Name

PatchFv.py – The python script that patches the firmware volumes (Fv) with in the flash device (FD) file post FSP build.

Synopsis

```
PatchFv FvBuildDir [FvFileBaseNames:]FdFileBaseNameToPatch [“Offset, Value”]+

| [“Offset, Value, @Comment”]+

| [“Offset, Value, $Command”]+

| [“Offset, Value, $Command, @Comment”]+
```

Description

The PatchFv.py tool allows the developer to fix up FD images to follow the FSP design specification. It also makes the FD image relocatable. The tool is written in Python and uses Python 2.7 or later to run. Consider using the tool in a build script.

FvBuildDir (Argument 1)

This is the first argument that PatchFv.py requires. It is the build directory for all firmware volumes created during the FSP build. The path must be either an absolute path or a relevant path, relevant to the top level of the FSP tree.

Example usage:

```
Build\YouPlatformFspPkg\%BD_TARGET%_%VS_VERSION%%VS_X86%\FV
```

The example used contains Windows batch script %VARIABLES%.

FvFileBaseNames (Argument 2: Optional Part 1)

The firmware volume file base names (FvFileBaseNames) are the independent Fv’s that are to be patched within the FD. (0 or more in the form “FVFILEBASENAME:”) The colon “:” is used for delimiting the single argument and must be appended to the end of each (FvFileBaseNames).

Example usage:

```
STAGE1:STAGE2:MANIFEST:YOURPLATFORM
```

In the example STAGE1 is STAGE1.Fv in YOURPLATFORM.fd.

FdFileNameToPatch (Argument 2: Mandatory Part 2)

Firmware device file name to patch (FdFileNameToPatch) is the base name of the FD file that is to be patched. (1 only, in the form “YOURPLATFORM”)

Example usage:

```
STAGE1:STAGE2:MANIFEST:YOURPLATFORM
```

In the example YOURPLATFORM is from YOURPLATFORM.fd

“Offset, Value[, Command][, Comment]” (Argument 3)

The “Offset” can be a positive or negative number and represents where the “Value” to be patched is located within the FD. The “Value” is what will be written at the given “Offset” in the FD. Constants may be used for both offsets and values. Also, this argument handles expressions for both offsets and values using these operators:

= - * & | ~ ( ) [ ] { } < >

The entire argument includes the quote marks like in the example argument below:

```
"0xFFFFFFC0, SomeCore:__EntryPoint - [0x000000F0],@SomeCore Entry"
```

Constants:

Hexadecimal (use “0x” as prefix) | Decimal

Examples:

Positive Hex	Negative Hex	Positive Decimal	Negative Decimal
0x000000BC	0xFFFFFFA2	188	-94

ModuleName:FunctionName | ModuleName:GlobalVariableName

ModuleGuid:Offset

Operators:

+ Addition.

- Subtraction.

* Multiplication.

& Logical and.

| Logical or.

~ Complement.

( ) Evaluation control.

[ ] Get a DWord value at the specified offset expression from [expr].

{ } Convert an offset {expr} into an absolute address (FSP_BASE + expr).

< > Convert absolute address <expr> into an image offset (expr & FSP_SIZE).

Special Commands:

Special commands must use the $ symbol as a prefix to the command itself. There is one command available at this time.

$COPY – Copy a binary block from source to destination.

Example:

```
"0x94, [PlatformInit:__gPcd_BinPatch_FvRecOffset] + 0x94, [0x98],
$COPY, @Sync up 2nd FSP Header"
```

Comments:

Comments are allowed in the “Offset, Value [, Comment]” argument. Comments must use the @ symbol as a prefix. The comment will output to the build window upon successful completion of patching along with the offset and value data.

License