<u>**Overview**</u>

** This tool is freeware and may not be sold or redistributed for a charge **

This tool inserts a SLIC table into Phoenix (original and Lenovo) and Dell BIOSes and EFIs.

There are 3 main code paths:
- One for Dell
- One for Phoenix (old and new, including Lenovo)
- One for EFI (and Insyde)

Exactly what happens varies for each one. Generally dynamic means an existing SLIC is updated and SSV2 means a NEW SLIC is inserted.

Dell
- SSV2 - a new SLIC is inserted in one of the first modules
- Dynamic - an existing SLIC and any OEM/Table ID strings are updated

Phoenix
- Module - for interest only - uses the Phoenix tools PREPARE/CATENATE (old) and FI/FP (new) - this often screws the bios and causes bricks
- SSV2 - a new SLIC is inserted in a new ACPIxx module
- Dynamic - an existing SLIC is updated, OEM/Table ID strings updated, locks removed

EFI (All)
- Module - a pubkey and marker are inserted as two new modules with a typical GUID (this was how Insyde was originally done - but still works for some EFIs). In addition for EFIs, if it is not already present, the SLPSupport module (GUID 4C494E55-5849-5342-4554-544552212121) will be added.
- SSV2 - a module is modified with code to build a SLIC table at boot. This can cause a brick and is no longer first line.
 - Dynamic - an existing SLIC table is modified and strings updated.

EFI (ASUS)
In addition to the above methods, there are 3 ASUS specific methods:
- NVRAM – The NVRAM is modified to contain SLIC data. This method requires a dumped BIOS (ie. It is individual to the computer). It is also persistent and survives subsequent flashes.
- New module – If a SLP2 module (GUID A1902AB9-5394-45F2-857A-12824213EEFB) is already present then a combined pubkey+marker module is added. In addition a signing key is added (significance uncertain).
- DMI – SLIC is added in the DMI region.

Method

1) Flash, using the manufacturers recommended method, the unmodified BIOS of the same version. This allows the bootblock etc. to be updated.
2) Mod the BIOS using the tool.
3) Flash the SLIC'd BIOS from DOS. This ensures the bootblock is left intact and maximises the chance of a successful recovery if a problem was to occur (see below for more information).
4) You then still need to use the appropriate key and certificate for Vista and Windows 7.

Many thanks to Yen, Busykid508, 911medic, shakeyplace, str8, phrunt, rbjack, Zort, qtm and many others for their invaluable advice and feedback. Special thanks to phrunt for experimental testing on hardware and to busykid508 for tirelessly using and giving feedback. Thanks also to middleton for guiding me through EFI modification.

Notes:
- Please check for an existing mod first: http://forums.mydigitallife.info/showthread.php?t=7500 and http://forums.mydigitallife.info/forums/33-BIOS-Mod-Requests-Post-Requests-Only.
- BIOS flashing is not for the faint hearted. There is ALWAYS an element of risk. Please ensure you are familiar with the procedure, and have necessary floppy disks etc., to allow a recovery in case of a flash failure. **THIS IS ESPECIALLY TRUE FOR PHOENIX BIOSES.**
- We assume you have a legitimate edition of Vista or Windows 7. Piracy is NOT condoned.
- Please remove any softmods. For Vistaloader this includes repairing the boot sector and MBR of the hard disk.
- The tool requires the .NET Framework 2.0 and VC 2008 runtime.

Flashing
- always best to flash from DOS of from within the BIOS/EFI itself
- For ACER Phoenix BIOSes – use the flash tool that comes with the BIOS
- For Phoenix Bios: *ph161700 /X /FORCE /O /C /S <NameOfBIOSFile>*
 (http://www.sendspace.com/file/g83uqv). Don't use the /BBL switch.
 You can use winphlash for x32 (ensure Advanced=1 and Hide=0 in [UI] section of phlash.ini).
 Do NOT use winphlash for x64.
- For Insyde – use the DOS flash tool that comes with the original BIOS or
 flashit <NameOfBIOSFile> /all
 (http://slics.myftp.org/Tools/FlashIt/FlashIt.rar)
- For Dell: use either the modified EXE created by the tool or run the original BIOS EXE with
 '*-readgzfile BIOS.GZ*' (minus quotes) as command line
- For Dell BIOSes with Multiple GZIPs (some Optiplex models) – look in log – use the Dell Client Configuration Utility (DCCU).

Flash recovery information:
- Read here - http://www.biosman.com/biosrecovery.html.
- Also read here - http://forums.mydigitallife.info/showthread.php?t=2105
- For Insyde read here - http://forums.mydigitallife.info/showthread.php?t=7033
- For ACER read here - http://forums.mydigitallife.info/showthread.php?t=13095
- Ensure that the version on the disk is the (unmodified) version you are trying to flash with SLIC - avoids problems with bootblock flashing.
- **PLEASE ENSURE YOU HAVE CREATED A PHOENIX CRISIS DISC OR INSYDE RECOVERY DISC BEFORE ANY FLASHING**

Other Information

SLIC files
The program needs:
- A 374 byte SLIC file. Checksum is automatically corrected.
- For 2.1 SLICs look in http://forums.mydigitallife.info/showthread.php?t=5952.

RW Everything reports
- **When performing ANY Phoenix or Insyde mod and SOME Dell mods YOU WILL NEED a RW Everything report** (see http://forums.mydigitallife.info/threads/29789-RW-Everything-Latest-News) from the UNMODDED BIOS to load into the tool before modding the BIOS.
- For phoenix, when the RW report is loaded the tool will automatically select SSV2 or dynamic depending on the presence of a SLIC table in the report:
     - If a SLIC table is present -> dynamic is selected
     - If a partial SLIC table is present -> SSV2 is selected and 'Replace existing SLIC elements is also selected
     - If no SLIC table is present -> SSV2 is selected and 'Replace existing SLIC elements is not selected
- For EFI
     - If a SLIC table is present -> dynamic is selected
     - Otherwise -> module is selected

Vista Keys and Certificates
- Keys can be found here - http://forums.mydigitallife.info/showthread.php?t=2581
- Certificates can be found here - http://rapidshare.com/files/118119842/39-CERTS.zip

Windows 7 Keys and Certificates
- Keys can be found here - http://forums.mydigitallife.info/showthread.php?t=10370
- Certificates can be found with SLICs here - http://forums.mydigitallife.info/showthread.php?t=5952

Note a certificate is NOT needed to mod.
The tool just confirms that the certificate specified matches the SLIC specified.

General Options

Don't alter any ACPI tables

No ACPI tables will have their OEM/Table IDs altered.
For phoenix this is true whether in the main ACPI (BIOSCODxx) module or in seperate ACPIxx modules.

Note – existing SLIC tables/elements will still be changed.

Don't alter any ACPI tables or OEM/Table ID strings if SLIC IDs match

If the OEM/Table ID of the NEW SLIC matches the OLD SLIC, then no ACPI tables or OEM/Table ID strings are altered.

Note – this overrides any other options involved in table modification and OEM/Table ID string modification
Note – If the OEM/Table IDs of the SLICs don't match then this option is ignored

Only alter RSDT and XSDT tables

Only the RSDT, XSDT and existing SLIC tables will have their OEM and Table IDs updated with that of the new SLIC table. DSDT table is also updated if manufacturer is ASRock.

Note – For Phoenix BIOSes 'Only alter ACPI modules in the main (BIOSCODxx) module' applies to this option.

Only replace OEM ID in additional tables

When any non-RSDT/XSDT tables are updated; only the OEM ID will be updated, not the Table ID.

Only alter tables in the main ACPI module

For Phoenix BIOSes, only tables found in the main ACPI module will be updated. Exactly which ones and how they are updated depends on the state of 'Only alter RSDT and XSDT tables' and 'Only replace OEM ID in additional tables'.

Note – if this is ticked then other options are also influenced but all BIOSCODxx modules (not just the main ACPI containing module) are altered by these options. It is only ACPI tables that are only altered if they occur in the one main BIOSCODxx module (Phoenix)
Note – For Dell BIOSes this is the module (normally 01 or 02) that contains the ACPI tables
Note – For EFI this is the module(s) with the GUID 16D0A23E-C09C-407D-A14A-AD058FDD0CA1

Scan ACPI modules for OEM/Table IDs (Phoenix only)

By default all OEM/Table IDs found in the main ACPI (BIOSCODxx) module can be replaced (depending on later options). This option also scans all ACPIxx modules for additional OEM/Table IDs.

Note – ticking this option does not alter anything in the modules directly. It just influences other options.

Scan BIOSCOD modules for LENOVO IDs (Phonenix only)

Scan the BIOSCODx.ROM modules for extra LENOVO OEM/Table IDs.

Note – 'Only replace complete OEM/Table IDs preceeded by C3h' does NOT apply.
Note – This options is selected by default if manufacturer is Lenovo
Note – This option is ignored with a valid RW Everything is parsed

Replace additional OEM/Table ID

Allows two user specified OEM/Table ID (14 character) or Table ID (8 character) strings.

Note – ticking this option does not alter anything in the modules directly. It just influences other options.
Note – 'Only replace complete OEM/Table IDs preceeded by C3h' does not apply to user specified Ids.
Note – A * is considered to be 00h

Replace all OEM/Table ID occurences

Replace any occurence of the identified OEM/Table IDs. If this is unticked then no OEM/Table ID strings, either complete or split, are changed, other than those changed as part of the ACPI tables.

Note – For Phoenix BIOSes, if 'Only alter tables in the main (BIOSCODxx) module' is ticked then OEM/Table ID strings will only be replaced in BIOSCODxx modules.
Note – if 'Don't alter any ACPI tables' is ticked then any OEM/Table ID found in a valid ACPI table is ignored.
Note – If the OEM/Table ID is found to be in a valid ACPI table then the OEM/Table ID is altered respecting 'Only alter RSDT and XSDT tables' and 'Only replace OEM ID in additional tables'.

Only replace complete OEM/Table IDs preceeded by C3h (Phoenix only)

Only replace complete (non-split) OEM/Table ID strings that are NOT part of a valid ACPI table (see above for these) that are preceeded by a C3h byte. See 'Scan ACPI modules for OEM/Table IDs' and 'Replace additional OEM/Table ID' which describe how possible OEM/Table ID strings are identified.

Note – if 'Only alter tables in the main (BIOSCODxx) module' is ticked then OEM/Table ID strings will only be replaced in BIOSCODxx modules.
Note – this does NOT apply to 1) User specified additional OEM/Table IDs, 2) OEM/Table IDs from the RSDT/XSDT tables in a RW Everything report (it DOES apply to Table IDs only from the RW Everything report) and 3) Split OEM/Table IDs and split Table IDs.

Replace split OEM/Table IDs

OEM/Table ID strings that are split 4+2+4+4 are replaced. See 'Scan ACPI modules for OEM/Table IDs' and 'Replace additional OEM/Table ID' which describe how possible OEM/Table ID strings are identified.

Note – 'Only replace complete OEM/Table IDs preceeded by C3h' does not apply.

Replace split Table IDs

Table ID strings that are split 4+4 are replaced. See 'Scan ACPI modules for OEM/Table IDs' and 'Replace additional OEM/Table ID' which describe how possible Table ID strings are identified.

Note – 'Only replace complete OEM/Table IDs preceeded by C3h' does not apply.

Replace 'ALASKA A M I' OEM ID (EFI only)

If no RW report is specified then the 'ALASKA A M I' OEM/Table ID is replaced.

Replace '_ASUS__ASUS_' OEM ID (EFI only)

Replace the '_ASUS__ASUS_' OEM IDs found in ASUS EFIs

Note – Selected as default when ASUS manufacturer detected/selected

Only replace OEM/Table IDs from RSDT/XSDT in the RW Everything report

Only the OEM/Table ID(s) that appear in the RSDT and XSDT tables in the RW Everything report will be replaced. This applies to those in ACPI tables and also those found in other modules. Note that all the other options still apply (ie. ACPI table selection options, preceeding C3h option and split IDs options).

Note – If there is no valid RW Everything report then NO OEM/Table IDs will be replaced.

Note – If 'Blank other ACPI table OEM/Table IDs' is selected then the OEM/Table ID in all applicable ACPI tables will be blanked, ignoring this option.

Replace Table ID from RSDT/XSDT tables in the RW Everything report

Treat the Table ID from the RSDT and XSDT tables as strings to be altered in their own right (eg. for HP BIOSes with Capell00).

Note – 'Only replace complete OEM/Table IDs preceeded by C3h' does apply.
Note – all relevant options above still apply.

Only replace SLIC elements that appear in the RW Everything report

Only SLIC elements that match the SLIC table from the RW Everything report are replaced. This does NOT include SLIC headers that are part of a full SLIC table or that are adjacent to a pubkey or marker (as the header can be modified by code and hence would not be expected to match). SLIC headers that are not in isolation are not matched but will be replaced if the other element(s) match, ie. If a non-matching SLIC header is part of a full SLIC table whose pubkey and marker match the RW Everything report, then the whole SLIC table, including the header, is replaced. All isolated SLIC headers will be replaced irrespective of matching (for the reasons above), but the log will only indicate a match if they do indeed match exactly.

Note – If there is no SLIC table in the RW Everything report then NOTHING will be replaced when this option is ticked.
Note – If there is no valid RW Everything report then this option is ignored.
Note – this is only active if 'Replace existing SLIC elements' for SSV2 mods.

Only copy SLIC header OEM/Table ID

When inserting a full SLIC table, or SLIC header, only the OEM and Table IDs are copied; the remainder of the header is left intact.

Replace empty modules

Replace any appropriately sized and empty modules with the corresponding SLIC element. Currently will look for 374 bytes (full SLIC table), 338 bytes (pubkey and marker), 156 bytes (pubkey) and 182 bytes (marker).

Remove manufacturer specific locks

Remove various manufacturer specific locks (governed by the manufacturer selceted in drop down box). If unticked then no locks are removed.

Force SLIC in last location in RSDT (Dell only)

The SLIC table address is always placed in the last location of the RSDT even if it is already occupied.

Force SLIC into MCFG location (Dell only)

Force placement of the SLIC address in the MCFG location in the RSDT tabe in the ACPI module. The MCFG table is lost.

Note – The MCFG table is automaticalyl replcaed if the tool determines the mod would not be successful in the traditional way
Note – This option overrides the 'Force SLIC in last location in RSDT' option

Insert SLP10 module for SLP mods (EFI only)

Default method of inserting an SLP 1.0 string is by modifiying the module containing 'AMIBIOS 0800'. With this method an additional module is included in the EFI that is respondible for inserting an SLP1.0 string.

Note - This is default for Module and New Module methods
Note - It is not possible to specify a location in memory

SSV2 Options

Replace existing SLIC elements

All identified SLIC elements, wherever they occur, are replaced with the corresponding element from the new SLIC. This includes headers, pubkeys and markers (and any combinations).

Note – this always happens in dynamic mods.
Note – this option is affected by the 'Only replace SLIC elements that appear in the RW Everything report' option

Replace existing SLIC table ID with [   ]

The 'SLIC' string in ANY existing SLIC element in ANY module will be replaced with the specified 4 character string (default 'OEMS')

Note – this is not influenced by any of the above options. ALL modules are included
Note – this is NEVER performed for a dynamic mod

Replace all 'SLIC' occurences

Replace any occurence of 'SLIC' string, that does not occur in an identified OEM/Table ID, with [   ] (see above).

Note – this is NEVER performed for a dynamic mod
Note – For Phoenix BIOSes, if 'Only alter tables in the main (BIOSCODxx) module' is ticked then 'SLIC' will only be replaced in BIOSCODxx modules AND in MOD_xxyy modules.

Only insert SLIC module (Phoenix only)

Does not attempt to find and patch ACPI modules. Will replace OEM/Table ID strings, SLIC elements and 'SLIC' strings.

Note – this is NEVER performed for a dynamic mod

Allow manual choice of SLIC location (Phoenix only)

Manually select the location, from a list of candidate locations, to insert the SLIC ACPI module. The tool will automatically scan for all valid locations before and after modules.

Manual location for SLIC insertion (Phoenix only)

The SLIC table will be inserted at this (hex) address in the ROM image. The tool will still check that it is valid (ie. not in a module or ROM hole).

Blank other ACPI table OEM/Table IDs (Phoenix only)

If ticked then the OEM/Table ID in ACPI tables, other than RSDT,XSDT and SLIC, are replaced with spaces. This aids module resizing.

Replace specific strings (Phoenix only)

Replace manufacturer specific strings to try and reduce module size. This is attempted after creator IDs are blanked and appropriate OEM/Table Ids are blanked, if 'Blank other ACPI table OEM/Table Ids' is selected. For example, 'Sony Corporation'.

Place SLIC module after last BIOSCOD module (Phoenix only)

Instead of placing the new ACPIxx module at the end of the module chain, it is inserted after the last BIOSCOD module.

Dynamic Options

Preserve module size

For Phoenix BIOSes, If ticked an SSV2 style mod is performed (and all the options for SSV2, other than those noted, apply). If unticked then Prepare/Catenate are used.

For EFIs this simply means that attempts will be made to adjust the altered module sizes so they are unchanged from the original. Selected as default for ASUS manufacturer.

Control Options

Ask prior to each modification

A dialog box allows the user to choose whether each modification to a module or the BIOS image is implemented.

Allow user modification of modules

If the tool fails to resize a module automatically the user is given the oppertunity to modify the module manually, the tool then has one further attempt (without any additional modification) to resize the module.

Always allow user modification of modules

The user is always given the oppertunity to modify the modules manually, irrespective of sizing success, the tool then confirmes the new module will fit.

Allow user to modify other modules

Pause to allow user to modify any module in the DUMP directory.

Extract modules when verifying

When verifying the BIOS, extract the modules as well.

No SLIC

No SLIC is inserted. Only SLP1.0, locks and DSDT adjustments are made.

Relax FV boundry checks (EFI only)

Sometimes a firmware volume is embdeed directly in another firmware volume. Normally the tool checks that the embedded firmware volume starts and ends in the containing module. When enabled the firmware volume is allowed to span modules (which according to specs mean the headers are wrong but this is seen in some Dell EFIs)

Insert gap if module shrinks (EFI only)

If a module shrinks then default behaviour is to adjust offsets of all subsequent modules until the end of the FV is reached. When ticked, and if suffcent space, a gap is inserted. This will preserve the offset of subsequent modules but alter the module count.

Insert new modules before gap (EFI Only)

When inserting a new module at the end of a Firmware Volume and there is a gap in the last 3 modules, insert the new module before the gap.

Allow FV selection for Module methods (EFI only)

Assume user to select FV to insert new mondules into for Module and New Module methods of modification.

Compress stored submodules when inserting new modules (EFI only)

When inserting new modules, if a submodules exists that is flagged as compressed (and has a compression header) but the compression method is 0 (stored – ie. uncompressed) then it is compressed and the header updated.

General Notes

OEM/Table ID handling

For Phoenix and Dell BIOSes

SCANNING - For all BIOSes these ALWAYS occur:
- the main BIOSCODxx module (containing the RSDT table) is scanned for valid ACPI tables
- all ACPI tables found are scanned for OEM/Table IDs
- any existing SLIC tables/headers/markers are scanned for OEM/Table IDs

EXTRAS:
- if the 'scan ACPI modules for OEM/Table IDs' is ticked then all valid ACPI modules in ANY module are scanned for OEM/Table IDs
- any additional user specified OEM/Table IDs are added
- If there is a RW report then every table in the report is scanned for OEM/Table IDs
- If 'Replace Table ID from RSDT/XSDT tables in the RW Everything report' is selected then the Table ID from RSDT and XSDT are included in their own right
- If 'Scan BIOSCOD modules for LENOVO IDs' and there is no RW report then the BIOSCODx.ROM modules are scanned for Lenovo OEM/Table IDs

The list can be seen in the log:

```
OEM/Table IDs identified are:
1.  FSC    PC
2.  FUJ    FJNB1AF
3.  FUJ    PC
4.  DOCK   FJNB1AF
5.  CPPM   CpuPm
6.  CP0I   Cpu0Ist
7.  CP1I   Cpu1Ist
8.  CP0C   Cpu0Cst
9.  CP1C   Cpu1Cst
10. AHCI   FJNB1AF
11. IDE    FJNB1AF
12. VISTA  FJNB1AF
```

For EFIs

SCANNING:
- any existing SLIC tables/headers/markers are scanned for OEM/Table IDs
- if the 'Replace 'ALASKA A M I' OEM ID is ticked, and there is no RW report, then 'ALASKA A M I' is included

EXTRAS:
- As above with the exception of the Lenovo IDs

Replacement

For Phoenix and Dell BIOSes

Then, depending on the selected options, occurrences of these OEM/Table IDs are changed to the one from the new SLIC table.

By default this includes:
- Those in ACPI tables in the main BIOSCODxx module (depends on the 'Don't alter any ACPI tables' option, the 'Only alter RSDT and XSDT tables' option and the 'Only replace OEM ID in additional tables') are changed

- Stand alone strings are replaced dependent upon the 'Only replace complete OEM/Table IDs preceded by C3h' - by default those preceded by a C3h byte are replaced (option is ticked)
- The C3h option does NOT apply to User specified additional OEM/Table IDs, OEM/Table IDs from the RSDT/XSDT tables in a RW Everything report (it DOES apply to Table IDs only from the RW Everyting report) and Split OEM/Table IDs and split Table IDs
- Split (in 4+2+4+4 pattern) OEM/Table IDs are replaced dependent on the 'Replace split OEM/Table IDs' option – by default this is selected

EXTRAs:
- If 'Only alter tables in the main ACPI (BIOSCODxx) module' is UNTICKED then all ACPI tables in all modules have their OEM/Table ID (or just OEM ID depending on the above options) changed.
- If 'Replace split Table IDs' is ticked then isolated split (in a 4+4 pattern) table IDs are replaced.

NOTE:
- if 'Only replace OEM/Table IDs from RSDT/XSDT in the RW Everything report' is selected then only these IDs are replaced

For EFIs

Likewise, depending on the selected options, occurrences of these OEM/Table IDs are changed to the one from the new SLIC table.

By default this includes:
- Those in ACPI tables (depends on the 'Don't alter any ACPI tables' option, the 'Only alter RSDT and XSDT tables' option and the 'Only replace OEM ID in additional tables') are changed
- Split (in 4+2+4+4 pattern) OEM/Table IDs are replaced dependent on the 'Replace split OEM/Table IDs' option.

Extras and notes as above

Handling of SLIC elements (Phoenix and Dell)

A SLIC element is considered to be a SLIC header, pubkey or marker (or any combination thereof). A full SLIC table includes 3 elements – the header, the pubkey and the marker.

<u>With NO RW report</u>

- SSV2 – No existing elements are replaced. A new SLIC table is added.
- Dynamic - ALL existing elements are replaced.

<u>With RW report</u>

The tool makes the following 3 decisions sequentially:

1) If there is an EXISTING whole SLIC in the BIOS AND a whole SLIC in the RW report then dynamic is selected with ONLY the elements matching those in the RW report being replaced in the BIOS (unless any element is in the bootblock in which case SSV2 is selected).

2) If there are ANY SLIC elements in the RW report then SSV2 is selected WITH matching elements from RW report replaced in the BIOS.

3) If there are NO SLIC elements at all in the RW report then SSV2 is selected with nothing replaced (as above).

Determining Compressed SLIC size

This is useful for SSV2 Phoenix mods. Click the '*' button after loading a BIOS. The tool will then compress each element of every SLIC in the SLIC directory and create a .CSV file with the results. This uses the PREPARE/CATENATE tools so is slow.

```
Filename,Full SLIC,SLIC Header,SLIC Header and Pubkey,SLIC Header and Marker,Pubkey,Marker,Pubkey and Marker
ACER.BIN,415,71,261,265,232,246,398
Acer[ACRSYSACRPRDCT-ANNI]2.1.BIN,415,71,261,265,232,246,398
Alienware[ALWAREALIENWRE-MSFT]2.1.BIN,421,68,259,263,231,250,402
```

For example the ACER.BIN file in entirety (header, pubkey, marker) compresses to 415 bytes, the marker alone compresses to 246 bytes.

Module Manipulation (EFI)

There are two methods to replace/insert/delete modules.

1. During a 'mod'

Ensure the 'Allow user to modify other modules' option is checked.
When the dialog box prompts modification do the following:

- Delete module – delete the relevant .ROM file from the DUMP directory
- Replace module – edit/replace the relevant .ROM file in the DUMP directory
- Insert module – place the new file in the DUMP directory. If it has a .ROMU extension then it is inserted uncompressed. If it has a .ROM extension then it is compressed using the prevelant method in the EFI.
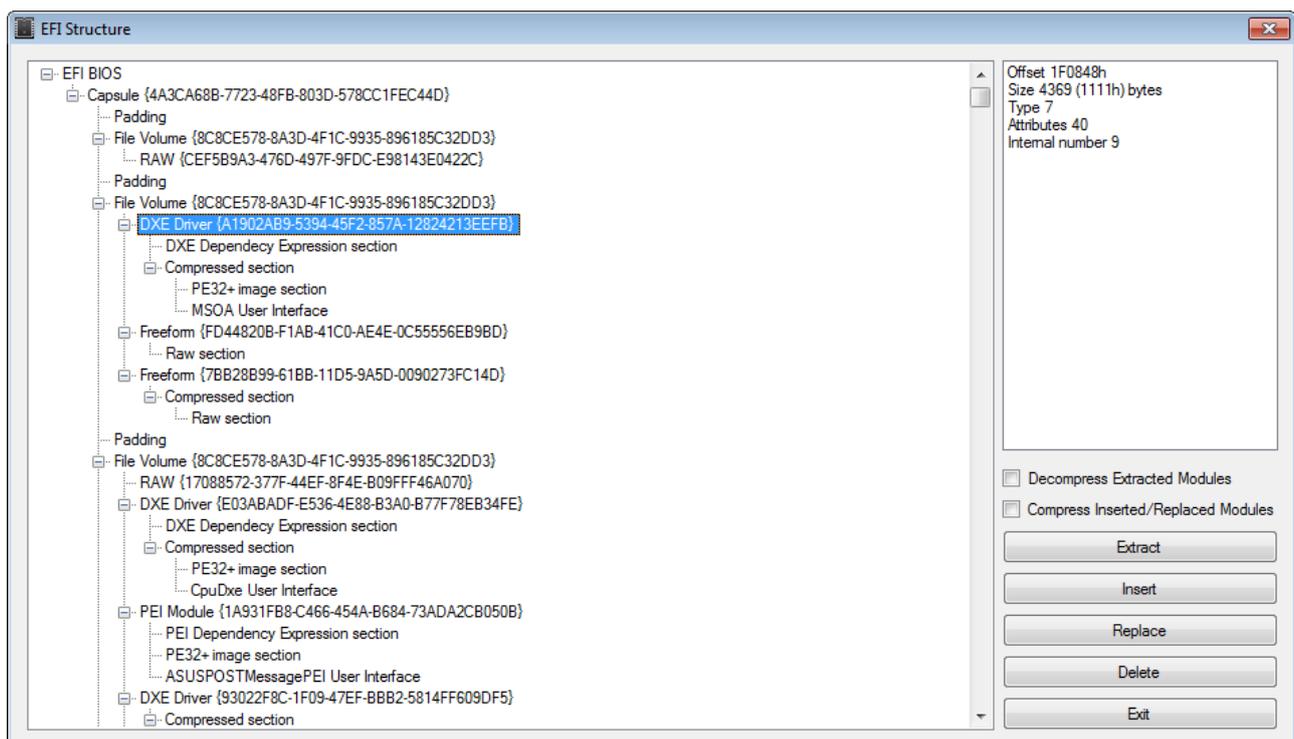
2. Using the tree view

After the EFI has loaded into the tool click the Structure button.
You can then do the following:

- Extract – highlight the relevant module and click Extract. If 'Decompress Extracted Modules' is ticked then then the module is decompressed (if compressed), otherwise it is extracted as is.
- Replace module – highlight the relevant module then click Replace. A dialog box then prompts selection of a replacement .ROM. It will be compressed as per the 'Compress Inserted/Replaced Modules' switch.
- Insert – Highlight a module in the relevant FV and click Insert. A dialog box then prompts selection of a replacement .ROM. The new ROM is placed at the end of the FV, compressed as per the 'Compress Inserted/Replaced Modules' switch.
- Delete – Highlight a module then click Delete.

The modifications are saved to a file with _MOD appended to the filename (eg. EFI.ROM becomes EFI_MOD.ROM).

Advanced Mode

To enable advanced mode create a file called PhoenixTool.ini in the tool directory. It can be blank

** This enables some modification methods that are potentially not safe, USE AT YOUR OWN RISK **

It also allows all mods (except some Dell) to occur without a RW report.

I suggest only experienced modders enable this mode.

It also enables batch mode (see below).

Using Batch Mode in the SLIC Tools

There are 3 checkboxes on the main window of each tool. They are only visible if the tool is in 'Advanced' mode.

Batch   Enables batch mode (see below for description of usage)
Upload   The tool will automatically log on to the server and upload a compressed (ZIP) archive containing the original and compressed BIOSes and any user specified files (see below)
Directory The tool will prompt the user for a directory name (case insensitive) where it will place the compressed archive on the server, creating the directory if necessary. It will do this for each BIOS.

Note that Upload and Directory will work independently of batch mode.

Directory structure for batch mode

In the base directory there should be:
- A .TXT file containing any SLP strings (not for Phoenix).
- A .BIN file containing the SLIC

These will be applied to all the BIOSes.

Each BIOS should be in it's own subdirectory:
- For Phoenix mods there can be a RW Everything report (.RW) in the subdirectory with the BIOS
- If the BIOS file has the extension .EXE, .ZIP or .RAR then an attempt to decompress it with 7-Zip is made, and the contents processed as below.
- A file is only considered a BIOS image if has the extension .BIN, .ROM, .WPH, .EXE or .BIO or if it's size is a multiple of 512K.

Any other files to be included in the compressed archive should be placed in the INCLUDE subdirectory in the base directory. They will be included in every archive. For single (non-batch) mods then the INCLUDE directory should be a subdirectory of the directory containing the BIOS image.

Upload

The INI file for the tool should contain a [FTP] section detailing the server, username and password. An SSL connection is made automatically.

[FTP]
Server="mods.myftp.org"
Username="username"
Password="password"