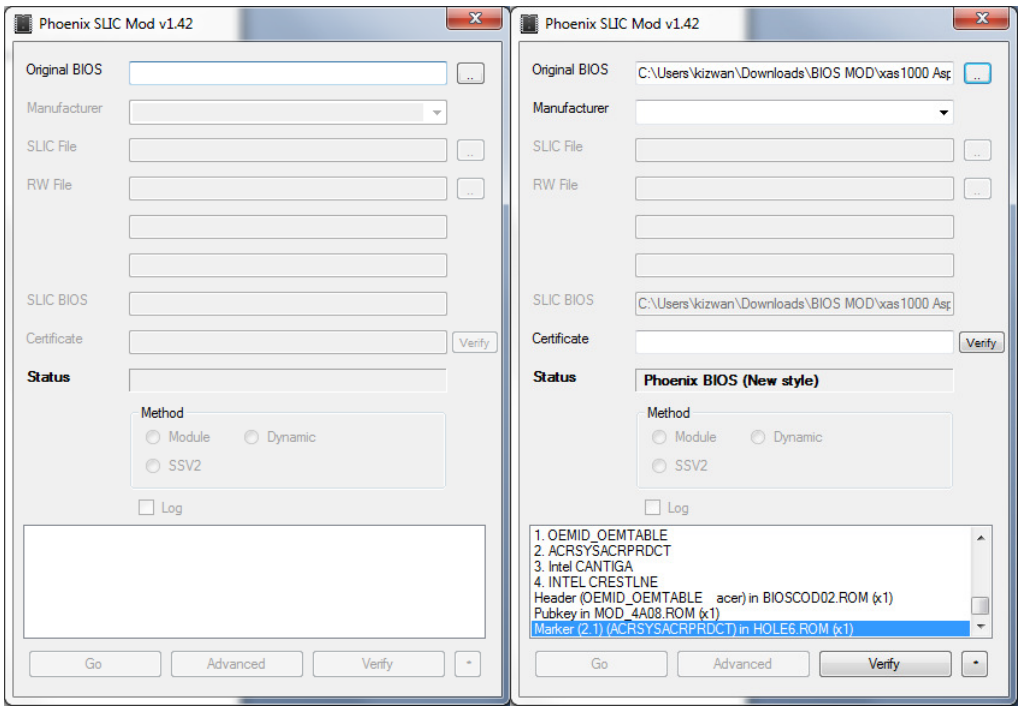
NOTE:-

- PBE = Phoenix BIOS Editor

1) Create a working directory, let say: C:\BIOS_MOD

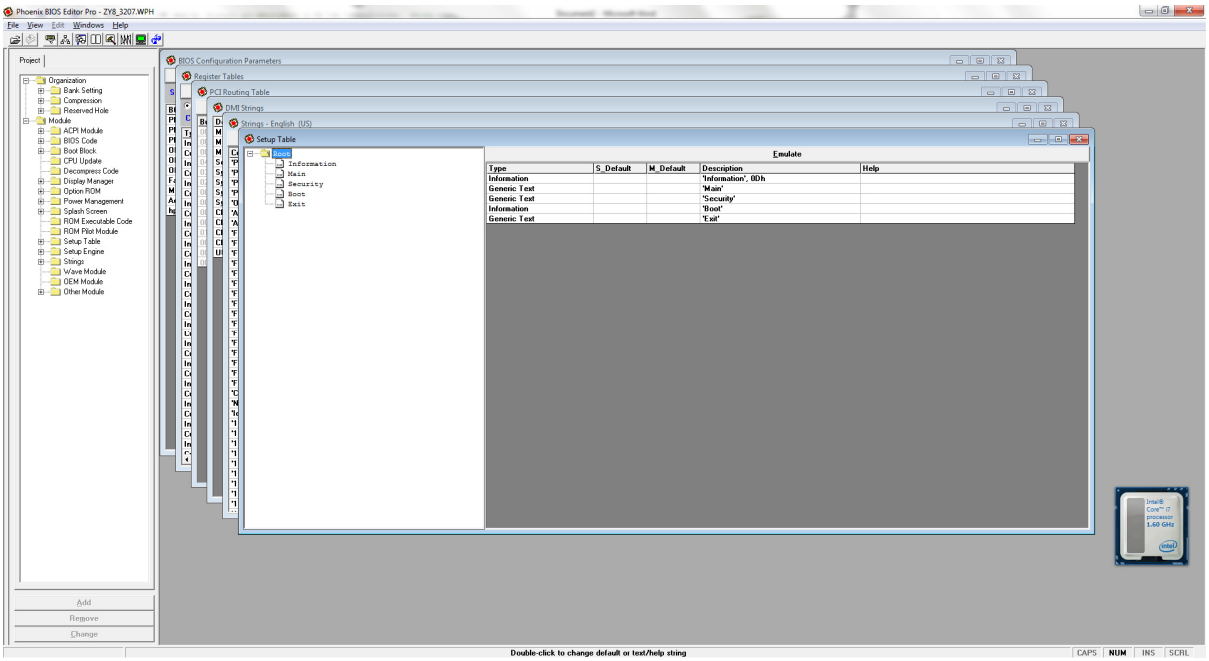
2) Download & extract the BIOS image (.ROM or .WPH) from the compress file or EXE file (BIOS self-update software) to working directory.

3) Decompress the BIOS image (.ROM or .WPH file) using Phoenix SLIC Tool/Mod (current version at time of writing this tutorial is 1.42):-



- All extracted files will be located in **DUMP** folder.
- The extracted files are useful if PBE produced error(s) while re-building the modified BIOS image.
- PBE have known issue where it (sometime) unable to extract the BIOS image properly & this will lead to bad BIOS image after re-build.

4) Open the BIOS image with PBE:-



5) Go to PBE's **TEMP** folder:-
   - 64bit OS: C:\Program Files (x86)\Phoenix Technologies Ltd\BIOS Editor\TEMP
   - 32bit OS: C:\Program Files\Phoenix Technologies Ltd\BIOS Editor\TEMP
6) Copy **OLD1.RLS**, **STRINGS0.ROM** (or **STRINGS00.ROM**) & **TEMPLAT0.ROM** (or **TEMPLAT00.ROM**) files from PBE's **TEMP** folder to working directory.
7) Open **OLD1.RLS** with HEX editor & copy the first two bytes:-

OLD1.RLS

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	D6	70	52	03	75	73	14	00	74	51	D4	07	E3	07	0C	08	ÖpR.us..tQÔ.ã...
00000010	1B	08	44	08	5F	08	6E	08	7D	08	8C	08	9B	08	AB	08	..D._.n.}.Œ.›.«.
00000020	BB	08	CB	08	DB	08	EB	08	FB	08	0A	09	19	09	28	09	»Ë.Û.ë.û.....(.
00000030	37	09	46	09	55	09	64	09	73	09	7A	09	7F	09	8A	09	7.F.U.d.s.z...Š.
00000040	8F	09	94	09	99	09	9E	09	A3	09	A8	09	AD	09	B2	09	..".™.ž.£.¨...ª.
00000050	B7	09	BC	09	C1	09	C6	09	CB	09	D0	09	D5	09	DA	09	·.¼.Á.Æ.Ë.Ð.Õ.Ú.
00000060	DF	09	E4	09	E9	09	EE	09	F3	09	F8	09	FD	09	02	0A	ß.ä.é.î.ó.ø.ý...
00000070	07	0A	0C	0A	11	0A	16	0A	1B	0A	20	0A	25	0A	2A	0A	..... .%.*.
00000080	2F	0A	34	0A	39	0A	3E	0A	43	0A	48	0A	4D	0A	52	0A	/.4.9.>.C.H.M.R.
00000090	57	0A	5C	0A	61	0A	66	0A	6B	0A	70	0A	75	0A	7A	0A	W.\.a.f.k.p.u.z.
000000A0	7F	0A	84	0A	89	0A	8E	0A	93	0A	98	0A	9D	0A	A2	0A	..„.‰.Ž.".˜...¢.
000000B0	A7	0A	AC	0A	B1	0A	B6	0A	BB	0A	C0	0A	C5	0A	CA	0A	§.¬.±.¶.».À.Å.Ê.
000000C0	CF	0A	D4	0A	D9	0A	DE	0A	E3	0A	E8	0A	ED	0A	F2	0A	Ï.Ô.Ù.Þ.ã.è.í.ò.
000000D0	F7	0A	FC	0A	01	0B	06	0B	0A	0B	0F	0B	13	0B	17	0B	÷.ü..............

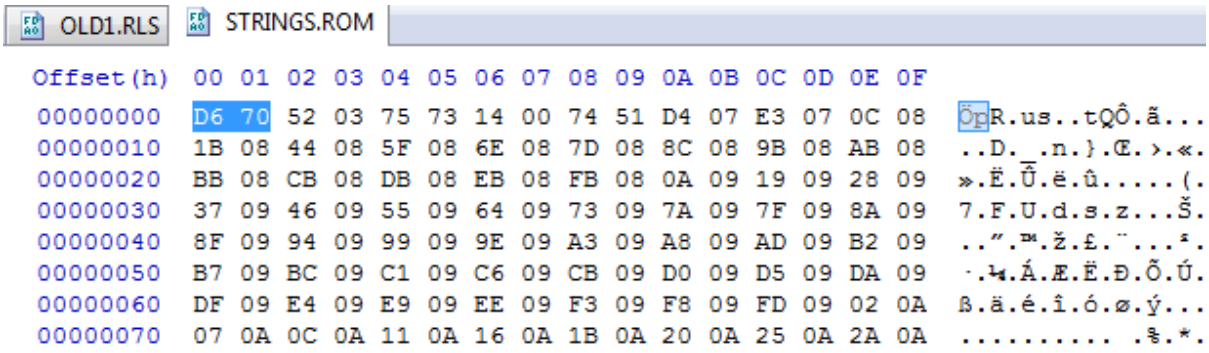
8) Open **STRINGS0.ROM** with HEX editor & search for the two bytes, copied from the **OLD1.RLS** file above. In this example, the two bytes are **D6 70**:-

OLD1.RLS STRINGS00.ROM

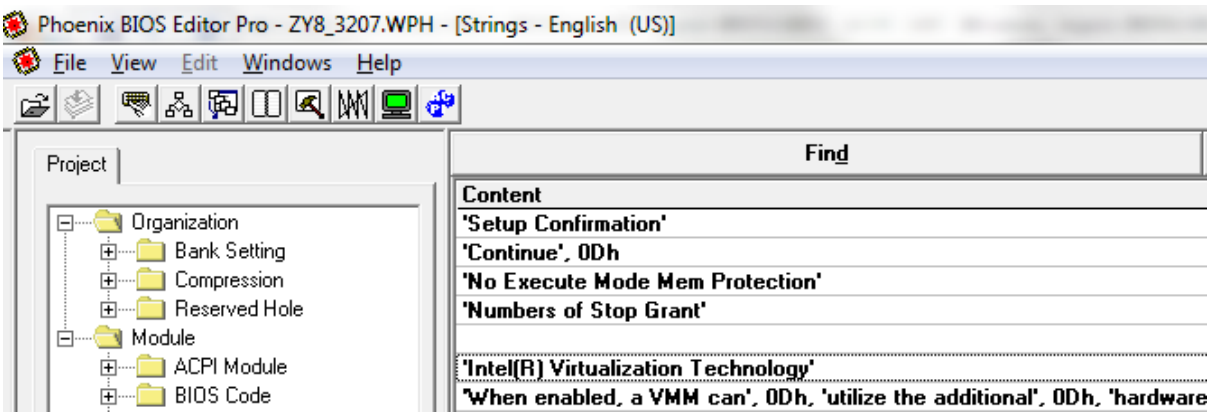
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000000	53	54	52	50	41	43	4B	2D	42	49	4F	53	00	00	00	00	STRPACK-BIOS....
00000010	00	00	00	00	00	00	00	00	01	00	02	00	D6	70	52	03	..............ÖpR.
00000020	75	73	14	00	74	51	D4	07	E3	07	0C	08	1B	08	44	08	us..tQÔ.ã.....D.
00000030	5F	08	6E	08	7D	08	8C	08	9B	08	AB	08	BB	08	CB	08	_.n.}.Œ.›.«.».Ë.
00000040	DB	08	EB	08	FB	08	0A	09	19	09	28	09	37	09	46	09	Û.ë.û.....(.7.F.
00000050	55	09	64	09	73	09	7A	09	7F	09	8A	09	8F	09	94	09	U.d.s.z...Š...".

- The **01 00 02 00** means there is one language in this file which is EN (US).

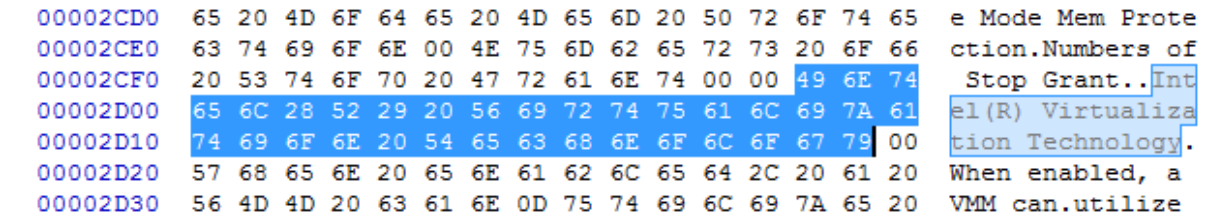
9) All of the bytes before **D6 70** are called header. The length of the header should be 0x1C. Delete the header & save it as **STRINGS.ROM**:-



10) Search for “Intel(R) Virtualization Technology” or “VT Feature” or just “Virtualization” in the **STRINGS** window in PBE. In this example, we will found “Intel(R) Virtualization Technology”:-

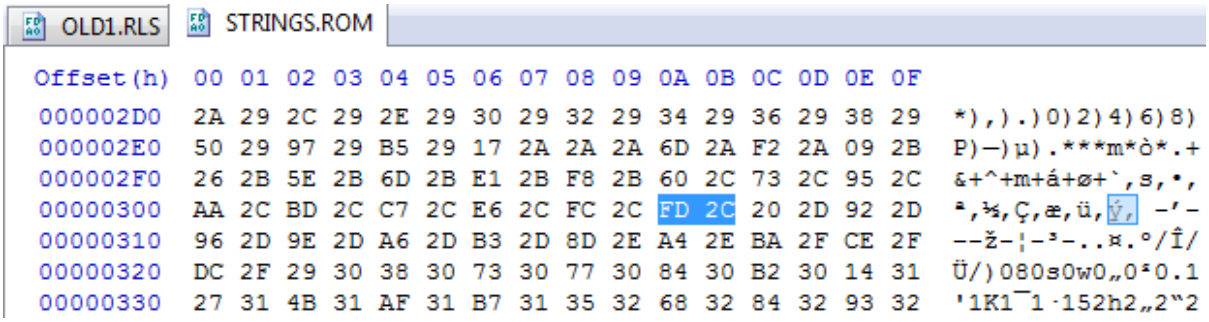


11) Search for “Intel(R) Virtualization Technology” in the **STRINGS.ROM** file:-



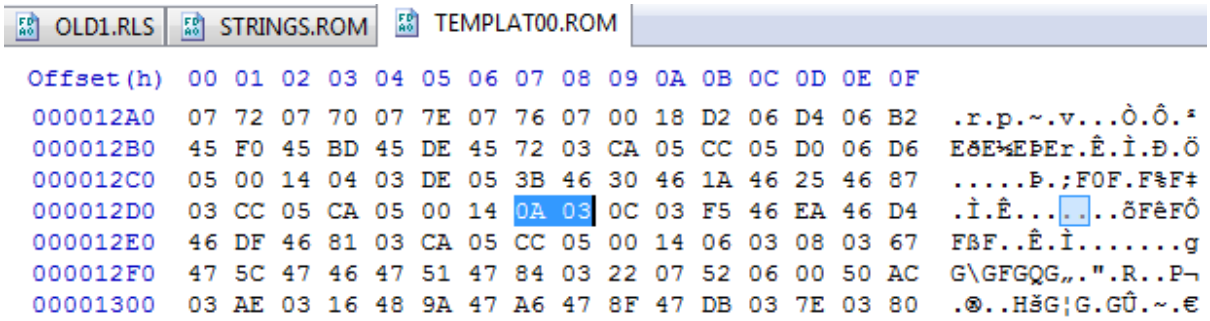
- In this example, we found out that the offset or location of the “Intel(R) Virtualization Technology” string or text is **2CFD**.

12) Now search for “FD 2C” (**2CFD** in endian format) in the **STRINGS.ROM** file. Make sure search from the beginning of the file:-



- In this example, the offset for “FD 2C” will be **030A**.
- Remember, the first result is what we are looking for.

13) Open **TEMPLAT0.ROM** with HEX editor & search for "0A 03" (**030A** in endian format):-



Note:-

00 = Pick Field

01 = Pick Field

10 = Generic Text

11 = Information

21 = Time

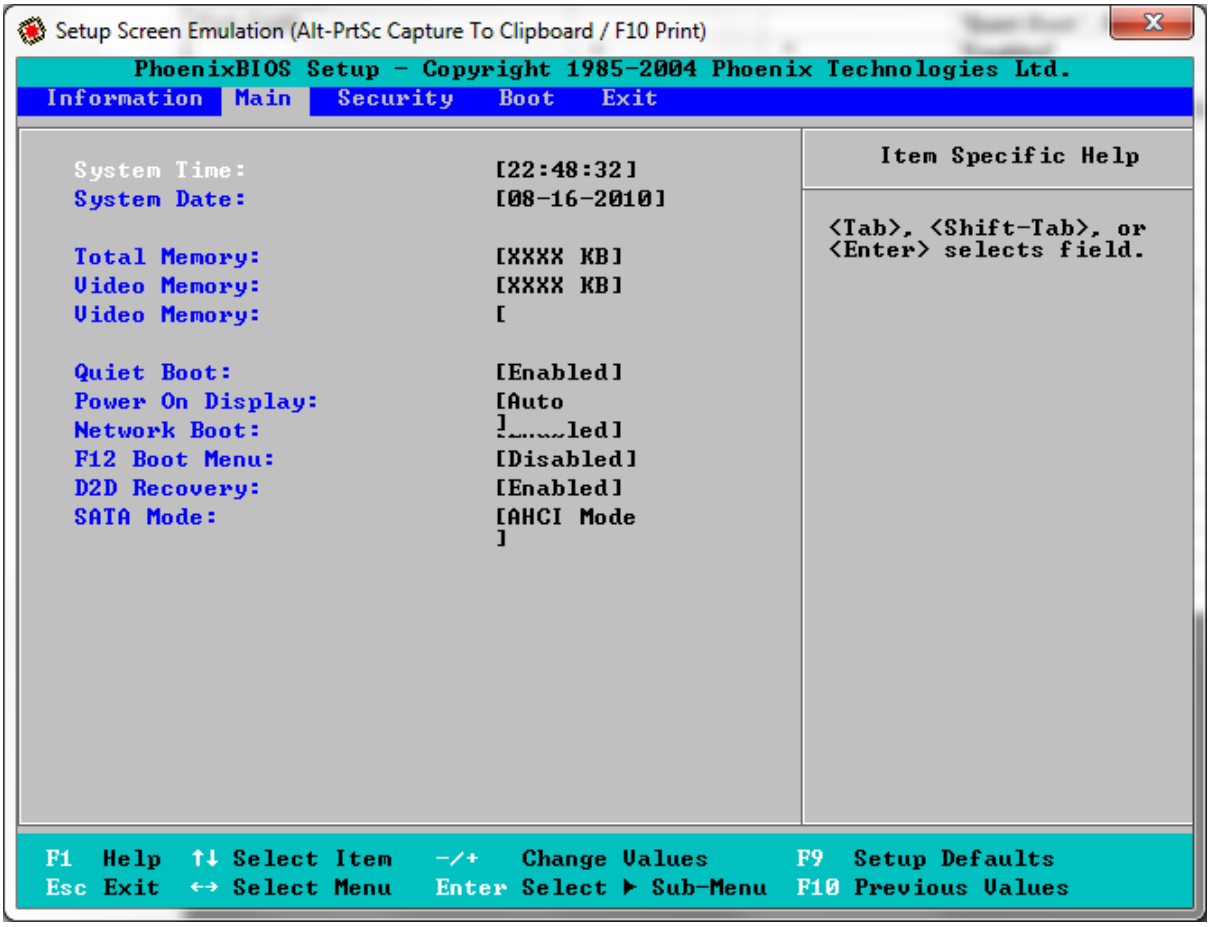
22 = Date

23 = Free form Hex

- Depend on the type of the menu we're going to unlock or unhidden but usually we're going to unlock or unhidden two types of menu. The two bytes before "0A 03" will determine which type of menu it is & we will call it **menu header** to differentiate it from others:-
  a) ***Pick Field*** type of menu. This type of menu allowed us to change the value such as enable/disable or change it to another value. In this example, it is clearly we're going to unlock or unhidden ***Pick Field*** type of menu. In this example, we found the **menu header** is "00 14"; **00** = Pick Field & **14** = Length. The **Length** value can be different. As I mention earlier, it is clearly "Intel(R) Virtualization Technology" menu is ***Pick Field*** type of menu, so if the first result have different pattern (**menu header**), continue the search until you found the right pattern.
  b) ***Information*** type of menu. I usually use this type of menu if I want to unlock hidden menu where it have sub-menus in it. It is a little bit complex to unlock this kind of menu but if you understand the process, it will be easy. The **menu header** will be look like this – "11 0C"; **11** = Information & **0C** = Length. The **Length** value can be different.
- Including the **menu header**, the offset for "Intel(R) Virtualization Technology" menu is **12D5**.
- Before we proceed, let us investigate more. Let us extract the 20 bytes, starting from the **menu header**:-
  ➔ 000012D5 00 14 0A 03 0C 03 F5 46 EA 46 D4 46 DF 46 81 03
  ➔ 000012E5 CA 05 CC 05
  00 = Pick field & 14 = Length
  0A 03 - Offset in strings
  0C 03 - Offset in strings
  TOKEN ID 0381 (81 03 in endian format)
  CA 05 - Offset in strings
  CC 05 - Offset in strings

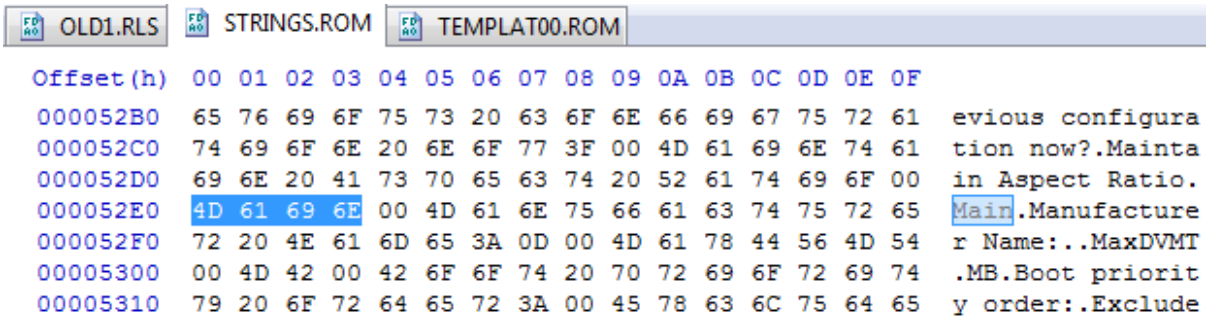
- Token ID = **0381** is actually the register which responsible in enabling or disabling the Virtualization feature. This is useful if we want to enable the Virtualization feature using **SYMCMOS.EXE** tool. Please visit these threads for more information:-
  http://forum.notebookreview.com/sony/189228-how-enable-intel-vt-ahci-napa-santa-rosa-platform-phoenix-bios-vaio-laptop.html#post2678924
  http://forum.notebookreview.com/acer/465936-acer-laptop-phoenix-bios-enable-virtualization-test-machine-acer-aspire-9420-a.html#post5991508

14) Now we have to decide where we want to put the "Intel(R) Virtualization Technology" menu. We can decide this by looking at the emulated BIOS Setup Menu in PBE.



- Referring to the screenshot above, the best place is to put it in **Main** section.
- Sometime there is another section called **Advanced** or **Intel** but it doesn't visible in real BIOS Setup Menu. So far there is no way to unhidden this section. The best way is to *move* the menu from the **Advanced** or **Intel** to **Main** section.
- Remember, the numbers of menu we can unhide are depending on the *space* available in the target section. Sometime we need to sacrifice other menu for more important menu.
- The most important rule that need to be follow & can't be broken is the size of the **TEMPLAT0.ROM** file must remain the same.

15) Search for “Main” string or text in **STRINGS.ROM** file:-

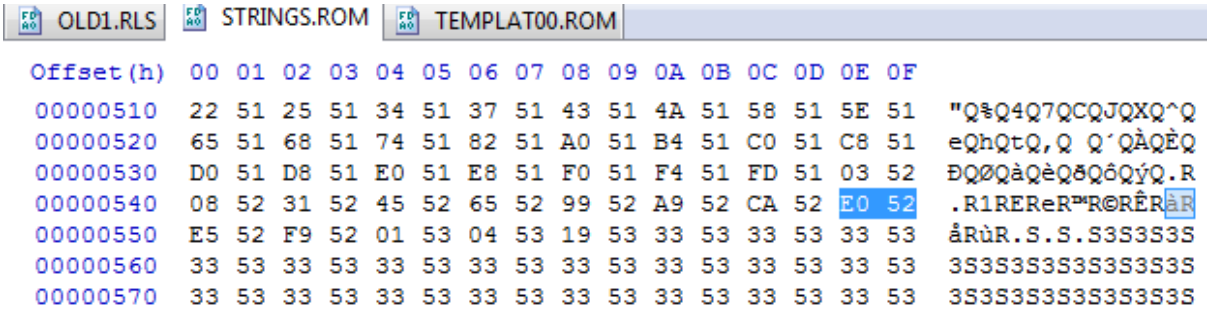


OLD1.RLS | STRINGS.ROM | TEMPLAT00.ROM

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
000052B0	65	76	69	6F	75	73	20	63	6F	6E	66	69	67	75	72	61	evious configura
000052C0	74	69	6F	6E	20	6E	6F	77	3F	00	4D	61	69	6E	74	61	tion now?.Mainta
000052D0	69	6E	20	41	73	70	65	63	74	20	52	61	74	69	6F	00	in Aspect Ratio.
000052E0	4D	61	69	6E	00	4D	61	6E	75	66	61	63	74	75	72	65	Main.Manufacture
000052F0	72	20	4E	61	6D	65	3A	0D	00	4D	61	78	44	56	4D	54	r Name:..MaxDVMT
00005300	00	4D	42	00	42	6F	6F	74	20	70	72	69	6F	72	69	74	.MB.Boot priorit
00005310	79	20	6F	72	64	65	72	3A	00	45	78	63	6C	75	64	65	y order:.Exclude

- The offset or location of the “Main” string or text is **52E0**.

16) Now search for “E0 52” (**52E0** in endian format) in the **STRINGS.ROM** file. Make sure search from the beginning of the file:-

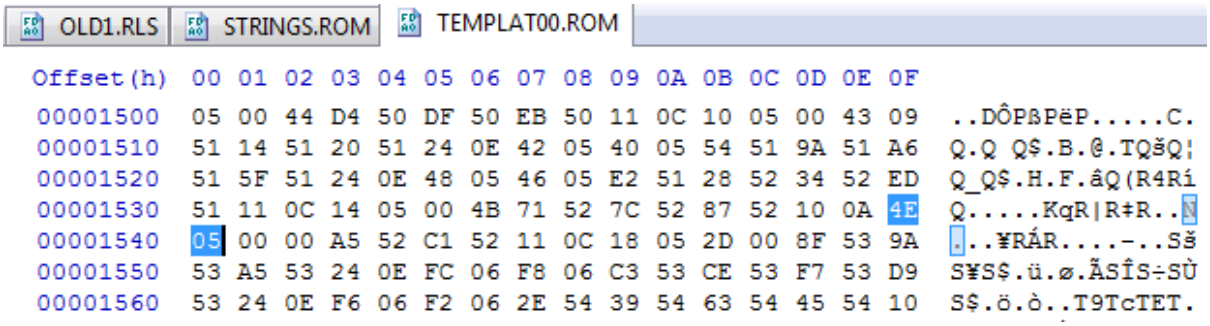


OLD1.RLS | STRINGS.ROM | TEMPLAT00.ROM

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000510	22	51	25	51	34	51	37	51	43	51	4A	51	58	51	5E	51	"Q%Q4Q7QCQJQXQ^Q
00000520	65	51	68	51	74	51	82	51	A0	51	B4	51	C0	51	C8	51	eQhQtQ,Q Q´QÀQÈQ
00000530	D0	51	D8	51	E0	51	E8	51	F0	51	F4	51	FD	51	03	52	ĐQØQàQèQðQôQýQ.R
00000540	08	52	31	52	45	52	65	52	99	52	A9	52	CA	52	E0	52	.R1REReR™R©RÊRàR
00000550	E5	52	F9	52	01	53	04	53	19	53	33	53	33	53	33	53	åRùR.S.S.S3S3S3S
00000560	33	53	33	53	33	53	33	53	33	53	33	53	33	53	33	53	3S3S3S3S3S3S3S3S
00000570	33	53	33	53	33	53	33	53	33	53	33	53	33	53	33	53	3S3S3S3S3S3S3S3S

- In this example, the offset for “E0 52” is **054E**.
- Remember, the first result is what we are looking for.

17) Search for “4E 05” (**054E** in endian format) in the **TEMPLAT0.ROM** file:-



OLD1.RLS | STRINGS.ROM | TEMPLAT00.ROM

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00001500	05	00	44	D4	50	DF	50	EB	50	11	0C	10	05	00	43	09	..DÔPßPëP.....C.
00001510	51	14	51	20	51	24	0E	42	05	40	05	54	51	9A	51	A6	Q.Q Q$.B.@.TQšQ¦
00001520	51	5F	51	24	0E	48	05	46	05	E2	51	28	52	34	52	ED	Q_Q$.H.F.âQ(R4Rí
00001530	51	11	0C	14	05	00	4B	71	52	7C	52	87	52	10	0A	4E	Q.....KqR|R‡R..N
00001540	05	00	00	A5	52	C1	52	11	0C	18	05	2D	00	8F	53	9A	...¥RÁR....-..Sš
00001550	53	A5	53	24	0E	FC	06	F8	06	C3	53	CE	53	F7	53	D9	S¥S$.ü.ø.ÃSÎS÷SÙ
00001560	53	24	0E	F6	06	F2	06	2E	54	39	54	63	54	45	54	10	S$.ö.ò..T9TcTET.

Note:-

00 = Pick Field

01 = Pick Field

10 = Generic Text

11 = Information

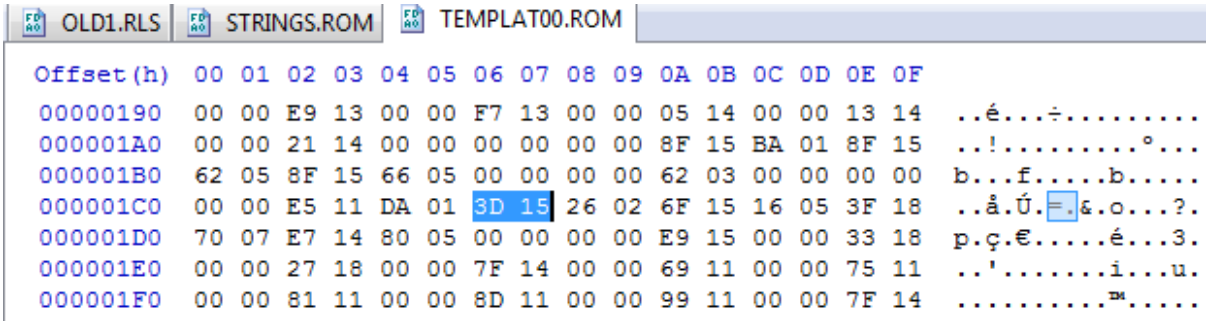
21 = Time

22 = Date

23 = Free form Hex

- The **menu header** is “10 0A”; **10** = Generic Text & **0A** = Length. The **Length** value can be different.
- The offset for “Main” section is **153D**.

18) Search for “3D 15” (**153D** in endian format) in the **TEMPLAT0.ROM** file:-

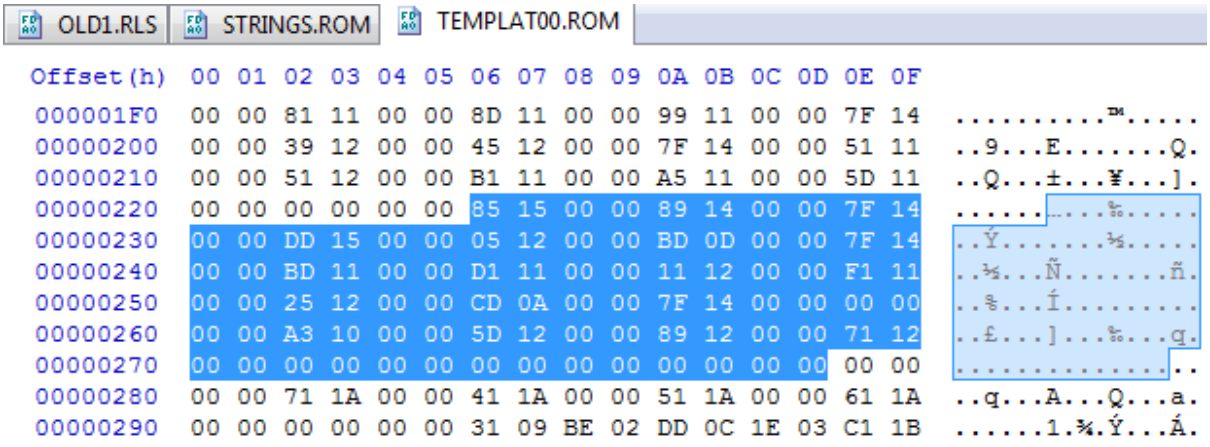


OLD1.RLS | STRINGS.ROM | TEMPLAT00.ROM

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00000190	00	00	E9	13	00	00	F7	13	00	00	05	14	00	00	13	14	..é...÷.....
000001A0	00	00	21	14	00	00	00	00	00	00	8F	15	BA	01	8F	15	..!.....°...
000001B0	62	05	8F	15	66	05	00	00	00	00	62	03	00	00	00	00	b...f.....b.....
000001C0	00	00	E5	11	DA	01	3D	15	26	02	6F	15	16	05	3F	18	..å.Ú.=.&.o...?.
000001D0	70	07	E7	14	80	05	00	00	00	00	E9	15	00	00	33	18	p.ç.€.....é...3.
000001E0	00	00	27	18	00	00	7F	14	00	00	69	11	00	00	75	11	..'.....i...u.
000001F0	00	00	81	11	00	00	8D	11	00	00	99	11	00	00	7F	14	.....™.....

- This is where the **Information**, **Main**, **Security**, **Boot** & **Exit** section are arranged & linked to their respective setup menu:-
  - ➔ 000001C0 00 00 E5 11 DA 01 3D 15 26 02 6F 15 16 05 3F 18
  - ➔ 000001D0 70 07 E7 14 80 05 00 00 00 00
  - (In endian format)
  - 11E5 = Information
  - 153D = Main
  - 156F = Security
  - 183F = Boot
  - 14E7 = Exit

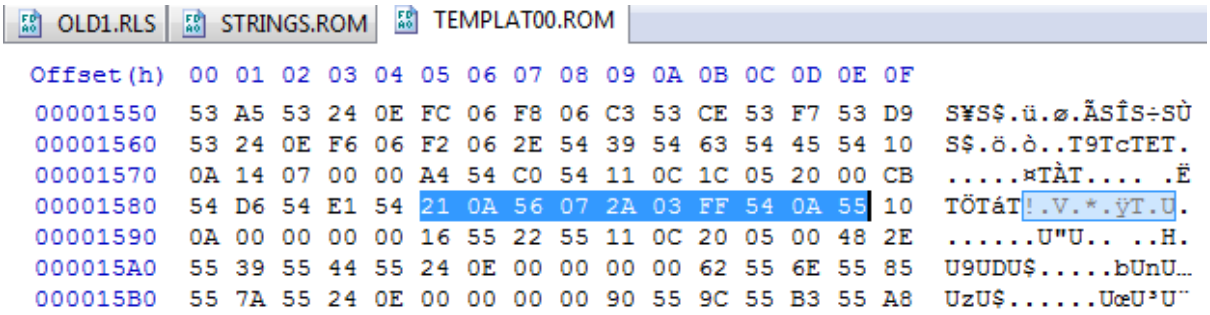
19) Referring to the above information, the **Main** (3D 15) section is linked to “26 02”. This is actually the offset for the **Main** setup menu. Jump to offset **0226**:-



OLD1.RLS | STRINGS.ROM | TEMPLAT00.ROM

Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
000001F0	00	00	81	11	00	00	8D	11	00	00	99	11	00	00	7F	14	.....™.....
00000200	00	00	39	12	00	00	45	12	00	00	7F	14	00	00	51	11	..9...E.....Q.
00000210	00	00	51	12	00	00	B1	11	00	00	A5	11	00	00	5D	11	..Q...±...¥...].
00000220	00	00	00	00	00	00	85	15	00	00	89	14	00	00	7F	14	..........‰.....
00000230	00	00	DD	15	00	00	05	12	00	00	BD	0D	00	00	7F	14	..Ý.....½.....
00000240	00	00	BD	11	00	00	D1	11	00	00	11	12	00	00	F1	11	..½...Ñ.....ñ.
00000250	00	00	25	12	00	00	CD	0A	00	00	7F	14	00	00	00	00	..%...Í.....
00000260	00	00	A3	10	00	00	5D	12	00	00	89	12	00	00	71	12	..£...]...‰...q.
00000270	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	00	.....
00000280	00	00	71	1A	00	00	41	1A	00	00	51	1A	00	00	61	1A	..q...A...Q...a.
00000290	00	00	00	00	00	00	31	09	BE	02	DD	0C	1E	03	C1	1B	......1.¾.Ý...Á.

- By referring to a screenshot of **Main** setup menu (at **#14**), we know that “85 15” is the “System Time” menu. The offset for “System Time” menu is **1585**.
- To confirm this, jump to offset **1585**:-

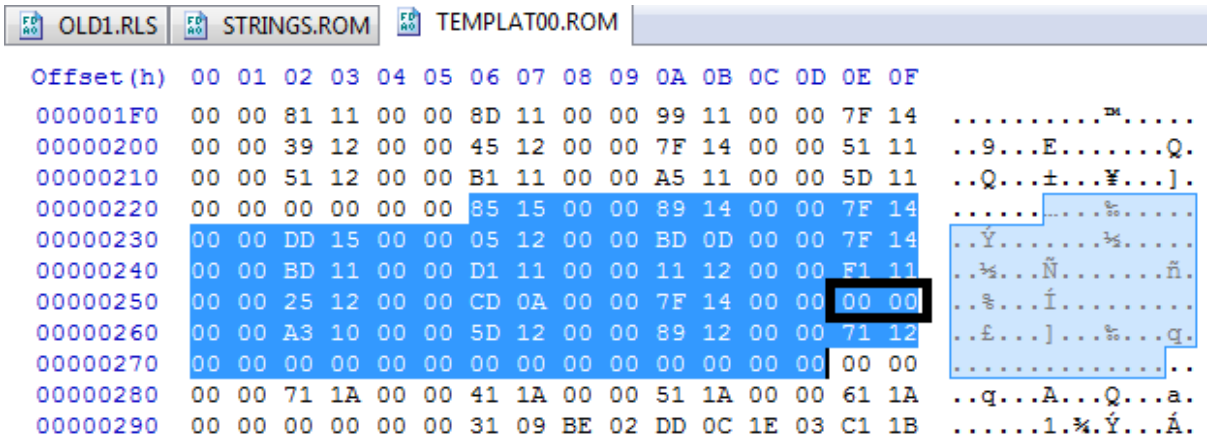


OLD1.RLS | STRINGS.ROM | TEMPLAT00.ROM

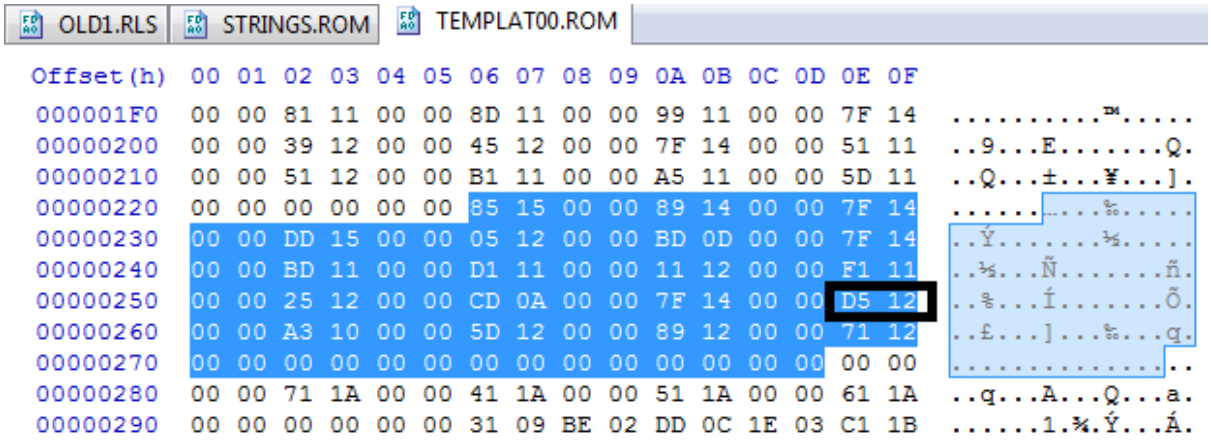
Offset(h)	00	01	02	03	04	05	06	07	08	09	0A	0B	0C	0D	0E	0F	
00001550	53	A5	53	24	0E	FC	06	F8	06	C3	53	CE	53	F7	53	D9	S¥S$.ü.ø.ÃSÎS÷SÙ
00001560	53	24	0E	F6	06	F2	06	2E	54	39	54	63	54	45	54	10	S$.ö.ò..T9TcTET.
00001570	0A	14	07	00	00	A4	54	C0	54	11	0C	1C	05	20	00	CB	.....¤TÀT.... .Ë
00001580	54	D6	54	E1	54	21	0A	56	07	2A	03	FF	54	0A	55	10	TÖTáT!.V.*.ÿT.U.
00001590	0A	00	00	00	00	16	55	22	55	11	0C	20	05	00	48	2E	......U"U.. ..H.
000015A0	55	39	55	44	55	24	0E	00	00	00	00	62	55	6E	55	85	U9UDU$.....bUnU…
000015B0	55	7A	55	24	0E	00	00	00	00	90	55	9C	55	B3	55	A8	UzU$.....UœU³U¨

  - ➔ 00001585 21 0A 56 07 2A 03 FF 54 0A 55
- Type = 21 (Time), Length = 0A, 1st offset in strings = 0756, 2nd offset in strings = 032A
  54FF & 550A are fillers

20) Now we will need to put the “Intel(R) Virtualization Technology” menu in **Main** setup menu. First, we need to identify where is the best place to put the new menu:-



- Since we have plenty *empty* space, we will use offsets **025E** & **025F**. This is where we will put the “Intel(R) Virtualization Technology” menu.
- Remember, each menu is separated with “00 00”. If, let say, “Menu A” & “Menu B” menu are separated with four bytes or more of **00** (e.g. “00 00 00 00”), the two menu are considered *disconnected* which means “Menu B” will not appear or visible in the setup menu.
- Referring to **#13**, we know the offset for “Intel(R) Virtualization Technology” menu is **12D5**. So, we will need to replace the “00 00” at offset **025E** & **025F** with “D5 12” (**12D5** in endian format):-



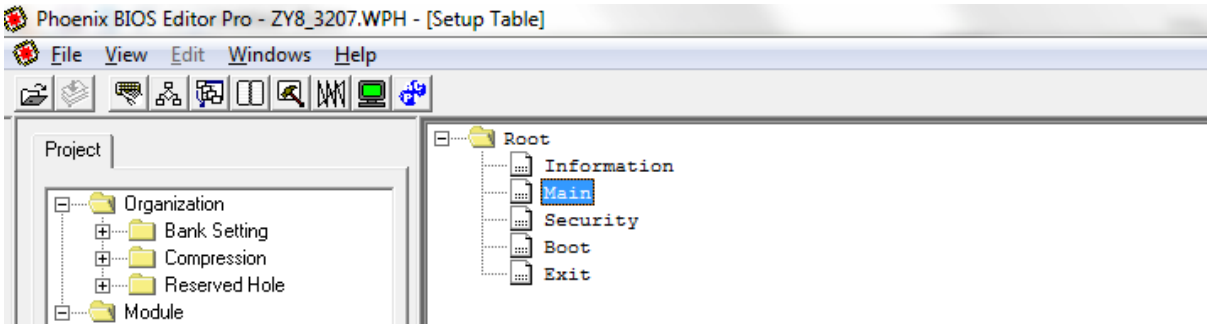
- Save the modified **TEMPLAT0.ROM** file.

21) Copy the modified **TEMPLAT0.ROM** file back to PBE’s **TEMP** folder:-

- 64bit OS: C:\Program Files (x86)\Phoenix Technologies Ltd\BIOS Editor\TEMP
- 32bit OS: C:\Program Files\Phoenix Technologies Ltd\BIOS Editor\TEMP

22) Finally, use PBE to re-build the modified BIOS image. The **Build BIOS** button is not enabled by default which will prevent us from re-building the modified BIOS image. The button will only enable if PBE detect modification to the original BIOS image. So, the solution is we'll need to do some *changes*.

- At PBE, go to **Setup Table** & click **Main**:-



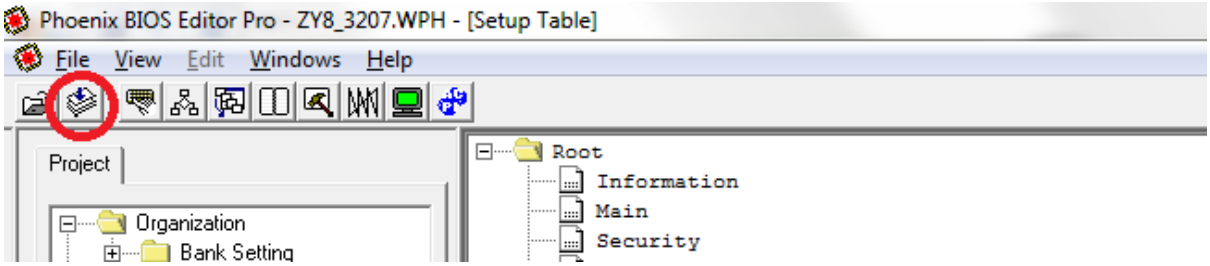
- We should see something like this:-

Emu

Type	S_Default	M_Default	Description
Time			'System Time:'
Date			'System Date:'
Generic Text			' '
Information			'Total Memory:', 0Dh
Information			'Video Memory:', 0Dh
Pick Field			'Video Memory:', 0Dh
			'32 MB'
			'64 MB'
			'128 MB'
	*	*	NULL
Generic Text			' '
Pick Field			'Quiet Boot:', 0Dh
	*	*	'Enabled'
			'Disabled'
Pick Field			'Power On Display:', 0Dh
	*	*	'Auto', 0Dh
			'Both'
Pick Field			'Network Boot:', 0Dh
			'Disabled'
	*	*	'Enabled'
Pick Field			'F12 Boot Menu:', 0Dh
	*	*	'Disabled'
			'Enabled'
Pick Field			'D2D Recovery:', 0Dh
	*	*	'Enabled'
			'Disabled'
Pick Field			'SATA Mode:', 0Dh
			'IDE Mode', 0Dh
	*	*	'AHCI Mode', 0Dh
			NULL
			NULL
Generic Text			' '

- As we can see, there are a lot of "*" character in **S_Default** & **M_Default** column. It is indicating the default value or setting for the specified menu. We're not going to change it though. Just double click at any "*" character & PBE will detect it like we just do a modification to the original value.

- This will enable the **Build BIOS** button:-



- Now we can proceed to re-build the modified BIOS image.

23) A little list of **DO's & DON'T's** & also notes:-
    - Make sure to compare the modified BIOS image with the original image.
    - Pay attention to anything out of the ordinary or weird like there is a lot of **FF** at the beginning of the modified BIOS image. If so, repeat the BIOS re-building process.
    - Just to make sure, repeat the process two to three time & compare the modified BIOS image with the one we re-build before. If we get identical modified BIOS image each time, repeatedly, the modified BIOS image was re-build properly but this doesn't mean the modified BIOS image won't brick our computer.
    - Sometime, manufacturers tend to put a header or footer, like a signature, to the BIOS image. If we forgot to reintegrate it with the modified BIOS image, when we try to flash BIOS with it, depend on the type of the signature, sometime the BIOS flash program will abort the process.
    - As I mention earlier, do not change the size of the files or BIOS image. If we do, there is high possibility it will brick our computer. I'm 100% sure about it.
    - Make sure to *master* the BIOS recovery technique.

As usual, I will not responsible for any problem including the problem caused by the information provided in this tutorial.

Good luck. :D

/* kizwan */