H2OFFT™ (Flash Firmware Tool)

User Guide for EFI version

Rev. 1.2l Approved
Prepared by SI-Driver Team
Apr 30, 2014
Insyde Software Corp.

Disclaimer

Insyde Software Corp. provides this document and the associated 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	Author	Approver	Summary
1.1o	05/25/2012	Kevin Liang	<Approver>	Initial version
1.2d	08/10/2012	Kevin Liang	<Approver>	Update command & add platform.ini & ifdpacker
1.2g	09/26/2012	Kevin Liang	<Approver>	Update command & add platform.ini & ifdpacker
1.2h	10/16/2012	Kevin Liang	<Approver>	Update command & ifdpacker & support PFAT image update
1.2i	01/15/2013	Kevin Liang	<Approver>	Update command & platform.ini & ifdpacker version
1.2k	02/20/2013	Kevin Liang	<Approver>	Update ifdpacker version
1.2l	05/09/2014	Jerry Yen		Update commands

Table of Contents

1. Introduction

1.1. Overview

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

1.2. What H2OFFT (for EFI) can do

H2OFFT offers you the following flexibility and ease:

1. Allows you to save and make a backup file of the current system BIOS before updating with a new one.
2. Allows you to easily update the system BIOS under EFI Shell environment.
3. Allows you to verify the system BIOS to ensure system reliability.
4. Auto detects hardware settings so you will know if H2OFFT is compatible with your system or not. This gives enhanced security.

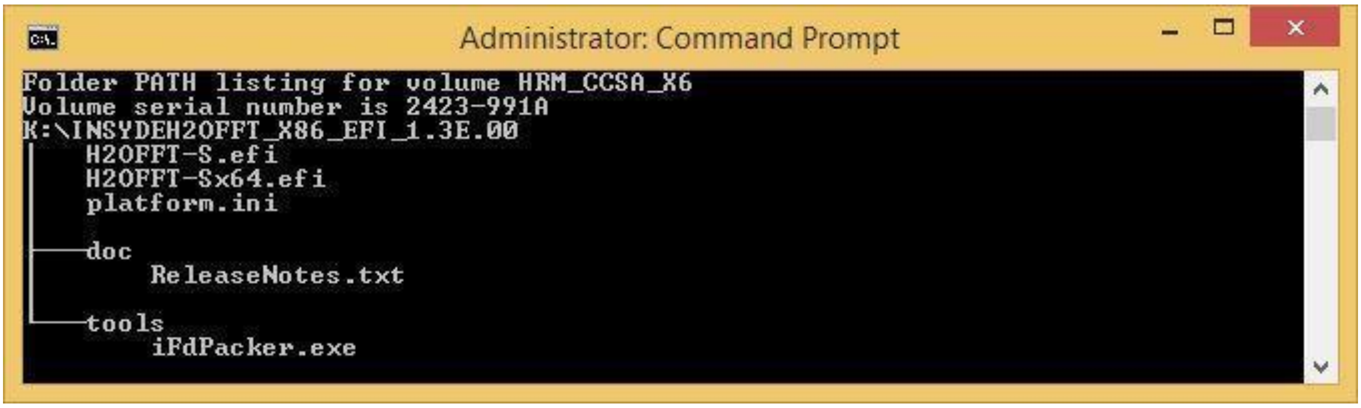
2. Installing H2OFFT-S

2.1. System requirements

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

- EFI shell environment
- Any Insyde BIOS-compatible motherboard (H2OFFT-S will auto detect the BIOS on motherboard if compatible)
- InsydeH2O BIOS with IHISI
- On secure boot supported system, the shell flash utility (H2OFFT-Sx64.efi) must be signed via iEFIFlashSigner.exe. Please refer “ReadMe.txt” in security flash package to know how to sign by iEFIFlashSigner.exe.

2.2. Member



Administrator: Command Prompt

```
Folder PATH listing for volume HRM_CCSA_X6
Volume serial number is 2423-991A
K:\INSYDEH2OFFT_X86_EFI_1.3E.00
│   H2OFFT-S.efi
│   H2OFFT-Sx64.efi
│   platform.ini
│
├───doc
│       ReleaseNotes.txt
│
└───tools
        iFdPacker.exe
```

Figure 2-1. H2OFFT-S member

This picture shows all of the members in this tool. Below is a quick overview about them:

<table>
<tr><td>K:\INSYDEH2OFFT_X86_EFI_1.3E.00</td><td></td></tr>
<tr><td>| H2OFFT-S.efi</td><td>– The main execution file that supports x86 systems</td></tr>
<tr><td>| H2OFFT-Sx64.efi</td><td>– The main execution file that supports x64 systems.</td></tr>
<tr><td>| platform.ini</td><td>– Configuration file</td></tr>
<tr><td>|</td><td></td></tr>
<tr><td>+---doc</td><td></td></tr>
<tr><td>| ReleaseNotes.txt</td><td>– Version release information</td></tr>
<tr><td>|</td><td></td></tr>
<tr><td>+---tools</td><td></td></tr>
<tr><td>iFdPacker.exe</td><td>– Utility to generate an executable file.</td></tr>
</table>

3. Flash usage

H2OFFT-S is a tool that allows access to the BIOS rom. It supports many features in this tool by (-option):

Option	Description
-h	This Flash utility help
-ab	Check battery life percent (only use when AC not plugged-in and with -AC)
-ac	Do not check ac power plug in
-acb	Check AC plug-in & battery percentage
-all	Flash all
-b	Flash PEI volume
-bios	Flash BIOS region
-dc	Disable comparison in normal flash process
-desc	Flash DESC region
-di	Disable ID display
-e:Offset,Size,Address	Update fix size from file offset to physical address.
-ecbp	Flash EC with BIOS with percentage
-ecp	Flash EC with percentage
-edt#@:"value"	updates customize data by IHISI
-fd	Flash DXE
-fe	Flash EC
-fl	Flash Logo
-fm	Flash CPU Microcode
-fn	Flash OEM NVS
-fp	Flash password
-ft:Type	Flash OEM special type
-fv	Flash Variable
-g	Save current BIOS to file (from IHISI)
-gbe	Flash GBE region
-mc	Skip all platform model check
-me	Flash ME region
-mfg	run in manufacturing mode
-n	Do not reboot after flash
-pbi:Type	Flash BIOS protect region.
-pdr	Flash Platform Data region
-pq	Query ROM protection MAP in current ROM
-pr	Query region MAP in current ROM
-r	Reboot after flash
-rt:Times,Delay	When SMI error, retry how many times and delay how many milliseconds between each retry attempt
-s	Shutdown after flash
-u	Show confirm message
-vrt:Times	When verify error, retry how many times

3.1. Option /h — Print help

H2OFFT-S.efi -h

All tools must let the user know how to use them. This feature will list all options in this tool on screen and the user can search for them. It will also check if it uses the OEM help list or AP help list.

3.2. Option /ab — Check battery life percent

H2OFFT-S.efi (filename) -ac -ab

This feature uses the option “ac” to skip AC check and the AC not plug in (Only use when AC is not plugged in and with /ac). This feature will check the battery percentage. It must be more than the value set in BIOS.

3.3. Option /ac — Do not check ac power plug in

H2OFFT-S.efi (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 the user. If you do not want to check it, please use this feature.

3.4. Option /acb — Check AC plug-in & battery percentage

H2OFFT-S.efi (filename) –acb

Before flashing BIOS, the tool will check AC plug-in & battery percentage.

3.5. Option /all — Flash all

H2OFFT-S.efi (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 regions.

3.6. Flash protecting BIOS area

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

H2OFFT-S.efi (filename) /b	Force to update the PEI area.
/fd	Force to update the DXE area.
/fe	Force to update the EC area.
/fl	Force to update the Logo area.
/fm	Force to update the CPU Microcode area.
/fp	Force to update the password area.
/fn	Force to update the OEM NVS area.
/ft	Force to update the special type area.
/fv	Force to update the variable area.

3.7. Flash region that describe in descriptor table

H2OFFT-S.efi (filename) -(desc, me, gbe, bios)

In descriptor mode there are five regions of the BIOS rom. Those commands will allow you to flash the region that you want to flash.

3.8. Option /dc — Disable comparison in normal flash process

H2OFFT-S.efi (filename) -dc

During flash BIOS tool will read the current block from BIOS and new block from file, and then compare them. If they are the same, tool will skip and not flash those blocks. This command will skip this compare feature.

3.9. Option /di — Disable ID display

H2OFFT-S.efi (filename) -di

Tool will show the name of this platform, and this feature will disable the name display.

3.10. Option /e — Update fix size from file offset to physical address

H2OFFT-S.efi (filename) –e:Offset(Hex),Size(Hex),Address(Hex)

This command will update fix size from file offset to physical address.

3.11. Option /ecp — Flash EC with percentage (by applicaion)

H2OFFT-S.efi (filename) -ecp

This command will flash EC with percentage (by application).

3.12. Option /edt — updates customize data by IHISI

H2OFFT-S.efi (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"

3.13. Option /ecbp — Flash EC with BIOS (by IHISI) and show percentage

```
H2OFFT-S.efi (filename) -ecbp
```

Flash non-share rom EC with new BIOS file. So “filename” is the EC and the BIOS merge filename.

Because this feature sends SMI many times, this EC is not flash by SMI one time. To learn more about this SMI, please read IHISI 1.8.0, or the higher version of IHISI spec.

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

```
H2OFFT-S.efi (filename) -g
```

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

3.15. Option /gbe — Flash GBE region

```
H2OFFT-S.efi –gbe:(GUID)
```

This command can to flash GBE region.

3.16. Option /mc — Skip all platform model check

```
H2OFFT-S.efi (filename) -mc
```

This tool will check the platform model for the version and the platform name reported from BIOS. If these do not match, the tool will show an error message and leave. If you don't want to use this feature, you can use this command to do it. As a result, the tool will not show the name and version.

3.17. Option /me — Flash ME region

```
H2OFFT-S.efi (filename) -me
```

This command can to flash ME region.

3.18. Option /mfg — run in manufacturing mode

```
H2OFFT-S.efi (filename) -mfg
```

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

3.19. Do what thing after flash

```
H2OFFT-S.efi (filename) -(s, n)
```

After flashing BIOS, tool will call ec to do the three things, “reboot”, “shutdown”, “nothing to do.” The default is to reboot. But sometimes you may not want to do this. In this situation, you can use “-n” to prevent system reboot or use “-s” to tell the tool to shutdown after flash.

3.20. Option /pbi — Flash BIOS protect region.

```
H2OFFT-S.efi (filename) -pbi:Type
```

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

3.21. Option /pdr — Flash Platform Data region.

```
H2OFFT-S.efi (filename) -pdr
```

Application will flash Platform Data region.

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

```
H2OFFT-S.efi -pq
```

Dump the protection map of BIOS region in current rom.

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

```
H2OFFT-S.efi -pr
```

Dump the region map of BIOS region in the current rom.

3.24. Option /r — Reboot after flash

```
H2OFFT-S.efi (filename)-r
```

Application will reboot after flash.

3.25. Option /rt — When SMI error, retry how many times

```
H2OFFT-S.efi -rt:Times,Delay
```

When SMI error occurs, this lets you set the number of retry attempts and the delay in milliseconds between each retry attempt.

3.26. Option /u — Show confirm message

```
H2OFFT-S.efi (filename) -u
```

Application will ask to confirm if you want to flash before flashing.

3.27. Option /vrt — When verifying error, retry how many times

```
H2OFFT-S.efi -vrt:Times
```

When verifying error, retry how many times.

4. Customizing H2OFFT-S

Supported sections

1. [AC_Adapter]
2. [Bios_Version_Check]
3. [BIOSVersionFormat]
4. [CommonFlash]
5. [FlashComplete]
6. [FlashSecureBIOSOverride]
7. [ForceFlash]
8. [Log_file]
9. [MessageStringTable]
10. [MULTI_FD]
11. [Others]
12. [Platform_Check]
13. [PlatformVersion]
14. [Region]
15. [SecureUpdate]
16. [UI]
17. [UpdateEC]
18. [UpdateOEMME]

4.1. [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**	String：A key name which list in [MessageStringTable].
SecurityAcWarning	**(default) A02**	String：A key name which list in [MessageStringTable].

4.2.[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 display 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.

4.3. [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 ignored in version compare.

4.4. [CommonFlash]

The function is only available in specific ODM.

Flag	**(default)** empty	A switch flag setting string. Ex: "CPVER:[1] ACEN DCEN FHRST" Detail parameter please reference following 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 FHRST completely. CPVER:[Num] System directly reboot after flash BIOS completely. Common Flash Version information ex: [Num] is decimal and start from 1.
ErrorMsg00	**(default)** empty	No error message.
ErrorMsg01	**(default)** empty	AC error message.
ErrorMsg02	**(default)** empty	DC error message.
ErrorMsg03	**(default)** empty	DC gas gauge under xx% message.
ErrorMsg04	**(default)** empty	BIOS version error message.
ErrorMsg05	**(default)** empty	Model name error message.
ErrorMsg10	**(default)** empty	No support this version of Flash Common Interface message.
ErrorMsg##	**(default)** empty	The number ## is in hex.

4.5. [FlashComplete]

The 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) 0**： 1： 2：	Don't display dialog. Display dialog. Display dialog and wait several seconds.
Counter	**(default) 15**	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].

4.6. [FlashSecureBIOSOverride]

The below configuration is to override default action when launching in secure flash mode.

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.

4.7. [ForceFlash]

Decide what flash region will be forced to update base on below configuration if the some region BIOS report had been protected.

ALL	**(default) 0 :** 1 :	Reserve all protected areas. Flash all ROM parts.
BB_PEI **CPU_Microcode** **Variable** **DXE** **EC** **Password** **OEM_NVS** **Logo** **Type#09** **Type#08**	**(default) 0 :** 1 :	Protect these areas if BIOS report them are protected areas. Force flash these areas if BIOS report them are protected areas.

4.8. [Log_file]

For debugging purposes, we can generate a log file and output error code via CMOS.

Flag	**(default) 0**： 1：	Don't log to file. Utility will log to specify file.
FileName		H2OFFT-W.log (default) String：Log file name.

4.9. [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.

4.10. [MULTI_FD]

Some of the flash packages can contain multiple firmware image for different SKU. The following settings can be configured for the condition to detect the firmware image.

Flag	**(default) 0** : 1 :	disable enable
FD#XX	FD#01 ~ FD#99	XX is decimal number from 01 to 99. String : 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 MPCIIO :	
MPCIIO,[Condition Number],	: Condition type numbrer
PCI-[Bus]-[Device]-[Function]-[Offset]-[Mask]-[Value],	: Condition of PCI
IO-[Offset]-[Mask]-[Value],	: Condition of IO
[File Name],	: File name of FD.
[ME File Name],	: ME File name of FD.

[INI File Name]	: INI File name for overwrite.

4.11. [Others]

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

DisableCompare	**(default) 0 :** 1 : 2 :	Read BIOS and compare difference before writing. If the readed 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	**(default) 0 :** 1 :	Verify BIOS after writing. Don't verify BIOS after writing.
VerifyErrorRetry	**(default) 3**	Retry times. If the value is not zero means enable verify retry, and will retry setted times. 0 for disable verify retry.
SMIErrorRetryDelay	**(default) 100**	Unit is millisecond Delay time during SMI error retry (millisecond).
SMIErrorRetry	**(default) 5**	Retry times when SMI fail. (Note: This may let flash time become longer.)
DisableSecureCapsuleFlash	**(default) 0 :** 1 :	Enable flash secure BIOS on normal platform. Disable flash secure BIOS on normal platform.

4.12. [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.

4.13. [PlatformVersion]

This flag is only available when the [Platform_Check] is enabled to compare the 20 platform IDs.
The Version is paired 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.

4.14. [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.

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

4.15. [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.

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

4.16. [UI]

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

Elapse	**(default) 0**： 1：	Disable to show elapse time during progressing BIOS update. Enable to show elapse time during progressing BIOS update.

4.17. [UpdateEC]

This configuration is configured for EC update.

EC_DockWarning		messagestringA04 (default) A key name which list in [MessageStringTable].

4.18. [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.
CheckVersion	**(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.
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.

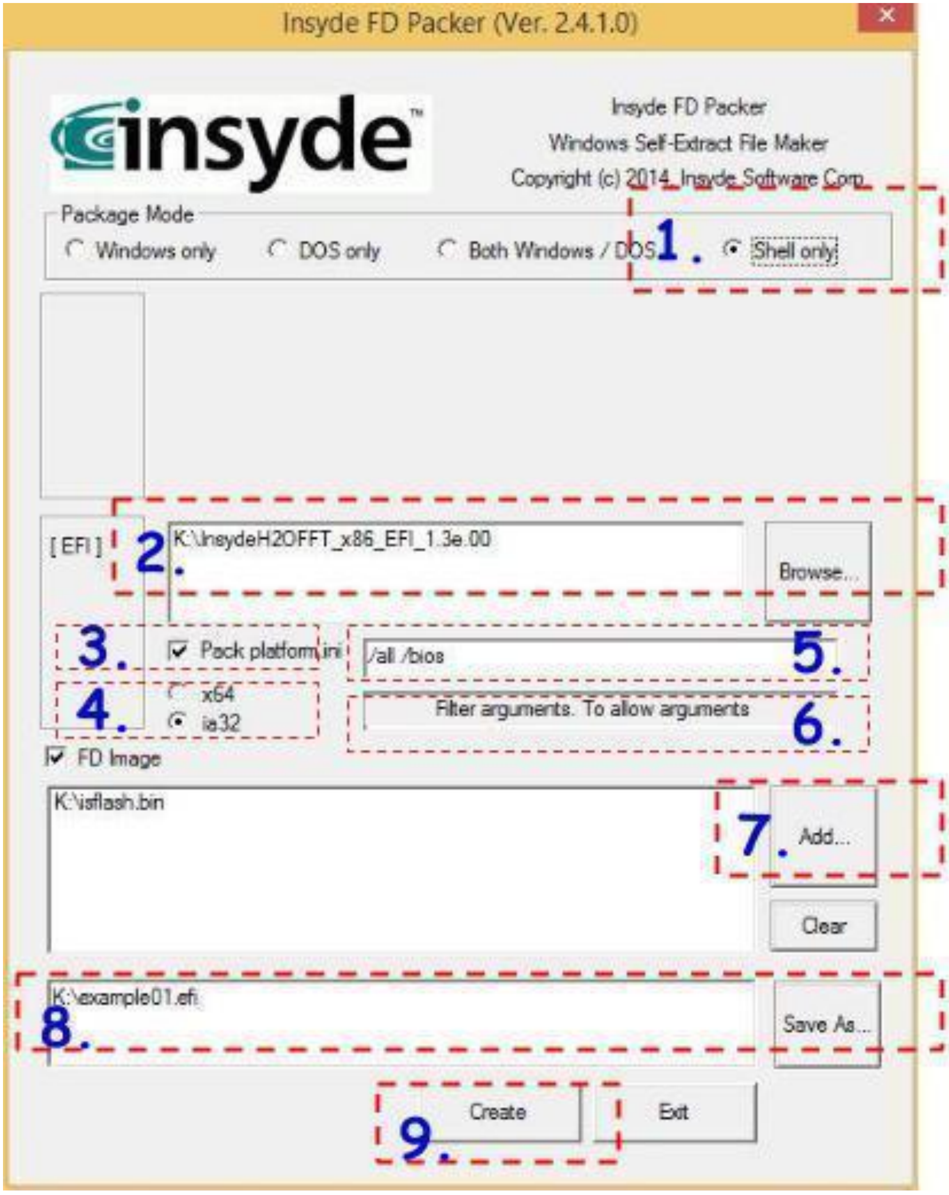
5. Using iFDPacker

5.1. Package H2OFFT-S

Locate "iFdPacker.exe" in the tool folder. You can use this tool to package H2OFFT-S and binary files with the setting in Platform.ini.

In addition to the H2OFFT-S 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-S folder into an executable file. To use InsydeFdPacker, follow these procedures:

A. Run package H2OFFT-S step (For Shell setting):


Insyde FD Packer (Ver. 2.4.1.0)

Insyde FD Packer
Windows Self-Extract File Maker
Copyright (c) 2014 Insyde Software Corp.

Package Mode: ○ Windows only ○ DOS only ○ Both Windows / DOS ● Shell only

[EFI] K:\InsydeH2OFFT_x86_EFI_1.3e.00

☑ Pack platform.ini   /all /bios

○ x64
● ia32

Filter arguments. To allow arguments

☑ FD Image

K:\isflash.bin

K:\example01.efi

1. Select Package mode item

2. Click "Browse" to select directory of H2OFFT-S package.

3. Select if using external configure file.

4. Select EFI mode.

5. Select default argument

6. Select filter argument.

7. Click [Add] to select BIOS image.

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

9. Click [Create] to pack.

B. Support run command line

1. Command Description:

	Command	Description
1	-winsrc PATH	The path of WinFlash
2	-dossrc PATH	The path of DosFlash
3	-shlsrc PATH	The path of ShellFlash
4	-b [3264 | 32 | 64]	The WinFlah/EFI Build Type 3264 - 32bit Ap runs on 32/64bit OS 32 - 32bit Ap runs on 32bit OS 64 - 64bit Ap runs on 64bit OS
5	-winini	Windows pack platform.ini
6	-dosini	Dos pack platform.ini
7	-shlini	Shell pack platform.ini
8	-arg “flag"	Dos/Shell argument with quotation marks
9	-argfilter "flag"	Dos/Shell argument filter with quotation marks
10	-fv FILE	The path of firmware file
11	-output FILE	The single package file
12	-h	The usage message

2. Example:

```
iFdPacker.exe   -shlsrc K:\InsydeH2OFFT_x86_EFI_1.3e.00
                -shlini
                -b 64
                -fv d:\isflash.bin
                -arg “-bios -all”
                -argfilter “-ac”
                -output d:\example01.efi
```

6. *Support for PFAT image update*

6.1. How to sign a PFAT BIOS image?

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

1. System Requirements:
   a. Microsoft Windows 7 or later.
   b. Microsoft "SignTool.exe". (Included in the Microsoft Windows SDK package. v6.1.7600.16385 or later).
   c. 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: H2OFFT-S.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)

6.2. How to update signed PFAT BIOS image via flash utility?

Please put H2OFFT-S.bin file into H2OFFT-S folder and run H2OFFT-S command to update.

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 platmform.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.efi), configure file will not be referenced.