H2OFFT™ (Flash Firmware Tool)

User Guide for Windows Version

Rev. 5.4 Approved
Prepared by DR-1 Team
June 07, 2018
Insyde Software Corp.

Disclaimer

Insyde Software Corp. provides this document and the programs "as is" without warranty of any kind, either expressed or implied, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose.

This document could contain technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in future revisions of this document. Insyde Software Corp. is under no obligation to notify any person of the changes.

The following trademarks are used in this document:
insyde and InsydeH2O are registered trademarks of Insyde Software Corp.

All other trademarks or trade names are property of their respective holders.

Revision History

Rev.	Date	Summary
3.72	09/27/2009	Initial version
3.93	05/19/2010	Update platform.ini & command setting
3.98	09/08/2010	Update platform.ini
4.02	10/06/2010	Update platform.ini & command setting
4.04	10/30/2010	Update platform.ini & command setting & iFdPacker explain
5.00	12/24/2010	Update platform.ini & iFdPacker version
5.02	3/14/2011	Update platform.ini & iFdPacker version
5.03	4/14/2011	Update platform.ini & command setting
5.05	5/05/2011	Update platform.ini & iFdPacker version
5.06	5/25/2011	Update platform.ini & iFdPacker version
5.08	9/02/2011	Update platform.ini & command setting
5.13	02/24/2012	Update platform.ini & command setting
5.16	03/20/2012	Update platform.ini & command setting & iFdPacker version
5.19	05/24/2012	Update platform.ini & command setting
5.20	06/21/2012	Update platform.ini
5.22	08/07/2012	Update platform.ini & iFdPacker version
5.24	08/15/2012	Update platform.ini & iFdPacker version
5.26	09/19/2012	Update platform.ini & command setting & iFdPacker version
5.27	10/12/2012	Update iFdPacker version & support PFAT image update
5.29	01/14/2013	Update platform.ini & command setting
5.30	02/07/2013	Update iFdPacker version
5.31	05/09/2014	Update platform.ini & command setting
5.4	06/07/2018	Update Commands, platform.ini and ifdpacker.

Table of Contents

1. Introduction

1.1. Overview

H2OFFT is a flash utility provided by Insyde Software. It serves as a powerful and intelligent tool for updating and maintaining the computer BIOS under various Windows environments. H2OFFT also features a friendly graphical user interface for saving, loading and updating the BIOS, as well as displaying the system BIOS information.

H2OFFT (for windows) works under the Windows 2000, Windows XP, Windows Vista, Windows 7 and Windows 8 operating systems. This chapter provides a quick introduction on H2OFFT, its key features, and what it can do for you.

1.2. Key Points

H2OFFT-W provides the following key features:

- **Pure WDM Solution:** H2OFFT-W is a pure Windows Driver Model (WDM) utility, requiring IHISI support service in the BIOS layer.
- **Easy To Customize:** H2OFFT-W is based on modularized code including UI, DLL, WDM and OEM layers. The OEM layer provides a convenient way to add hardware dependent codes such as a special ROM write bit controller, EC commands, and high voltage control.
- **Multiple OS Support:** H2OFFT-W has been fully tested with the most popular Microsoft Windows operating systems, such as Windows Vista, Windows 7, Windows 8 and Windows 10. This gives you great flexibility in using H2OFFT-W to support various Windows OS platforms.
- **2MB/4MB/8MB/16MB/32MB Flash Parts:** H2OFFT-W supports update for 2MB, 4MB, 8MB, 16MB and 32MB Flash BIOS.
- **Boot Block Protection:** H2OFFT-W also includes Boot Block Protection. Many systems come with a feature where a “boot block “program is included as part of the BIOS. Boot Block is used to recover corrupted BIOS id in case the system does not boot properly.
- **64-bit OS Support:** H2OFFT-W supports Windows Vista x64, Windows 7 x64, Windows 8 x64 and Windows 10 x64 versions.˙Windows PE Vista support.

1.3. Support Features

H2OFFT-W offers you the following functionality:

- Allows to easily update the system BIOS under Windows environment (Legacy Flash) and secure flash.
- Allows to verify the system BIOS to ensure system reliability.
- Provides you with a friendly Windows graphical user interface (GUI) for maintaining your system BIOS.
- Auto detects hardware settings to help determine H2OFFT-W system compatibility.

1.4. Support Requirements

Installing H2OFFT-W is quick and easy. However, you need to be aware of its system requirements before using the

program:

- Windows Vista/7/8/10 or Windows PE4.0.
- InsydeH2O® BIOS-compatible motherboard (H2OFFT-W will auto detect the BIOS on motherboard for compatibility).
- InsydeH2O® BIOS that supports Insyde IHISI.

2. Using H2OFFT-W

Option	Description
-h	This flash utility help.
-b	Force suspend BitLocker.
-ecp	Flash EC with percentage.
-edt#@:"VALUE"	Update customizes data (such as logo with signature).
-extec:FILE	Update external EC region with input file.
-extrfd PATH	Extract binary file from package.
-g	Save current BIOS to file.
-iv	Show tool and bios support IHISI version.
-mfg	Run in manufacture mode.
-n	Do not reboot after flash.
-noconfirm	Do not show confirm dialog.
-OemCus	Passing specific string to BIOS at program start to do specific action in BIOS.
-pbi:Type(Hex)	Flash BIOS protect region. Type is the protected region type.
-pi	Dump BVDT protection MAP.
-pq	Query ROM protection MAP in current ROM.
-pr	Query external region MAP in current ROM.
-priv	Query ROM private MAP in current ROM.
-pwd:	Input password by command.
-r	Reboot after flash.
-s	Run with silent mode.

2.1. Option /h — Print help

H2OFFT-W -h

All of the tools contain help descriptions to provide guidance in how to use them. This feature will list all options in this tool on screen, allowing users to search through it.

2.2. Option /b — Force suspend BitLocker

H2OFFT-W (filename) -b

When user doing flash on a system which enabled BitLocker, there will pop up a dialog to notify user and allow to suspend BitLocker by tool to continue flash process.

This command will auto suspend BitLocker and don’t pop up the notify dialog.

In silent mode, tool will NOT auto suspend BitLocker on a BitLocker enabled platform, that means it will NOT flash in default. If you want to flash in this case, the -b command must be used.

2.3. Option /ecp — Flash Non-share EC with percentage

H2OFFT-W (filename) -ecp

This command will flash EC with percentage.

2.4. Option /edt — updates customize data

H2OFFT-W (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"

2.5. Option /extec: — Update external EC region

H2OFFT-W –extec:(filename)

Update external EC region with input file.

Note: This command supports on Intel platform only.

2.6. Option /extrfd — Extract binary file from package

H2OFFT_package.exe –extrfd D:\

Extract the binary file in package to specific path.

2.7. Option /g — Save current BIOS to file (from IHISI)

H2OFFT-W (filename) -g

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

2.8. Option /iv — Display IHISI version of tool and platform

H2OFFT-W -iv

This common for querying supported IHISI version of tool and platform.

2.9. Option /mfg — Manufacture mode

H2OFFT-W (filename) -mfg

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

2.10. Do which action after flash

H2OFFT-W (filename) -(r, n)

After flash, the tool will call ec to do the following three things; “reboot”, “shutdown”, and “nothing to do”. The default is to reboot. If this is not your desired behavior, you can use “-n” to prevent system reboot.

2.11. Option /noconfirm — Not show confirm dialog

H2OFFT-W (filename) –noconfirm

Tool will display a confirm dialog to tell user it will do flash, this command can skip this dialog and go ahead.

2.12. Option /OemCus — OEM customize token

H2OFFT-W (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”.

2.13. Option /pbi — Flash BIOS protect region

H2OFFT-W (filename) -pbi:Type(Hex)

You can update partial region base on protecting map. Type the protected region type(Hex).

2.14. Option /pi — Dump BVDT protection MAP (Need BIOS support)

H2OFFT-W (filename) -pi

Tool will dump the BVDT protection map in new BIOS and current BIOS.

2.15. Option /pq — Query ROM protection MAP in current ROM

H2OFFT-W -pq

Dump the protection map of BIOS region in current ROM.

2.16. Option /pr — Query region MAP in current ROM

H2OFFT-W -pr

Dump the region map of BIOS region in current ROM.

2.17. Option /priv — Query ROM private MP in current ROM

H2OFFT-W -priv

Dump the private map of BIOS region in current ROM.

2.18. Option /pwd: — Input password

H2OFFT-W -pwd:PASSWORD

When doing flash on a password required platform, tool will pop up a dialog to request the password. This command can feed the password directly without pop up password dialog.

2.19. Option /s — Silent mode

H2OFFT-W -s

Run flash in silent mode without pop up any dialog.

3. Customizing H2OFFT-W

Prior to shipping H2OFFT-W to your end users, you may need to customize it. H2OFFT-W allows customization of a variety of options.

Supported Configuration Section List

1. [AC_Adapter]
2. [AutoWakeup]
3. [Bios_Version_Check]
4. [BIOSVersionFormat]
5. [CapsuleAudit]
6. [CommonFlash]
7. [FDFile]
8. [FlashComplete]
9. [FlashSecureBIOSOverride]
10. [ForceFlash]
11. [Log_file]
12. [MessageStringTable]
13. [MULTI_FD]
14. [Option]
15. [Others]
16. [PassToBios]
17. [PasswordCheck]
18. [PermitFlashVersion]
19. [Platform_Check]
20. [PlatformVersion]
21. [Region]
22. [ReturnCodeDefinition]
23. [ReturnErrorCode]
24. [SecureUpdate]
25. [UI]
26. [UpdateEC]
27. [UpdateExtraData]
28. [UpdateOEMME]
29. [Version]

3.1. [AC_Adapter]

To do AC/DC check before firmware update.

Flag	**(default) 0:** 1:	Don't check AC. Check AC.
BatteryCheck	**(default) 0:** 1:	Don't check battery. Check battery.
BatteryBound	1~100 **(default) 20**	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	**(default) A02**	A key name which list in [MessageStringTable].
SecurityAcWarning	**(default) A02**	A key name which list in [MessageStringTable].
AcNotConnectError	**(default) A05**	A key name which list in [MessageStringTable].
DcNotConnectError	**(default) A06**	A key name which list in [MessageStringTable].
DcNotEnoughError	**(default) A07**	A key name which list in [MessageStringTable].

3.2. [AutoWakeup]

Supporting auto wake up when flash update is completed and shutdown system.

Flag	**(default) 0:** 1:	Disable. Enable auto power on via RTC.
Interval	**(default) 120**	Integer: Unit is second. Interval time after system turn off.

3.3. [Bios_Version_Check]

To do firmware version check before update firmware.

Flag	0: 1: **(default) 2:**	Don't check rom file version. Check BIOS version. When rom file version is older than BIOS, it will displa a warning message and close application. Depend on BIOS report.
CheckByBios	**(default) 0:** 1:	Normal process. It will pass version by IHISI to BIOS and check by BIOS. When BIOS return not allow to flash, it will be terminated the process. When BIOS allow to flash, it will go ahead and do its normal process. When this flag is enable but BIOS not support, it will skip version check and assume allow to flash.
CheckByBiosErrorMessage		**"This BIOS file is not allow to flash. The flash process will be terminated." (default)** String：User defined error message when BIOS is not allow to flash this version image.

3.4. [BIOSVersionFormat]

The below configuration for firmware version format is used to define version format check.

BIOSVFEnable	**(default) 0:** 1:	Disabled. Enabled.
VersionFormat	[X] [N] [T] [.] [D]	The field is masked. It will NOT be compared. The digit field can be ASCII, case-sensitive. It's the same definition with N. But T is a case-insensitive. Dot is also a mask. It will NOT be compared. 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]

The below configuration for firmware version format is used to define version format check.

Flag	**(default) 0:** 1:	Disabled. Enabled.
QAKey	**(default) empty**	image file name signing by Insyde QA test key.
OemKey	**(default) empty**	image file name signing by OEM/ODM key.

3.6. [CommonFlash]

This feature is only available for specific ODM.

<table>
<tr><td>Flag</td><td>(default) empty</td><td>A switch flag setting string. Ex: "CPVER:[1] ACEN DCEN FHRST"
Detail parameter please reference following table.

<table><tr><th>Parameter</th><th>Description</th></tr><tr><td>PTEN</td><td>All protection enable.</td></tr><tr><td>PTDIS</td><td>All protection disable.</td></tr><tr><td>ACEN</td><td>AC protect checking enable.</td></tr><tr><td>ACDIS</td><td>AC protect checking disable.</td></tr><tr><td>DCEN</td><td>DC & Gangue protect checking enable.</td></tr><tr><td>DCDIS</td><td>DC & Gangue protect checking disable.</td></tr><tr><td>RESSEN</td><td>BIOS Regression enable.</td></tr><tr><td>RESSDIS</td><td>BIOS Regression disable.</td></tr><tr><td>PJMDEN</td><td>Project Model string protect checking enable.</td></tr><tr><td>PJMDDIS</td><td>Project Model string protect checking disable.</td></tr><tr><td>FHOS</td><td>System back to OS after flash BIOS completely.</td></tr><tr><td>FHST</td><td>System directly shutdown after flash BIOS completely.</td></tr><tr><td>FHRST</td><td>System directly reboot after flash BIOS completely.</td></tr><tr><td>CPVER:[Num]</td><td>Common Flash Version information
ex: [Num] is decimal and start from 1.</td></tr></table></td></tr>
<tr><td>ErrorMsg00</td><td>(default) empty</td><td>No error message.</td></tr>
<tr><td>ErrorMsg01</td><td>(default) empty</td><td>AC error message.</td></tr>
<tr><td>ErrorMsg02</td><td>(default) empty</td><td>DC error message.</td></tr>
<tr><td>ErrorMsg03</td><td>(default) empty</td><td>DC gas gauge under xx% message.</td></tr>
<tr><td>ErrorMsg04</td><td>(default) empty</td><td>BIOS version error message.</td></tr>
<tr><td>ErrorMsg05</td><td>(default) empty</td><td>Model name error message.</td></tr>
<tr><td>ErrorMsg10</td><td>(default) empty</td><td>No support this version of Flash Common Interface message.</td></tr>
<tr><td>ErrorMsg##</td><td>(default) empty</td><td>The number ## is in hex.</td></tr>
</table>

3.7. [FDFile]

FileName	**(default)** empty	Utility always load this file. If the FileName is empty, utility will search current directory and load the first found FD file.

3.8. [FlashComplete]

Below configuration is to override default action and to apply optional action in end of flash when launching in normal flash mode.

Action	0: 1: **(default) 2:**	Do nothing. Shutdown. Reboot.
Dialog	**(default) 15:**	Don't display dialog. Display dialog. Display dialog and wait several seconds.
Counter	**(default) 0:** 1:	Unit is second The number of seconds for countdown to reboot or shutdown.
ActionOverride	**(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	**(default) 0:** 1:	Disable to pause after flash complete. Enable to pause after flash complete.
PauseWarning		default：messagestringA03 A key name which list in [MessageStringTable].

3.9. [FlashSecureBIOSOverride]

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

3.10. [ForceFlash]

ALL	**(default) 0:** 1:	Reserve all protected areas. Flash all protected areas.
BB_PEI **CPU_Microcode** **Variable** **DXE** **EC** **Password** **OEM_NVS** **Logo**	**(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	**(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	**(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.

3.11. [Log_file]

Enable it will generate a log file for debugging purposes.

Flag	**(default) 0:** 1:	Don't log to file. Utility will log to specify file.
FileName	**(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	**(default) 0:** 1:	Disable CMOS debug. Enable CMOS debug.
CMOS_INDEX_PORT	**(default) 70:** 72:	Use 0x70 port as index port. Use 0x72 port as index port.
CMOS_DATA_PORT	**(default) 71:** 73:	Use 0x71 port as index port. Use 0x73 port as index port.
CMOS_OFFSET	**(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)

3.12. [MessageStringTable]

Provide message string table to define customize message.

messagestring1 **messagestringA00** **messagestringA01** **messagestringA02** **messagestringA03** **messagestringA04**	**(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.

3.13. [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.

Flag	**(default) 0:** 1:	Disable Enable
FD#XX	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 IO :	
IO,[Offset],	Offset in hex.
[Mask],	IO type supports BYTE, WORD and DWORD in hex.
[Value],	IO type supports BYTE, WORD and DWORD in hex.
[File Name],	File name of FD.
[ME File Name],	ME File name of FD. If it exists, utility will run OEMME flash feature.
[INI File Name]	INI File name for overwrite.

condition of PCI :	
PCI,[Bus],	Bus number
[Device],	Device number
[Function],	Function number
[Offset],	Offset in hex.
[Mask],	PCI type supports DWORD in hex only.
[Value],	PCI type supports DWORD in hex only.
[File Name],	File name of FD.
[ME File Name],	ME File name of FD. If it exists, utility will run OEMME flash feature.
[INI File Name]	INI File name for overwrite.

condition of ID :	
ID,[Model Name],	The platform ID, model name string.
[File Name],	File name of FD.
[ME File Name],	ME File name of FD.
[INI File Name]	INI File name for overwrite.

condition of OS :	
OS,[OS Version],	32bit or 64bit OS. 32 for 32bit OS, 64 for 64bit OS.
[File Name],	File name of FD.
[ME File Name],	ME File name of FD.
[INI File Name]	INI File name for overwrite.

condition of MEMORY :	
MEMORY,[Physical Address],	A DWORD value in hex.
[Mask],	MEMORY type supports BYTE, WORD and DWORD in hex.
[Value],	MEMORY type supports BYTE, WORD and DWORD in hex.
[File Name],	File name of FD.
[ME File Name],	ME File name of FD.
[INI File Name]	INI File name for overwrite.

condition of MPCIIO :	
MPCIIO,[Condition Number],	Condition type number

PCI-[Bus]-[Device]-[Function]-[Offset]-[Mask]-[Value], IO-[Offset]-[Mask]-[Value], [File Name], [ME File Name], [INI File Name]	Condition of PCI Condition of IO File name of FD. ME File name of FD. INI File name for overwrite.

3.14. [Option]

The below setting is to detect whether to automatically or manually start to enter flash process.

Flag	**(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.

3.15. [Others]

The below setting is to detect whether to automatically or manually start to enter flash process.

DisableCompare	**(default) 0:**	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.
	1:	Don't do compare before writing. Just do write action directly.
DisableVerify	**(default) 0:**	Verify BIOS after writing.
	1:	Don't verify BIOS after writing.
VerifyErrorRetry	**(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	**(default) 100:**	Delay time during SMI error retry (millisecond).
SMIErrorRetry	**(default) 5:**	Retry times when SMI fail. **Note: This may let flash time become longer.**
ForceIHISIVersion	**(default) empty:**	This flag will force override the IHISI version which will passing to BIOS. Ex: 1.9.1
EnablePrivateRegionMovement	**(default) 0:**	Disable private region movement.
	1:	Enable private region movement. It will backup original private regions and relocate to new address which defined in new image BVDT when update whole BIOS.
EnableProtectionRegionMovement	**(default) 0:**	Disable protection region movement.
	1:	Enable protection region movement. It will backup original 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	**(default) 0:**	Enable flash secure BIOS on normal platform.
	1:	Disable flash secure BIOS on normal platform.
BootSafeMode	**(default) 0:**	Disable this feature. Normal boot.
	1:	Enable this feature. Next boot into Windows safe mode.

3.16. [PassToBios]

The settings in this section will by pass to BIOS. Tool won't do any action on them.

ClearTXE	**(default) 0:** 1:	Disable. Tell BIOS to clear TXE at this flash.

3.17. [PasswordCheck]

Perform platform name check before firmware update.

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

3.18. [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.

PFVFunctionEnable	**(default) 0:** 1:	Disable conditional version check. Enable conditional version check.
SingleVersion	**(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	**(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.

3.19. [Platform_Check]

Perform platform name check before firmware update.

Flag	0:	Don't check project ID.
	1:	Check project ID of new file. If ID is different with current BIOS, the utility will close.
	2:	Utility will compare current platform ID with the 20 platform IDs. If anyone is match, it will go ahead, otherwise utility will close.
	(default) 3:	Depends on BIOS report.
PlatformName1~20		**(default)** empty String： If ROM file do not contain correct ID, user can define ID here.

3.20. [PlatformVersion]

This flag 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.

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

3.21. [Region]

This section is used to update region of Intel firmware.
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.

DESC	**(default) 0:** 1:	Don't flash. Flash Descriptor region.
GbE	**(default) 0:** 1:	Don't flash. Flash GbE region.
ME	**(default) 0:** 1:	Don't flash. Flash ME region.
EC	**(default) 0:** 1:	Don't flash. Flash EC region.
BIOS	**(default) 0:** 1:	Don't flash. Flash BIOS region.
Platform_Data	**(default) 0:** 1:	Don't flash. Flash Platform Data region.

3.22. [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.

RETURN_SUCCESSFUL	**(default) 0:**	Use input value as return code. 99999: Use the default value.
RETURN_MODEL_CHECK_FAIL	**(default) 259:**	Use input value as return code. 99999: Use the default value.
RETURN_USER_CONFIRM_CANCEL	**(default) 1602:**	Use input value as return code. 99999: Use the default value.
RETURN_AC_NOT_CONNECT	**(default) 1602:**	Use input value as return code. 99999: Use the default value.
RETURN_LOAD_DRIVER_FAIL	**(default) 259:**	Use input value as return code. 99999: Use the default value.
RETURN_NEED_REBOOT	**(default) 3010:**	Use input value as return code. 99999: Use the default value.
RETURN_USER_EXIT	**(default) 1602:**	Use input value as return code. 99999: Use the default value.
RETURN_SAME_VERSION_CHECK	**(default) 1602:**	Use input value as return code. 99999: Use the default value.

3.23. [ReturnErrorCode]

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

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

3.24. [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.

viaESP	**(default) 0:** 1:	Disable. Write the capsule to ESP (EFI system partition).
DeviceOrder	**(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.

3.25. [UI]

Below configuration is provided to user decide what information is present and what action is applied in user interface.

Confirm	**(default) 0:** 1:	Don't display confirm dialog. Display confirm dialog.
Silent	**(default) 0:** 1:	Normal mode. Silent mode, hide main dialog.
SilentWithDialog	0: 1: **(default) 2:**	Don't display any dialog. Display main dialog only. Display all dialogs except main dialog.
DisplayID	**(default) 0:** 1:	Don't display BIOS ID. Display BIOS ID.
InsydeInfo	**(default) 0:** 1:	Don't display Insyde copyright information and URL. Display Insyde copyright information and URL.
VersionInfo	**(default) 0:** 1:	Don't display BIOS version. Display BIOS version.
GroupInfo	**(default) 0:** 1:	Don't display group box. Display group box.
ConfirmInfo	**(default) 0:** 1:	Don't display BIOS version and build date information in confirm dialog. Display BIOS version and build date information in confirm dialog.
ConfirmInfoRDate	**(default) 0:** 1:	Don't display BIOS version and release date information in confirm dialog. Display BIOS version and release date information in confirm dialog. When this flag set to 1, the ConfirmInfo flag won't be referenced.
OnFlashingBeep	**(default) 0:** 1:	Don't beep on flashing. Beep on flashing.
OnFlashingBeepDelayTime	**(default) 800:**	Set beep delay time (milliseconds).
DisableMouseAndKeyboardInput	**(default) 0:** 1:	Don't hook mouse and keyboard. Hook mouse and keyboard without "CTRL+ALT+DEL".
BeforeRunToolDelayTime	**(default) 0:**	Delay number of seconds before flash process.
ProgramStartToWrongMessageBox	**(default) 0:** 1:	Don't pop-up warning dialog before flash process. Pop-up a warning dialog before flash process.
GetFDFileButton	**(default) 0:** 1:	Hide FD file browse button. Show the button for browsing FD file.

DelayBeforeFlash	**(default) 0:**	Delay number of seconds before flashing.
ConfirmDialogCustomizeMessage	**(default) empty:**	A key name which list in [MessageStringTable]. For inserting additional notification message in confirm dialog.
ShowCustomizeMessageOnConfirmDialogOnly	**(default) 0:** 1:	Display default confirm message and insert the customize message on confirm dialog. Do not display default confirm message, only show customize message on confirm dialog.
ShowUpdateROMAddress	**(default) 0:** 1:	Show progress bar when updating BIOS. Show ROM address when updating BIOS.
Elapse	**(default) 0:** 1:	Disable to show elapse time during progressing BIOS update. Enable to show elapse time during progressing BIOS update.
DisableAccessRightWarningMessage	**(default) 0:** 1:	It will show warning message when any region is locked (such as ME) in input image. Do not show warning message when any region is locked in input image.
DisableAccessRightCheck	**default) 0:** 1:	It will check access right when any one region flag of [Region] section is set. It will not do access right check.
PassUpdateProgressToBios	**(default) 0:** 1:	Do not pass update progress to BIOS. Will pass update progress to BIOS during flash.
ShowEcUpdateProgress	**(default) 0:** 1:	Hide the EC update progress during EC updating. Show the EC update progress during EC updating.

3.26. [UpdateEC]

This configuration is configured for EC update.

Flag	**(default) 0:** 1:	Don't flash EC by BIOS. Flash EC by BIOS.
EC_Dialog	**(default) 0:** 1:	Don't display confirm dialog when begin to update EC. Show confirm dialog.
BIOS_Only	**(default) 0:** 1:	Flash EC and BIOS file. Only flash BIOS part of the merge file.
EC_Only	**(default) 0:** 1:	Flash EC and BIOS file. Flash only EC binary file.
EC_Path	**(default) 0:** 1:	EC file name.
EC_Compare	**(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	**(default) 0:** 1:	Don't verify EC. Verify EC after writing.
EC_VerifyErrorRetry	**(default) 3:**	Retry times. If the value is not zero means enable verify retry, and will retry set times. 0 for disable verify retry.
EC_BlockSize	**(default) 64:**	The block size of EC update. Unit is KB.

3.27. [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).

MEFileName		empty (default) If this file name or Multi-FD ME file name exist tool will run this case to flash ME.
Command		empty (default) 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.

3.28. [Version]

Version	**(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. Using iFDPacker

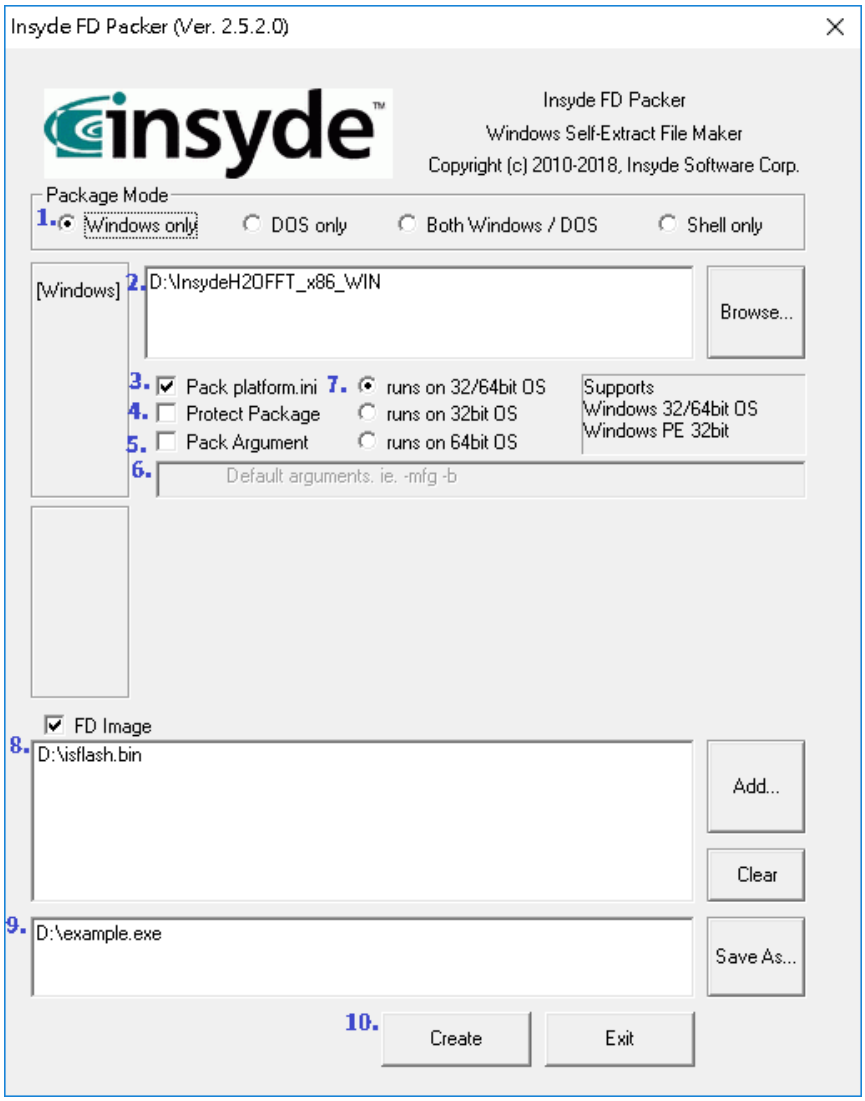
4.1. Pack H2OFFT

4.1.1 Pack H2OFFT steps

Locate “iFdPacker.exe” in the Packer folder. You can use this tool to package H2OFFT-W and binary files with the setting in Platform.ini.

In addition to the H2OFFT-W utility, BIOS ROM files, etc., the “iFdPacker.exe” is the InsydeFdPacker utility for you to generate an executable file.

Additionally, you can use InsydeFdPacker to pack the entire H2OFFT-W folder into an executable file. To use InsydeFdPacker, follow these procedures:



1. Select Package mode item.
2. Click [Browse] to select directory of H2OFFT-W package.
3. Checked the check box if using external configure file.

4. Checked the check box if want to protect the package. A protected package will not allow to view, extract and modify via 3-party program such as 7-zip.

5. Checked the check box to input the default argument.

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

7. Select windows mode support OS.

- runs on 32/64bits supports
  Windows 32/64bit OS
  Windows PE 32bit
- runs on 32bits supports
  Windows 32bit OS
  Windows PE 32bit
- runs on 64bits supports
  Windows 64bit OS
  Windows PE 64bit

8. Click [Add] to select BIOS image.

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

10. Click [Create] to pack.

4.1.2 Packer support command list

You can use "iFdPacker -h" to see the usage

Command	Description
-winsrc PATH	The path of WinFlash
-dossrc PATH	The path of DosFlash
-shlsrc PATH	The path of ShellFlash
-b [3264 | 32 | 64]	The WinFlah Build Type 3264 - 32bit Ap runs on 32/64bit OS 32 - 32bit Ap runs on 32bit OS 64 - 64bit Ap runs on 64bit OS
-winini	Windows pack platform.ini
-dosini	Dos pack platform.ini
-shlini	Shell pack platform.ini
-protect	Protect packed package that will not allow to unzip or modify with third-party program.
-winarg “flag”	Windows argument with quotation marks
-arg “flag"	Dos/Shell argument with quotation marks
-argfilter "flag"	Dos/Shell argument filter with quotation marks
-fv FILE	The path of firmware file
-output FILE	The single package file
-h	The usage message

Example:

a. Pack Windows Only:

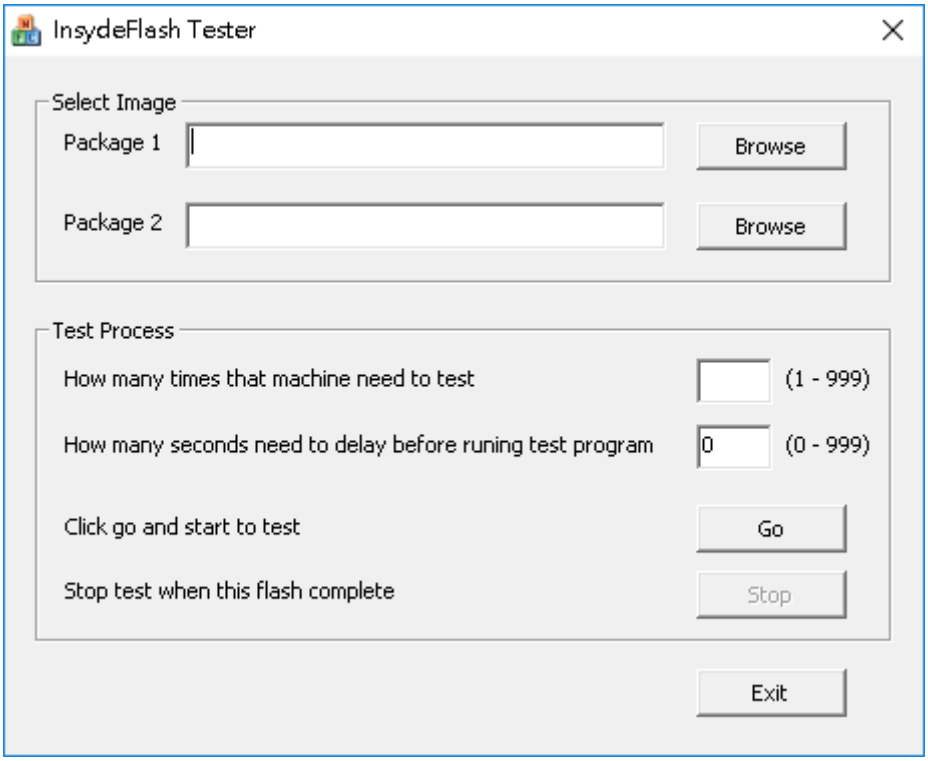
```
iFdPacker.exe  -winsrc D:\InsydeH2OFFT_x86_WIN
               -winini
               -b 3264
               -fv d:\file1.fv
               -fv d:\file2.fd
               -output d:\output.exe
```

b. Pack both Windows and Dos:

```
iFdPacker.exe  -winsrc D:\InsydeH2OFFT_x86_WIN
               -winini
               -b 3264
               -dossrc D:\InsydeH2OFFT_x86_DOS
               -dosini
               -fv d:\file1.fv
               -arg "-bios"
               -output d:\output.exe
```

5. Flash Tester

There is the Flash Tester in the tools folder, which provides a long run test feature:



1. This tool will execute the execution file which packed via IFdPacker.exe.
2. Package 1 sets the package file that you want to test.
3. Package 2 sets an additional package file to test.
4. The delay time in second can wait until time is up before execute the package.
4. Enter the number of tests you would like to run.
5. Click “GO” to start this tool.

The “Stop” button allows you to stop the extended test.
The “Exit” button is to close this dialog.

Note:

- Flash tester needs to be run as Administrator account in Windows 8.
- Run "Net user administrator /active:yes" in cmd then use netplwiz utility to set Administrator account automatic logon for Windows 8.
- The file name in Package1 and Package2 can be the same. Both of them are necessary. You cannot enter one file and run the test.

6. Support for BIOS Guard (PFAT) image update

6.1. How to sign a BIOS Guard (PFAT) BIOS image?

Support for PFAT image update requires BIOS version: SharkBay 03.72.37.0018.

1. System Requirements:
   - Microsoft Windows 7 or later.
   - Microsoft "SignTool.exe". (Included in the Microsoft Windows SDK package. v6.2.8229.0 in Win8 SDK or later).
   - Make sure the SignTool.exe is in the System Environment Variable "path".
2. Please install QA Certificate in your system. (Reference QA Certificate Installation Guide) You need QA.pfx file and double-click to install it into your system.
3. Use iEFIFlashSigner.exe to sign PFAT BIOS image in command mode. (Output file name: isflash.bin)

Example:

Sign PFAT bios image only command:

iEFIFlashSigner.exe -n "QA Certificate." -bios BIOS_PFAT.fd

(PS: BIOS_PFAT.fd is PFAT BIOS image file name)

7. FAQ

7.1 Which configure file (platform.ini) will be referenced if the configure file built in capsule image and also exist external folder?

External configure file is higher priority,

- **In secure update**
  H2OFFT will reference the platform.ini in the same folder instead of configuration of secure FD (isflash.bin).
- **In package mode**
  H2OFFT will reference the external configure file, if click “pack platform.ini”.
  If there is platform.ini in the same folder with package file (package01.exe), configure file will not be referenced.

7.2 When AC/DC status is correctly reported via IHISI 10h, why H2OFFT still indicate AC/DC status is in correct?

In the Windows version, H2OFFT gets AC/DC power status via Windows API not via IHISI 10h. In DOS/Shell and WinPE is reference the power status from IHISI 10h.

7.3 When complete to configure for update ME via [UpdateOEMME] section, why Intel ME firmware tool still show ME update is failed?

The Intel firmware update tool (FWUpdLcl.exe) is dependent on each chipset generation, the tool in release flash package is a sample and may be not suitable for your project. So please remember to replace the FWUpdLcl.exe with right version before you will utilize function to update Intel firmware (ME or TXE).

Other, please also make sure ME=1 in [Region] section if you would update ME firmware region.