12   Flash non-share EC block by block (-ecp)

H2OFFT-Sx64.efi FILENAME -ecp

This option will flash non-share EC block by block.

2.13   Updates customized data (-edt:#@:)

H2OFFT-Sx64.efi [BIOS_FILENAME] -edt:#@:"value"

You can use -edt#@:"Value" for updating customized data (such as logo with signature) by IHISI.

> \# — from 4 ~ C.

> @ — F, S, W, DW

>> F — means file

>> S — means string

>> W — means word value

>> DW — means double word value

Example:

Update type 4 data, the source is file.

And update type 5 data, the source is string.

> -edt4f:logo.jpg -edt5s:"Input string"

Update a type 9 data, the source is WORD.

> -edt9w:"0x1234"

Update a type C data, the source is DWORD

> -edtcdw:"0x12345678"

2.14 Update BIOS only of EC merged BIOS binary (-eob)

H2OFFT-Sx64.efi BIOS_FILENAME -eob

The file is merged by EC and BIOS. This feature will flash BIOS and skip EC.

2.15 Update external EC region (-extec:)

H2OFFT-Sx64.efi –extec:EC_FILENAME

Update external EC region with input file.

Note: This command supports on Intel platform only.

2.16 Read current ROM and save to file (-g)

H2OFFT-Sx64.efi FILENAME -g

This option allows you to read the BIOS from IHISI to a file.

2.17 Skip model name and BIOS version check (-mc)

H2OFFT-Sx64.efi BIOS_FILENAME -mc

Flash tool will check the platform model name and BIOS version between input BIOS file and onboard BIOS. When it does not match, flash tool will show an error message and leave. If you don't want to use this option, you can use this command to skip it. As a result, the tool will not show and not check the model name and BIOS version.

2.18 Manufacture mode (-mfg)

H2OFFT-Sx64.efi BIOS_FILENAME -mfg

Application notifies BIOS that current system is in manufacturing mode. BIOS can do some special process while in manufacturing mode.

2.19 Do which action after flash

H2OFFT-Sx64.efi BIOS_FILENAME [-r | -s | -n]

Flash tool allow to do one of following three actions after flash; "reboot", "shutdown", and "nothing to do (return to OS)". The default is to reboot. If this is not your desired behavior, you can use "-n" to prevent system reboot or use "-s" to tell tool to shut down after flash.

2.20 Update ME region with native ME file (-nativeme:)

H2OFFT-Sx64.efi –nativeme:FILENAME [-oemid:GUID]

Update Intel ME region with input native ME file.

It allow to input an OEM ID via –oemid: command.

Note: This command supports on Intel platform only.

2.21   OEM customize token (-oemcus)

H2OFFT-Sx64.efi BIOS_FILENAME –oemcus CUSXXX

This command can pass the following string token to BIOS at program start. BIOS can use it to do some specific action.

The CUSXXX means the string start with "CUS", for example CUSV1, CUSDEBUG, CUSSKU2.

Note: One command only can follow one token.

When there is more than one token need to pass to BIOS, you can use "–oemcus CUSTOKEN1 –oemcus CUSTOKEN2".

2.22   Flash BIOS protect region (-pbi:)

H2OFFT-Sx64.efi FILENAME -pbi:TYPE

You can update partial region via specified protection region type. The type value in hex.

2.23   Query BVDT protection/private region MAP in input BIOS file (-pi)

H2OFFT-Sx64.efi BIOS_FILENAME -pi

Dump the BVDT protection and private region MAP in input BIOS file.

2.24   Query BIOS protection region MAP in current ROM (-pq)

H2OFFT-Sx64.efi -pq

Dump the protection MAP of BIOS region in current ROM.

2.25   Query external region MAP in current ROM (-pr)

H2OFFT-Sx64.efi -pr

Dump the Intel external region MAP in current ROM.

2.26 Query BIOS private region MAP in current ROM (-priv)

H2OFFT-Sx64.efi -priv

Dump the private MAP of BIOS region in current ROM.

2.27 Query whole region MAP in current ROM (-pw)

H2OFFT-Sx64.efi -pw

Dump the whole BIOS region MAP which defined in current ROM.

2.28 Input password (-pwd:)

H2OFFT-Sx64.efi BIOS_FILENAME –pwd:PASSWORD

When BIOS had set a system password, it may need to input password before update BIOS. This option can pass the password to BIOS before update.

2.29 When SMI error, retry how many times (-rt:)

H2OFFT-Sx64.efi BIOS_FILENAME -rt:TIMES,DELAY

When SMI error, retry the set amount of times and delay set amount of milliseconds between each retry.

2.30 Show confirm message (-u)

H2OFFT-Sx64.efi BIOS_FILENAME -u

Flash tool will ask if you want to flash before performing it.

2.31 When verify error, retry how many times (-vrt:)

H2OFFT-Sx64.efi -vrt:TIMES

When verify error, retry the set amount of times.

3 Customizing Insyde Flash

This chapter guides you on how to use Insyde H2OFFT Windows as well as understanding all its features.

Modifying the Configure File

Follow the steps below to modify the configuration file:

1. Open Platform.ini.

2. For each item in the configuration, change the values of related parameters as needed.

3. Save the file.

Supported Configuration Section List

[AC_Adapter]

[AutoWakeup]

[Bios_Version_Check]

[BIOSVersionFormat]

[CapsuleAudit]

[CommonFlash]

[FactoryCopy]

[FDFile]

[FlashComplete]

[FlashSecureBIOSOverride]

[ForceFlash]

[Log_file]

[MessageStringTable]

[MULTI_FD]

[Option]

[Others]

[ParamForBiosReference]

[PassToBios]

[PasswordCheck]

[PermitFlashConditionalData]

[PermitFlashVersion]

[Platform_Check]

[PlatformVersion]

[PreFlash]

[Region]

[ReturnCodeDefinition]

[ReturnErrorCode]

```
[SecureUpdate]
[UI]
[UpdateEC]
[UpdateExtraData]
[UpdateOEMME]
[UpdateDeviceFirmware]
[Version]
```

Valid Environment

Because some setting items only supported on specific OS flash. Here use following method to identify each setting item.

    d: means for DOS normal flash.

    w: means for Windows normal flash.

    s: means for Shell normal flash.

    D: means for DOS secure flash.

    W: means for Windows secure flash.

    S: means for Shell secure flash.

Normal flash means it write data to ROM in DOS/Windows/Shell flash.

Secure flash means it will pass capsule to BIOS (via memory or via ESP), and flash in UEFI environment after reboot.

3.1   [AC_Adapter]

This section used to set AC/DC check option.

Item	OS	Valid Value	Description
Flag	dwsDWS	**(default) 0:** 1:	Don't check AC. Check AC.
BatteryCheck	dwsDWS	**(default) 0:** 1:	Don't check battery life percentage. Check battery life percentage.
BatteryBound	dwsDWS	Integer (1 ~ 100)	Low battery boundary (percentage). When BatteryCheck=1 this value will be referenced. And only when the battery life percentage is bigger than inputted value, it can do flash.
LauncherAcWarning	dsDS	String	**(default) messagestringA02** A key name which list in [MessageStringTable].
SecurityAcWarning	dwsDWS	String	**(default) messagestringA02** A key name which list in [MessageStringTable].
AcNotConnectError	wW	String	**(default) messagestringA05** A key name which list in [MessageStringTable].
DcNotConnectError	wW	String	**(default) messagestringA06** A key name which list in [MessageStringTable].
DcNotEnoughError	wW	String	**(default) messagestringA07** A key name which list in [MessageStringTable].

3.2　　[AutoWakeup]

This section used to auto wake up system when flash update is completed and system shutdown.

Item	OS	Valid Value	Description
Flag	wsDWS	**(default) 0:** 1:	Disable. Enable auto power on via RTC.
Interval	wsDWS	Integer	**(default) 120** An integer value unit is second. Interval time after system turn off.

3.3　　[Bios_Version_Check]

This section used to set firmware version check option.

Item	OS	Valid Value	Description
Flag	dwsDWS	0: 1: **(default) 2:** 3: 4:	Don't check BIOS version. Check BIOS version by tool. When BIOS version in file is older than onboard BIOS, it will display a warning message and close application. Does tool need to do version check is depend on onboard BIOS report. When BIOS report not to check from IHISI, it's the same as Flag=0. When BIOS report need to check from IHISI, it's the same as Flag=1. Check BIOS version by BIOS only. (Application will not do version check.) Check BIOS version by BIOS and application. After it pass version check by BIOS, it will do version check by application. Only pass both BIOS and application check allow to flash.
SameVersionErrorMessage	wsWS	String	**(default) "The version of ROM file is the same as that of BIOS."** User defined error message when input BIOS version is the same as platform BIOS version.
OlderVersionErrorMessage	wsWS	String	**(default) "The version of ROM file is not newer than that of BIOS."** User defined error message when input BIOS version is older than platform BIOS version.
CheckByBiosErrorMessage	dwsDWS	String	**(default) "This BIOS file is not allow to flash. The flash process will be terminated."** User defined error message when BIOS is not allow to flash this version image.

3.4 [BIOSVersionFormat]

This section used to define the format of BIOS version, this format will be used on version check feature.

Item	OS	Valid Value	Description
BIOSVFEnable	dwsDWS	**(default) 0:**	Function is disabled.
		1:	Function is enabled.
VersionFormat	dwsDWS	X:	The field is masked. It will NOT be compared.
		N:	The digit field can be ASCII, case-sensitive.
		T:	It's the same definition with N. But T is a case-insensitive.
		.:	Dot is also a mask. It will NOT be compared.
		D:	Don't care field. It will NOT be compared.
			This field can be empty.
			It only allow to put at start or end of the version format.
			For example:
			Onboard version 1.21B flash to 1.22, VersionFormat must be N.NND.
			N.NND means the valid format is N.NN and N.NNX, the 5th character will be ignore in version compare.

3.5 [CapsuleAudit]

This section used to choice the update capsule base on onboard public key.

Item	OS	Valid Value	Description
Flag	dw	**(default) 0:**	Disable.
		1:	Enable capsule update audit.
OAKey	dw	String	**(default) empty** Capsule file which sign with Insyde QA test key.
OemKey	dw	String	**(default) empty** Capsule file which sign with OEM/ODM key.

3.6   [CommonFlash]

This section only available for specific ODM.

NOTE:

To disable this section need to remove entire [CommonFlash] section.

When enable and BIOS supported, power check will take care by IHISI, and NOT reference [AC_Adapter].

Item	OS	Valid Value	Description
SwitchString	dwsDWS	String	**(default) empty** A switch flag setting string. Ex: "CPVER:[1] ACEN DCEN FHRST" Detail parameter please reference following Switch String Parameter Table.
ErrorMsg00	dwsDWS	String	**(default) empty** The message of no error.
ErrorMsg01	dwsDWS	String	**(default) empty** AC error message.
ErrorMsg02	dwsDWS	String	**(default) empty** DC error message.
ErrorMsg03	dwsDWS	String	**(default) empty** DC gas gauge under xx% message.
ErrorMsg04	dwsDWS	String	**(default) empty** BIOS version error message.
ErrorMsg05	dwsDWS	String	**(default) empty** Model name error message.
ErrorMsg10	dwsDWS	String	**(default) empty** No support this version of Flash Common Interface message.
ErrorMsg##	dwsDWS	String	**(default) empty** The error message of ##. The number of ErrorMsg## is in hex.

Switch String Parameter Table

Parameter	Description
PTEN	All protection enable.
PTDIS	All protection disable.
ACEN	AC protect checking enable.
ACDIS	AC protect checking disable.
DCEN	DC & Gangue protect checking enable.
DCDIS	DC & Gangue protect checking disable.
RESSEN	BIOS Regression enable.
RESSDIS	BIOS Regression disable.
PJMDEN	Project Model string protect checking enable.
PJMDDIS	Project Model string protect checking disable.
FHOS	System back to OS after flash BIOS completely.
FHST	System directly shutdown after flash BIOS completely.
FHRST	System directly reboot after flash BIOS completely.
CPVER:[Num]	Common Flash Version information ex: [Num] is decimal and start from 1.

3.7 [FactoryCopy]

Factory copy is a region in BIOS which is used to store default keys of Secure Boot.

This section used to restore the default data for factory copy region.

Item	OS	Valid Value	Description
Restore	DWS	**(default) 0:** 1:	Don't do FACTORY_COPY restore after flash. Restore FACTORY_COPY after flash BIOS.

3.8 [FDFile]

Item	OS	Valid Value	Description
FileName	wW	String	**(default) empty** Utility always load this file. If the FileName is empty, utility will search current directory and load the first found FD file.

3.9 [FlashComplete]

This section set the end of flash action after writing ROM.

Item	OS	Valid Value	Description
Action	dwsDWS	0: 1: **(default) 2:**	Do nothing. Return to OS. Shutdown. Reboot.
Dialog	W	**(default) 0:** 1: 2:	Don't display dialog. Display dialog. Display dialog and wait several seconds.
Counter	W	**(default) 15:**	An integer value, unit is second. The number of seconds for countdown to reboot or shutdown.
ActionOverride	W	**(default) 0:** 1:	This key is disabled. Flash utility bases action key setting to do original behavior. This key is enabled. Flash utility just does close itself in silent mode.
Pause	dwsDWS	**(default) 0:** 1:	Not pause after flash complete. Pause after flash complete.
PauseWarning	dwsDWS	String	**(default) messagestringA03** A key name which list in [MessageStringTable].

3.10 [FlashSecureBIOSOverride]

Item	OS	Valid Value	Description
EnableFlashSecureBIOSOverride	DWS	**(default) 0:** 1:	Disable action override. Use the action which returned from BIOS. Enable the action override when flashing secure BIOS in OS.
Action	DWS	0: **(default) 1:** 2: 3:	S3 Reboot. Shutdown. Do nothing.

3.11 [ForceFlash]

Item	OS	Valid Value	Description
ALL	dwsDWS	**(default) 0:** 1:	Reserve all protected areas. Flash all protected areas.
BB_PEI **CPU_Microcode** **Variable** **DXE** **EC** **Password** **OEM_NVS** **Logo**	dwsDWS	**(default) 0:** 1:	Protect this area. Force flash this area. They are the predefined types each one indicate a type value. BB_PEI = Type#00 CPU_Microcode = Type#01 Variable = Type#02 DXE = Type#03 EC = Type#04 Password = Type#0F OEM_NVS = Type#10 Logo = Type#05
Type#NN	dwsDWS	**(default) 0:** 1:	Protect this area. Force flash this area. Type#NN is an extended method for those didn't have predefine type. The NN is a number in Hex. For example: If BIOS report to protect type 13h from IHISI, and you want to flash this protect area. Then the setting in platform.ini can add Type#13=1.
UseBvdtRomMap	DWS	**(default) 0:** 1:	Disable private/protection ROM map override feature. AP will reference IHISI report to protect regions. Enable private/protection ROM map override feature. AP will reference BVDT defined map to protect regions.

3.12 [Log_file]

Enable it will generate a log file for debugging purposes.

Item	OS	Valid Value	Description
Flag	wsWS	**(default) 0:** 1:	Don't log to file. Utility will log to specify file.
FileName	wsWS	String	**(default) H2OFFT.log** Log file name string. It can contain full path for logging to specific location. Only file name will log to the same folder of execution file. Ex: D:\H2OFFT.log
CMOS_Flag	wW	**(default) 0:** 1:	Disable CMOS debug. Enable CMOS debug.
CMOS_INDEX_PORT	wW	**(default) 70:** 72:	Use 0x70 port as index port. Use 0x72 port as index port.
CMOS_DATA_PORT	wW	**(default) 71:** 73:	Use 0x71 port as index port. Use 0x73 port as index port.
CMOS_OFFSET	wW	**(default) 0,0:**	CMOS_OFFSET=xx,yy the xx is high byte offset; yy is low byte offset, and the range is 0~FF (hex)

3.13   [MessageStringTable]

Provide message string table to define customize message.

Item	OS	Valid Value	Description

messagestring1
messagestringA00
messagestringA01
messagestringA02
messagestringA03
messagestringA04
messagestringA05
messagestringA06
messagestringA07
	dwsDWS	String	**(default) empty**
			The message string must as following format
			    messagestring#="Your message here."
			The # is a number in Decimal or Hex.
			If a multi-line message is required, you can use "\n"
			    in message string for new line.

3.14   [MULTI_FD]

Some of flash package would contain multiply firmware image for different SKU. Following setting can be configured what condition is to detect firmware image.

Item	OS	Valid Value	Description
Flag	dwsDWS	**(default) 0:** 1:	Disable. Enable. Multi-FD mode. It will reference FD#XX values.
FD#XX	dwsDWS	FD#01 ~ FD#99	XX is decimal number from 01 to 99. The setting is a string with following format: **condition type, condition**

Condition of ID

ID, [Model Name], [BIOS Filename] [ME Filename] [INI Filename]	 The platform ID, model name string. BIOS filename. ME filename. If it exists, utility will run OEMME flash feature. INI filename for override.

Condition of IO

IO, [Offset], [Mask], [Value], [BIOS Filename] [ME Filename] [INI Filename]	 Offset in hex. IO type supports BYTE, WORD and DWORD in hex. IO type supports BYTE, WORD and DWORD in hex. BIOS filename. ME filename. If it exists, utility will run OEMME flash feature. INI filename for override.

Condition of PCI

PCI, [Bus], [Device], [Function], [Offset], [Mask], [Value], [BIOS Filename] [ME Filename] [INI Filename]	 Bus number Device number Function number Offset in hex. PCI type supports DWORD in hex only. PCI type supports DWORD in hex only. BIOS filename. ME filename. If it exists, utility will run OEMME flash feature. INI filename for override.

Condition of CPUID

CPUID, [Mask], [Value], [BIOS Filename] [ME Filename] [INI Filename]	 CPUID type supports DWORD in hex only. CPUID type supports DWORD in hex only. BIOS filename. ME filename. If it exists, utility will run OEMME flash feature. INI filename for override.

Condition of CPUMSR

CPUMSR,	
[Offset],	Offset in hex.
[Mask],	CPUMSR type supports QWORD in hex only.
[Value],	CPUMSR type supports QWORD in hex only.
[BIOS Filename]	BIOS filename.
[ME Filename]	ME filename. If it exists, utility will run OEMME flash feature.
[INI Filename]	INI filename for override.

Condition of OS

OS,	
[OS Version],	32bit or 64bit OS. 32 for 32bit OS, 64 for 64bit OS.
[BIOS Filename]	BIOS filename.
[ME Filename]	ME filename. If it exists, utility will run OEMME flash feature.
[INI Filename]	INI filename for override.

Condition of MEMORY

MEMORY,	
[Physical Address],	A DWORD value in hex. Value range 0xFE000000 ~ 0xFFFFFFFF.
[Mask],	MEMORY type supports BYTE, WORD and DWORD in hex.
[Value],	MEMORY type supports BYTE, WORD and DWORD in hex.
[BIOS Filename]	BIOS filename.
[ME Filename]	ME filename. If it exists, utility will run OEMME flash feature.
[INI Filename]	INI filename for override.

3.15   [Option]

This section used to set the flash tool starting UI or just start flash.

Item	OS	Valid Value	Description
Flag	wW	**(default) 0:**	Auto flash mode.
		1:	User option mode, including option, start, exit buttons. (Option button will disable on secure flash mode.)
		2:	User flash mode, including start, exit buttons.

3.16　　[Others]

Item	OS	Valid Value	Description
DisableCompare	wsWS	(default) 0: 1:	Read BIOS and compare difference before writing. If the read data is the same as the data we want to write, it will not do the write action. Don't do compare before writing. Just do write action directly.
DisableVerify	wsWS	(default) 0: 1:	Verify BIOS after writing. Don't verify BIOS after writing.
VerifyErrorRetry	wsWS	Integer	(default) 3 Retry times. If the value is not zero means enable verify retry, and will retry set times. 0 for disable verify retry.
SMIErrorRetryDelay	wsWS	Integer	(default) 100 Delay time during SMI error retry (millisecond).
SMIErrorRetry	wsWS	Integer	(default) 5 Retry times when SMI fail. **Note: This may let flash time become longer.**
ForceIHISIVersion	w	String	(default) empty This flag will force override the IHISI version which will passing to BIOS. Ex: 1.9.1
EnablePrivateRegionMovement	wsDWS	(default) 0: 1:	Disable private region movement. Enable private region movement. It will backup orginal private regions and relocate to new address which defined in new image BVDT when update whole BIOS.
EnableProtectionRegionMovement	sDWS	(default) 0: 1:	Disable protection region movement. Enable protection region movement. It will backup orginal protection regions and relocate to new address which defined in new image BVDT when update whole BIOS. **This flag cannot be used with [ForceFlash] all=1.**
DisableSecureCapsuleFlash	dwsDWS	(default) 0: 1:	Enable flash secure BIOS on normal platform. Disable flash secure BIOS on normal platform.
BootSafeMode	wW	Format: [Number] (default) 0: 1: [Version String] String	**[Number],[Version String]** Disable this feature. Normal boot. Enable this feature. Next boot into Windows safe mode. **(default) empty** The limit version. When onboard BIOS version smaller than or equal to this version, it allow to enable this feature. When onboard BIOS version bigger than this version, this feature always disable even the value is set to 1.

3.17　　[ParamForBiosReference]

This section used to pass specific parameter string to BIOS via IHISI 19h.

Item	OS	Valid Value	Description
Param1 ~ 20	dwsDWS	String	(default) empty The parameter string you want to pass to BIOS.

3.18 [PassToBios]

This section only used to pass settings to BIOS, application won't do any action of the settings.

Item	OS	Valid Value	Description
ClearTXE	dwsDWS	(default) 0: 1:	Disable. Tell BIOS to clear TXE at this flash.

3.19 [PasswordCheck]

This section used to set the dialog UI and message for password check.

Item	OS	Valid Value	Description
PasswordCheckEnable	wW	(default) 0: 1:	Disable the password check function. Enable the password check function.
MsgCaption	wW	String	**(default) "Password confirmation"** The dialog box caption.
MsgPreface	wW	String	**(default) "Please enter password"** The preface of password entering.
MsgSuccess	wW	String	**(default) "Password is correct."** The string displayed when the comparing result returned by BIOS is 0x00.
MsgFailure	wW	String	**(default) "Password is incorrect. Please retry again."** The string displayed when the comparing result returned by BIOS is 0x01.

3.20   [PermitFlashConditionalData]

Item	OS	Valid Value	Description
PFCDFunctionEnable	d	**(default) 0:** 1:	Function is disabled. Function is enabled. The function is enabled or disabled by the flag. If the flag is disabled, application will keep flash progress without conditional check.
Conditionalcount	d	Integer	**(default) 0** Total count of conditional data.
Conditionaldata1 ~ 20	d	Format	[Data Type],[Data Format],[Data Field],[Message String]
	d	[Data Type]	The type of Conditionaldata, when you have more than one conditional data need to check before flash, you can give them different type. It will allow flash when all types are pass. It also can use the same type with more than one conditional data, which means if one of the conditional data is pass, this type is pass. Ex: We have the following settings, Conditionaldata1 is type 0, and others are type 1. Conditionalcount=3 Conditionaldata1=0,w,"0x1234","This BIOS is not allow to flash." Conditionaldata2=1,w,"0x1111","This BIOS is not allow to flash." Conditionaldata3=1,w,"0x1222","This BIOS is not allow to flash." When Conditionaldata1 is pass and Conditionaldata2 or Conditionaldata 3 is pass that will allow flash. Allow = data1 pass && (data2 pass || data3 pass).
	d	[Data Format] b: s: w: d:	Can support different as binary, string, WORD, DWORD. The data format is binary. The data format is string. The data format is WORD. The data format is DWORD.
	d	[Data Field] BINARY: String: WORD: DWORD:	Relative data base on data format. If the data format is binary. Data field is HEX. The data is as like "AA BB 00 12 55". If the data format is string. Data field is string. The data is as like "Data.123.AABE.". If the data format is WORD. Data field is HEX and unit is a WORD length. The data is as like "0x1223". If the data format is DWORD. Data field is HEX and unit is a DWORD length. The data is as like "0x12345678".
	d	[Message string]	Define customize message string here. The message is pop up when flash utility get a process interrupted from BIOS. The message string is optional field. It will not pop any message if you do not provide message string in this field.

3.21   [PermitFlashVersion]

This section is for conditional version definition for BIOS update.

When this function enabled, flash utility will compare on board BIOS version with conditional version and then decide to continue flash progress or not.

For example: In following condition PFVFunctionEnable=1, SingleVersion=A03, "Please update to BIOS version A03 first!".

　　When on board BIOS version is older than A03, it will pop up the error dialog with the message "Please update to BIOS version A03 first!" and then terminated.

　　Otherwise it will continue original flash process.

The MultiVersion1~20 are similar as SingleVersion, but they only available when the [Platform_Check] is enable to compare the 20 platform IDs.

They must be pair of the PlatformName of [Platform_Check] section.

For example: When the platform ID matches with PlatformName2, the Version2 will be used.

Item	OS	Valid Value	Description
PFVFunctionEnable	wW	**(default) 0:** 1:	Disable conditional version check. Enable conditional version check.
SingleVersion	wW	String	**(default) empty** String format is **XXXX, "Message String"** XXXX is the conditional version, regarding version check rule please refer section "BIOSVersionFormat". "Message String" is the message to show when the on board BIOS version is older than the conditional version.
MultiVersion1 ~ 20	wW	String	**(default) empty** String format is **XXXX, "Message String"** XXXX is the conditional version, regarding version check rule please refer section "BIOSVersionFormat". "Message String" is the message to show when the on board BIOS version is older than the conditional version.

3.22   [Platform_Check]

This section used to enable or disable model name check and set platform ID (model name) compare source

Item	OS	Valid Value	Description
Flag	dwsDWS	0: 1: 2: (default) 3:	Don't check project ID. Check project ID of new file. If ID is different with current BIOS, the utility will close. Utility will compare current platform ID with the 20 platform IDs. If anyone is match, it will go ahead, otherwise utility will close. Does tool need to do platform check is depend on onboard BIOS report. When BIOS report not to check from IHISI, it's the same as Flag=0. When BIOS report need to check from IHISI, it's the same as Flag=1.
NotTargetPlatformErrorMessage	wsWS	String	**(default) "This BIOS is not for your Notebook PC."** User defined error message when input BIOS model name is not match with platform model name.
PlatformName1~20	dwsDWS	String	**(default) empty** If ROM file do not contain correct ID, user can define ID here.

3.23   [PlatformVersion]

This section only available when the [Platform_Check] is enable to compare the 20 platform IDs.

The Version is pair with the PlatformName.

For example: When the platform ID matches with PlatformName2, the Version2 will be used.

Item	OS	Valid Value	Description
Flag	dwsDWS	**(default) 0:** 1:	Don't use multi version. Use the version in the list instead of the version in file.
Version1~20	dwsDWS	String	**(default) empty** If ROM file do not contain correct version, user can define version here.

3.24   [PreFlash]

Item	OS	Valid Value	Description
EraseDataType	DWS	Hex Value	**(default) empty** The address and size of the data type must reported via IHISI. When the data type is not empty, flash utility will erase (fill with 0xFF) the area of the specific data type before flash. Note: In secure BIOS, the NV Storage will be always protected. If the specific data type is NV Storage, it will not take effect.

3.25 [Region]

This section is used to control SPI descriptor mode.

Default is flash all regions when the values all set to 0.

If any one of the regions set to 1, it will only flash specific regions.

If the BIOS is built without additional Intel firmware as like ME, GBE and Descriptor or BIOS is an AMD firmware which does not support ME, please ignore this section.

Item	OS	Valid Value		Description
BIOS	dwsDWS	**(default) 0:**		Don't flash.
		1:		Flash BIOS region.
GbE	dwsDWS	**(default) 0:**		Don't flash.
		1:		Flash GbE region.
ME	dwsDWS	**(default) 0:**		Don't flash.
		1:		Flash ME region.
EC	dwsDWS	**(default) 0:**		Don't flash.
		1:		Flash EC region.
DESC	dwsDWS	**(default) 0:**		Don't flash.
		1:		Flash Descriptor region.
Platform_Data	dwsDWS	**(default) 0:**		Don't flash.
		1:		Flash Platform Data region.

3.26 [ReturnCodeDefinition]

This section allow to set different value for silent mode and non-silent mode (normal mode).

The value behind comma is the return value in silent mode.

Ex: value=0,3010

It will return 3010 in silent mode and return 0 in normal mode.

Item	OS	Valid Value	Description
RETURN_SUCCESSFUL	wW	Integer	**(default) 0** Use input value as return code. 99999: Use the default value above.
RETURN_MODEL_CHECK_FAIL	wW	Integer	**(default) 259** Use input value as return code. 99999: Use the default value above.
RETURN_USER_CONFIRM_CANCEL	wW	Integer	**(default) 1602** Use input value as return code. 99999: Use the default value above.
RETURN_AC_NOT_CONNECT	wW	Integer	**(default) 1602** Use input value as return code. 99999: Use the default value above.
RETURN_LOAD_DRIVER_FAIL	wW	Integer	**(default) 259** Use input value as return code. 99999: Use the default value above.
RETURN_NEED_REBOOT	wW	Integer	**(default) 3010** Use input value as return code. 99999: Use the default value above.
RETURN_USER_EXIT	wW	Integer	**(default) 1602** Use input value as return code. 99999: Use the default value above.
RETURN_SAME_VERSION_CHECK	wW	Integer	**(default) 1602** Use input value as return code. 99999: Use the default value above.

3.27 [ReturnErrorCode]

This section used to set different value for silent mode and non-silent mode (normal mode).

The value behind comma is the return value in silent mode.

Ex: value=3,259

It will return 259 in silent mode and return 3 in normal mode.

Item	OS	Valid Value	Description
FileNotFound	wW	Integer	**(default) 3** If utility can't find the BIOS file, it will return this error code.
ErrorBeforeFlash	wW	Integer	**(default) 4** If error occur before flash process, it will return this error code.
BatteryNotConnect	wW	Integer	**(default) 4** If battery not connect, it will return this error code. (This error is separated from ErrorBeforeFlash.)
BatteryCapacityNotEnough	wW	Integer	**(default) 4** If battery capacity not enough, it will return this error code. (This error is separated from ErrorBeforeFlash.)
WriteROMFail	wW	Integer	**(default) 5** If error occur during write ROM process, it will return this error code.
WriteECFail	wW	Integer	**(default) 6** If error occur during write EC process, it will return this error code.
WriteExtraDataFail	wW	Integer	**(default) 7** If error occur during write Extra Data process, it will return this error code.

3.28   [SecureUpdate]

In secure flash mode, we need somewhere to temporarily save the secure flash capsule. The below flag is to decide whether the capsule is put in ESP or default is put memory space.

Item	OS	Valid Value	Description
viaESP	WS	0:	Disable. Pass capsule via memory only.
		1:	Write the capsule to ESP (EFI system partition). Pass capsule via ESP only. Onboard BIOS need to support via ESP method, if BIOS not support then the update failed.
		2:	Pass capsule via ESP and memory. It will copy capsule to ESP then pass capsule to BIOS via memory. It success when both steps are success, and failure when any one failed.
		(default)3:	Pass capsule via ESP or memory. It will try to update capsule via ESP first. When via ESP method failed it will use pass capsule to BIOS via memory. When via ESP method success it will return success. **NOTE: DOS flash does not support via ESP method, if the setting need to run in DOS please set viaESP as 0 or 3.**
ViaESPCopyFileWaitingTime	WS	Integer	**(default) 0** The waiting time after ESP file copy, unit is second. To prevent file copy success but the file not actually write to disk, you can use this delay to wait hardware action done. The flash utility also has a mechanism to read back and compare to make sure the data is correct.
FreeEspSpaceWhenNeed	W	0: **(default) 1:**	Do not try to delete file in ESP. Try to delete .log file in root folder of ESP.
DeviceOrder	S	String	**(default) eMMC, NVMe, SATA, ATAPI, USB** The FAT device detection sequence of secure flash via ESP feature. Now we support eMMC, NVMe, SATA, ATAPI, USB.
PhysicalMemoryAllocateFail RetryTimes	W	Integer	**(default) 3** Retry times. If the value is not zero means enable physical memory allocate fail retry, and will retry the setting times. 0 for disable retry.
PhysicalMemoryAllocateFail RetryDelay	W	Integer	**(default) 500 (Unit is millisecond)** Delay time during secure update physical memory allocate fail retry (millisecond).

3.29 [UI]

This section used to set the UI related configure.

Item	OS	Valid Value	Description
Confirm	wW	0:	Don't display confirm dialog.
		(default) 1:	Display confirm dialog.
Silent	wW	**(default) 0:**	Normal mode.
		1:	Silent mode, hide main dialog.
SilentWithDialog	wW	0:	Don't display any dialog.
		1:	Display main dialog only.
		(default) 2:	Display all dialogs except main dialog.
DisplayID	wW	0:	Don't display BIOS ID.
		(default) 1:	Display BIOS ID.
InsydeInfo	wW	0:	Don't display Insyde copyright information and URL.
		(default) 1:	Display Insyde copyright information and URL.
VersionInfo	wW	0:	Don't display BIOS version.
		(default) 1:	Display BIOS version.
GroupInfo	wW	0:	Don't display group box.
		(default) 1:	Display group box.
ConfirmInfo	wW	**(default) 0:**	Don't display BIOS version and build date information in confirm dialog.
		1:	Display BIOS version and build date information in confirm dialog.
ConfirmInfoRDate	wW	**(default) 0:**	Don't display BIOS version and release date information in confirm dialog.
		1:	Display BIOS version and release date information in confirm dialog.
			When this flag set to 1, the ConfirmInfo flag won't be referenced.
OnFlashingBeep	W	**(default) 0:**	Don't beep on flashing.
		1:	Beep on flashing.
OnFlashingBeepDelayTime	W	Integer	**(default) 800**
			Set beep delay time (milliseconds).
DisableMouseAndKeyboardInput	wW	**(default) 0:**	Don't hook mouse and keyboard.
		1:	Hook mouse and keyboard without "CTRL+ALT+DEL".
BeforeRunToolDelayTime	wW	**(default) 0:**	Delay number of seconds before flash process.
ProgramStartToWrongMessageBox	wW	**(default) 0:**	Don't pop-up warning dialog before flash process.
		1:	Pop-up a warning dialog before flash process.
GetFDFileButton	wW	**(default) 0:**	Hide FD file browse button.
		1:	Show the button for browsing FD file.
			NOTE: This setting only valid when [Option] Flag=2.
DelayBeforeFlash	wW	Integer	**(default) 0**
			Delay number of seconds before flashing.
ConfirmDialogCustomizeMessage	wW	String	**(default) empty**
			A key name which list in [MessageStringTable].
			For inserting additional notification message in confirm dialog.
ShowCustomizeMessageOnConfirmDialogOnly	wW	**(default) 0:**	Display default confirm message and insert the customize message on confirm dialog.
		1:	Do not display default confirm message, only show customize message on confirm dialog.
ShowUpdateROMAddress	d	**(default) 0:**	Show progress bar when updating BIOS.
		1:	Show ROM address when updating BIOS.

Elapse	dwsDWS	**(default) 0:**	Disable to show elapse time during progressing BIOS update.
		1:	Enable to show elapse time during progressing BIOS update.
DisableAccessRightWarning Message	wW	**(default) 0:**	It will show warning message when any region is locked (such as ME) in input image.
		1:	Do not show warning message when any region is locked in input image.
DisableAccessRightCheck	wW	**(default) 0:**	It will check access right when any one region flag of [Region] section is set.
		1:	It will not do access right check.
PassUpdateProgressToBios	DWS	**(default) 0:**	Do not pass update progress to BIOS.
		1:	It will pass update progress to BIOS during flash.
ShowEcUpdateProgress	DWS	**(default) 0:**	Hide the EC update progress during EC updating.
		1:	Show the EC update progress during EC updating.
ExtEcUpdateErrorMessage	WS	String	**(default) empty** The error to display when extern EC update failed.
ExtEcUpdateErrorMessage# ##	WS	String	**(default) empty** The error to display when extern EC update failed. ### is the return code (from IHISI) in Hex. Value range from 1 ~ FFFFFFFF. When the return code is 0x00000005 the ExtEcUpdateErrorMessage5 will be used. When the return code is 0x00100002 the ExtEcUpdateErrorMessage100002 will be used.
NotShowIconOnMessageBox	wW	**(default) 0:**	Show question, information, warning, error icon on message box.
		1:	Not show icon, only message string on message box.

3.30 [UpdateEC]

This section used to set the EC update related configure.

Item	OS	Valid Value	Description
Flag	w	**(default) 0:** 1:	Don't flash EC by BIOS. Flash EC by BIOS.
EC_Dialog	w	**(default) 0:** 1:	Don't display confirm dialog when begin to update EC. Show confirm dialog
BIOS_Only	w	**(default) 0:** 1:	Flash EC and BIOS file. Only flash BIOS part of the EC merged BIOS binary file.
EC_Only	w	**(default) 0:** 1:	Flash EC and BIOS file. Only flash EC binary file.
EC_Path	w	String	**(default) empty** EC filename.
EC_Compare	w	**(default) 0:** 1:	Don't do compare before writing. Just do write action directly. Read EC and compare difference before writing. If the read data is the same as the data we want to write, it will not do the write action.
EC_Verify	w	**(default) 0:** 1:	Don't verify EC. Verify EC after writing.
EC_VerifyErrorRetry	w	Integer	**(default) 3** Retry times. If the value is not zero means enable verify retry, and will retry the setting times. 0 for disable verify retry.
EC_BlockSize	WDS	Integer	**(default) 64** The block size of EC update. Unit is KB.
EC_DockWarning	dsS	String	**(default) messagestringA04** A key name which list in [MessageStringTable].

3.31 [UpdateOEMME]

The Intel firmware update tool (FWUpdLcl.exe) is dependent on each chipset generation, and the tool in release flash package is a sample and may be not suitable for your project.

Please remember to replace the FWUpdLcl.exe with right version before you will utilize function to update Intel firmware (ME or TXE).

Item	OS	Valid Value	Description
MEFileName	ws	String	**(default) empty** If this field not empty or Multi-FD ME filename exists, tool will use Intel firmware update tool to update ME in OS before flash BIOS.
CheckVersion	w	**(default) 0:** 1: 2: 3:	Don't check ME file version. Check ME file version. When ME flash error is cause of same version and downgrade version do not show error and continue to flash. When ME flash error, show error but continue to flash. **NOTE: This filed will not valid when Command field used.**
Command	ws	String	**(default) empty** When this field is empty and don't want to check ME version, utility will use "-f %filename -generic -allowsv" as default command. The %filename is a keyword which will be replaced with the value in MEFileName within this section or the filename in MULTI_FD section.

3.32　[UpdateDeviceFirmware]

Item	OS	Valid Value	Description
VerifyErrorRetry	DWS	Integer	**(default) 0 (Don't verify after update.)** Retry times. When this value bigger than 0, means enable verifying device firmware after writing and it will retry the setting times. 0 for disable verify retry.

3.33　[Version]

This section used to set the customize version for appending Insyde H2OFFT version and show on UI.

| Item	OS	Valid Value	Description
Version	wW	String	**(default) empty** Version number string for display. It will be show on UI, and append on current version number. When current version is 5.01 and this version string set to "12", then it will show "H2OFFT V5.01.12" on main dialog caption.

4 Tools in Insyde Flash

4.1 Using iFdPacker

The "iFdPacker.exe" is the Insyde FD packer utility for you to pack flash utility, binary files, setting file (platform.ini) etc. and generate an executable file.

To use Insyde FD Packer, follow these procedures:

4.1.1 Pack Insyde Flash with UI (For Shell Setting)
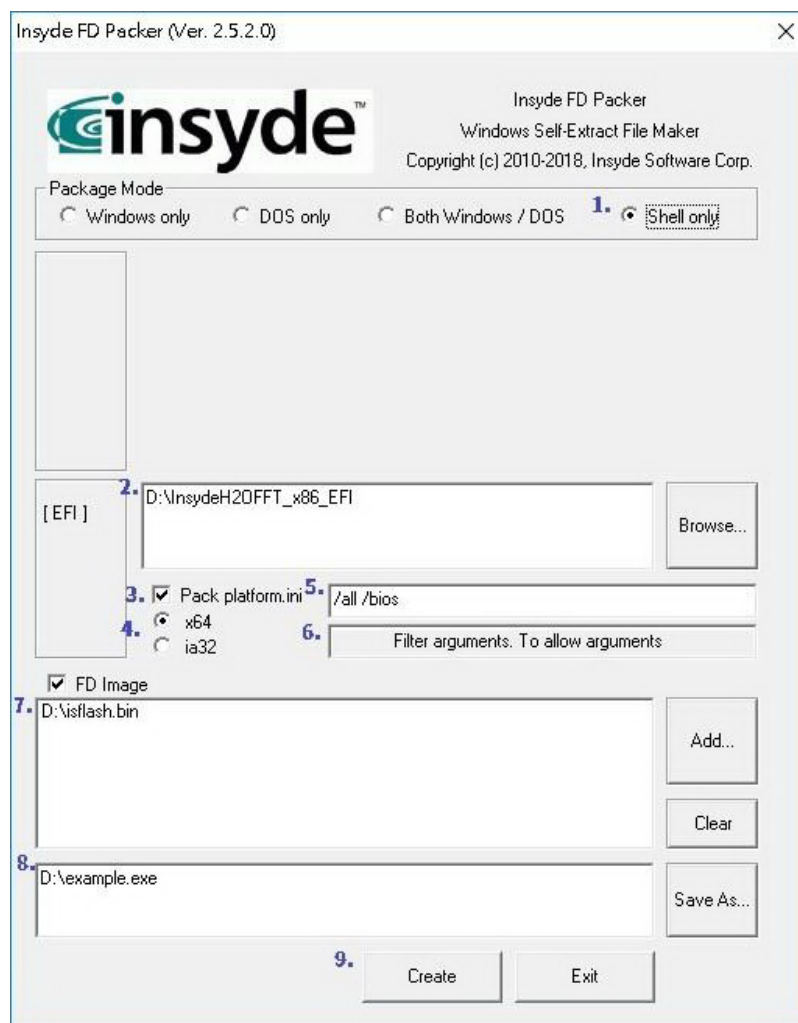
1. Select Package mode item.
2. Click [**Browse...**] to select directory of H2OFFT-S package.

3. Checked the check box if using external configure file.

4. Select EFI mode.

5. Input default argument which will auto be used when packaged execution file launched.

6. Input filter argument which will allow to be used with packaged execution file.

7. Click [**Add...**] to select BIOS image.

8. Click [**Save As...**] to select output file folder and specify a file name.

9. Click [**Create**] to pack.

10. If everything is OK, "Create package finished" message box will appear. Click [**OK**] to close it.

4.1.2 Pack Insyde Flash with Command

Run *iFdPacker.exe –h* in command prompt to view following usage.

4.1.2.1 Usage

iFdPacker.exe -winsrc PATH [-winini] [-protect] -b [3264 | 32 | 64] [-fv FILE [-fv FILE]] [-winarg "flag"] -output FILE

-dossrc PATH [-dosini] -fv FILE [-arg "flag"] [-argfilter "flag"] -output FILE

-shlsrc PATH [-shlini] –b [32 | 64] -fv FILE [-arg "flag"] [-argfilter "flag"] -output FILE

-winsrc PATH [-winini] [-protect] -b [3264 | 32 | 64] -dossrc PATH [-dosini] -fv FILE [-arg "flag"] [-argfilter "flag"] -output FILE

4.1.2.2 Command Description

Command	Description
-winsrc PATH	The path of Windows flash
-dossrc PATH	The path of DOS flash
-shlsrc PATH	The path of Shell flash
-b [3264 | 32 | 64]	The Windows flash build type 3264 – 32 bit Application runs on 32/64 bit OS (Windows only) 32 - 32 bit Application runs on 32 bit OS 64 - 64 bit Application runs on 64 bit OS
-winini	Windows package want to pack platform.ini
-dosini	DOS package want to pack platform.ini
-shlini	Shell package want to pack platform.ini
-protect	Protect packed package that will not allow to unzip or modify with third-party program.
-winarg "flag"	Argument with quotation marks for Windows package
-arg "flag"	Argument with quotation marks for DOS/Shell package
-argfilter "flag"	Argument filter with quotation marks for DOS/Shell package
-fv FILE	The path of firmware file
-output FILE	Specify the single package file
-h	The usage message

![insyde logo]

4.1.2.3 Example

a. Pack Windows Only:
iFdPacker.exe -winsrc d:\Insyde\H2OFFT_ WIN
 -winini
 -b 3264
 -fv d:\file1.fv
 -fv d:\file2.fd
 -output d:\OutputFile\myFlashUtility.exe

b. Pack Dos Only:
iFdPacker.exe -dossrc d:\Insyde\H2OFFT_DOS
 -dosini
 -fv d:\file1.fv
 -arg "-bios -all"
 -argfilter "-fv"
 -output d:\OutputFile\myFlashUtility.exe

c. Pack Shell Only:
iFdPacker.exe -shlsrc d:\Insyde\H2OFFT_EFI
 -shlini
 -fv d:\file1.fv
 -arg "-bios -all"
 -argfilter "-fv"
 -output d:\OutputFile\myFlashUtility.efi

d. Pack both Windows and DOS:
iFdPacker.exe -winsrc d:\Insyde\H2OFFT_ WIN
 -winini
 -b 3264
 -dossrc d:\Insyde\H2OFFT_DOS
 -dosini
 -fv d:\file1.fv
 -arg "-bios"
 -output d:\OutputFile\myFlashUtility.exe