

Porting Lib Functions

Porting Lib Functions

by [Edgpaez](#) on 22 Dec 2008 01:32

Good day Developers :kool:

Usually patch porters start his way by porting Lib functions, this porting is very alike the patch porting in the redirection part, besides updating you lib so all the elfs can work correctly you'll acquire knowledge to become a Patch Developer, so let's start:

Materials:

- > IDA
- > Entrypoint Converter
- > Note Pad
- > 512 Mb (or more) of RAM Memory.
- > Firmware of the phone with the needed functions in Lib (we'll call it **A**)
- > Firmware of the phone which Lib is going to be updated (we'll call it **B**)

In this example we'll port 312 Func from W850 R1KG001 to W610 R6BC002.

Code: [Select all](#)

```
W850
0C48: 00000000 E9D14045 ; 312: void StringInput_Dispatch_SetText(DISPATCH *,int StrID);
```

Code: [Select all](#)

```
W610
0C48: XXXXXXXX YYYYYYYY ; 312: void StringInput_Dispatch_SetText(DISPATCH *,int StrID);
```

Download Entrypoint Converter

[url=http://rapidshare.com/files/152893575/Entrypoint_Converter_1.1.rar]here[/url].

First of all you have to open the Firm of both phones (the one with Lib updated and the one with Lib Out of date) with IDA:

[url=http://se developers.oxyhost.com/showthread.php?tid=147]Open Firmware with IDA[/url]

Open Note Pad (Note Pad ++ is so much better ;)) and write the phone model of **A** and after the functions to be ported.

after some lines you give some space to see the function that is being ported. And at the end a space to see the ported functions, as you see here:

[non-available image posted]

Now we're free to go:

1. Open Entrypoint Converter, then copy the bytes in green in last pic (**E9D14045**), and paste it Full Offset space like this:

[non-available image posted]

in Entrypoint (byteswapped) the value is going to change for something starting at 44 or 45 copy that value (Ctrl + C) (In red in the Pic)

2. After that go to **A** FW in IDA and press 'G', a box like this is going to appear, paste in here the last copied value, and press OK:

[non-available image posted]

3. IDA will jump to the desired address, in this address is the wanted function, here you must press 'C' and code will be analyzed.
if this address ends in 1,3,5,7,9,B,D or F and you press 'C' nothing is going to happen so you have to press down key (down a line, or to next offset) and press again 'C'. then the code will be analyzed and you'll see something like this:

[non-available image posted]

If you made any mistake, press 'U' and then OK and code will be Un Analyzed.

4. Then you have to open **B** FW with Smelter ([url=http://sedevelopers.oxyhost.com/showthread.php?tid=139]here[/url]).

leave IDA in the background and press B in Smelter to Search byte patterns (as in[url=http://sedevelopers.oxyhost.com/showthread.php?tid=139]tutorial[/url]) here, you have to start copying all the bytes (starting for two bytes before the start of function (E2B5)) for any instruction (ADD, BEQ, LDR, CMP, etc) **except** for those that include sub_XXXXXX, loc_XXXXXX or dword_XXXXXX (like in 4540D1EC and 4540D1F0 offsets) instead of copying the bytes, we'll copy '??' (the symbol of interrogation) for each byte, and then press OK.

[non-available image posted]

If it appears more than one coincidence you'll have to increase the number of bytes copied (including '??') until you get only ONE coincidence. If you have problems (like not finding anything or Smelter not searching) you can, search in another function part (like the end) in the same way, not all functions are that easy to get ported.

After you get one coincidence press Right click in it, then List > and Copy to clipboard "Bytes" (or press F5) like this:

[non-available image posted]

5. Now open **B** in IDA and press Alt + B, the paste (Ctrl + V) the copied value, make sure it is configured like this and press OK:

[non-available image posted]

IDA will jump to the address with those bytes, and you'll see something like this:

[non-available image posted]

Then click the offset where the function start (guide yourself by looking at **A** FW in IDA) in this case in 452E3442 (usually after B5) and press 'C' and the code will be analyzed. Getting something like this:

[non-available image posted]

Here you have to compare if functions are the same, look at bytes after and before the function and make sure they're the same, as well as in function body looking at the instructions and the bytes. Only practice will give you the knowledge to know wether it is well or bad ported. If functions aren't the same then jump back to step 4 and copy again the bytes, maybe you made a mistake.

6. When your completely sure those are the same functions the you have to copy the offset, and paste it in Entrypoint Converter, here is where most people gets confused and write and incorrect address, if in step 3 you had to down a line because your function ended in 1,3,5,7,9,B,D or F (in **A** FW in IDA) then you have to go one line/offset **UP** of the start of function and copy that. but if in step 3 you didn't changed the offset by going up or down then you copy the real start of function offset. In this case our function ended in 1,3,5,7,9,B,D or F (like most of it), so we had to down a line in step

3, so we'll copy the offset Up the offset of the function start, it means we'll copy 452E3441. We re open Entrypoint Converter and paste 452E3441 in Full Offset, and copy the value in Entrypoint (byteswapped).

[non-available image posted]

7. That's it!!

Now we have to open our Note pad and change the original value, for the one copied from Entrypoint (byteswapped). and put the function in the space of Ported functions 😊

[non-available image posted]

so what we get is:

Code: [Select all](#)

```
W610
0C48: 00000000 41342E45 ; 312: void StringInput_Dispatch_SetText(DISPATCH *,int
StrID);
```

Keep porting so you can get expert at it :kool: I recommend to port already ported functions so you can see and prove you wrong and find yourself the mistakes, when you get good the you'll publish and update Libs yourself. Also at the beginning port functions from DB2020 to DB2020 or DB2010 to DB2010, porting between DB's is very difficult and needs pretty much knowledge. I hope this Guide helps you understanding and entering the VKP patches world, feel free to post all your doubts and problems.

(c) Edgpaez

Best wishes,
SE Developers

by [Stonos](#) on 01 Jan 2009 14:17

Nice tutorial, but I think that it is quicker to use PATSearch ;)
Diezil has written a tutorial [here](#)

I have also written a similar program to Entrypoint converter which will also do the jump for you (sounds useless, but it can save some time in the long run :P) (available [here](#), screenshot in [post #181](#).

w. junior has also written another tutorial [here](#) for porting functions (and 8xxx functions).

I have also made a small program that automatically ports libraries :) ([click](#))

by [Vitor_Boss®](#) on 03 May 2010 02:34

Can i make a suggestion?

Changes the first post.

on 4.:

First configure offset on smelter, Flash>Base>Enter the address.

To create a pattern use only the function id, like:

Code: [Select all](#)

```
E2 B5
05 1C
05 D0
68 68
E7 49
88 42
```

Use: ??B5??1C??????68??49??42 OR ??B5??1C??D0??68??49??42. If not found anything remove some ID's and put "???" on the place, ever start by LDR (68,78,98 ...), LDR(48...4D) , CMP and Bxx functions ID.

Justification: Some FW don't use same values for same functions.

Press F3 on list to get the offset address, then go to IDA and press G, past from clipboard and press OK.