http://forums.se-nse.net/topic/51763-elfpack-compiling/

Elfpack Compiling

Started by Timod, Aug 13 2010 09:29 AM

Compiling Elfpacks is a horrible thing to do, right?

Nah, it's not so bad.
There is a good tutorial for it.
Only problem, it's in Russian and the links are quite outdated.
We also have one over here, on SE-NSE, but that's even more outdated, and incomplete.
That's why I've decided to write a tutorial myself.

First things first: **What tools do I need**
• TortoiseSVN (Link) *Just install it like you would install any program*
• IAR Embedded workbench (link) thanks to Boysie. Unpack everything in

```
C:\Program Files\Embedded Workbench 4.0 Evaluation
```

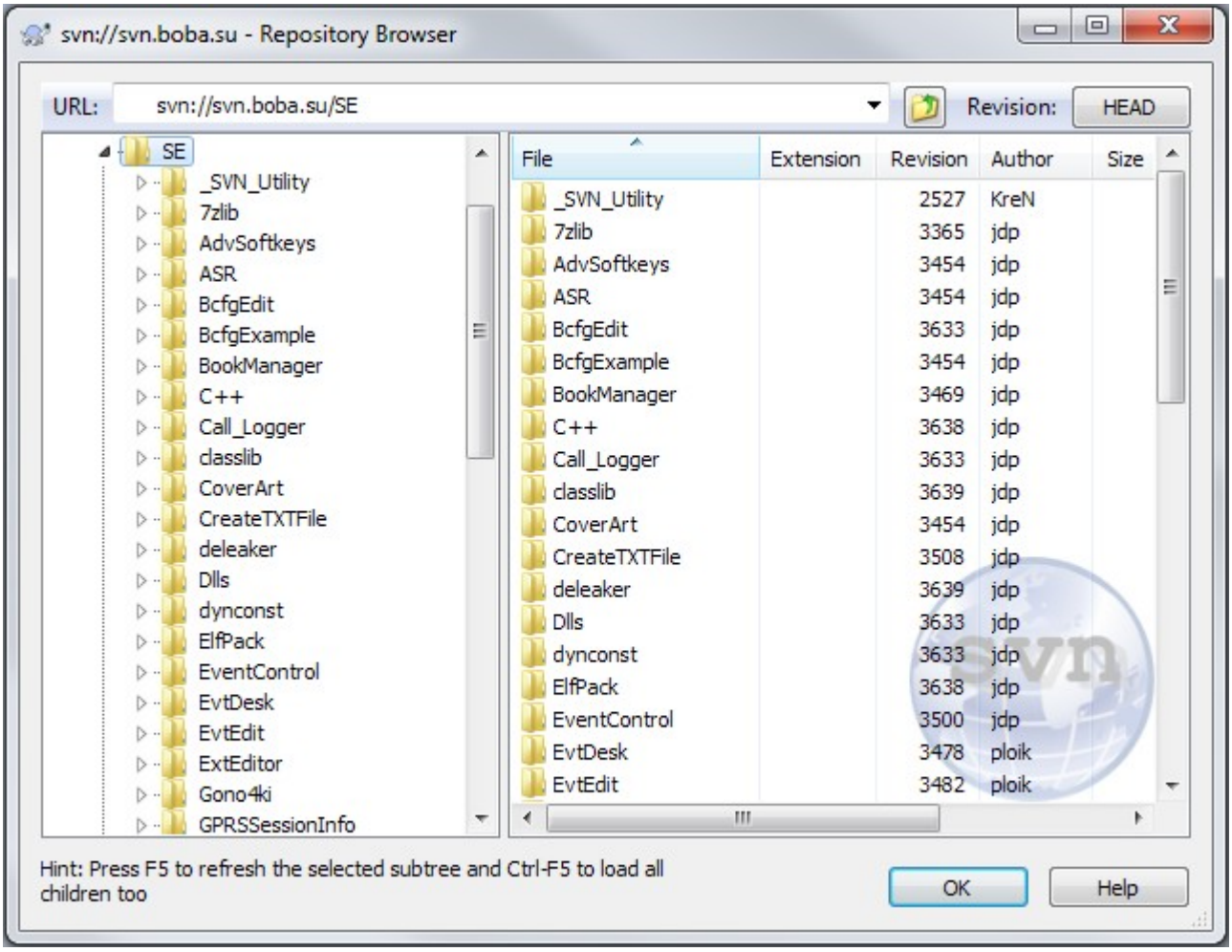
Now we've got those things covered, let's **get the files we need for compilation**.
Easy things first:
• Your phone's .mbn, .raw, phone_app.cxc and when available phone_emp_app.cxc.*

(* Obviously if you're compiling for DB3150 and lower then phone_emp_app.vkp isn't needed.)
It is important that db3xxx obtain these files using Read3150 v0.7.

```
http://justdanpo.cheb.ru/data/projects/gsm/read3150_0.7.rar
```

Next, we **go to svn://svn.boba.su/SE** and get the workspace files.
It'll ask for a username and password.
Username: guest
Password: guest
Don't forget to check the 'save authentication' box.
[non-available image posted]

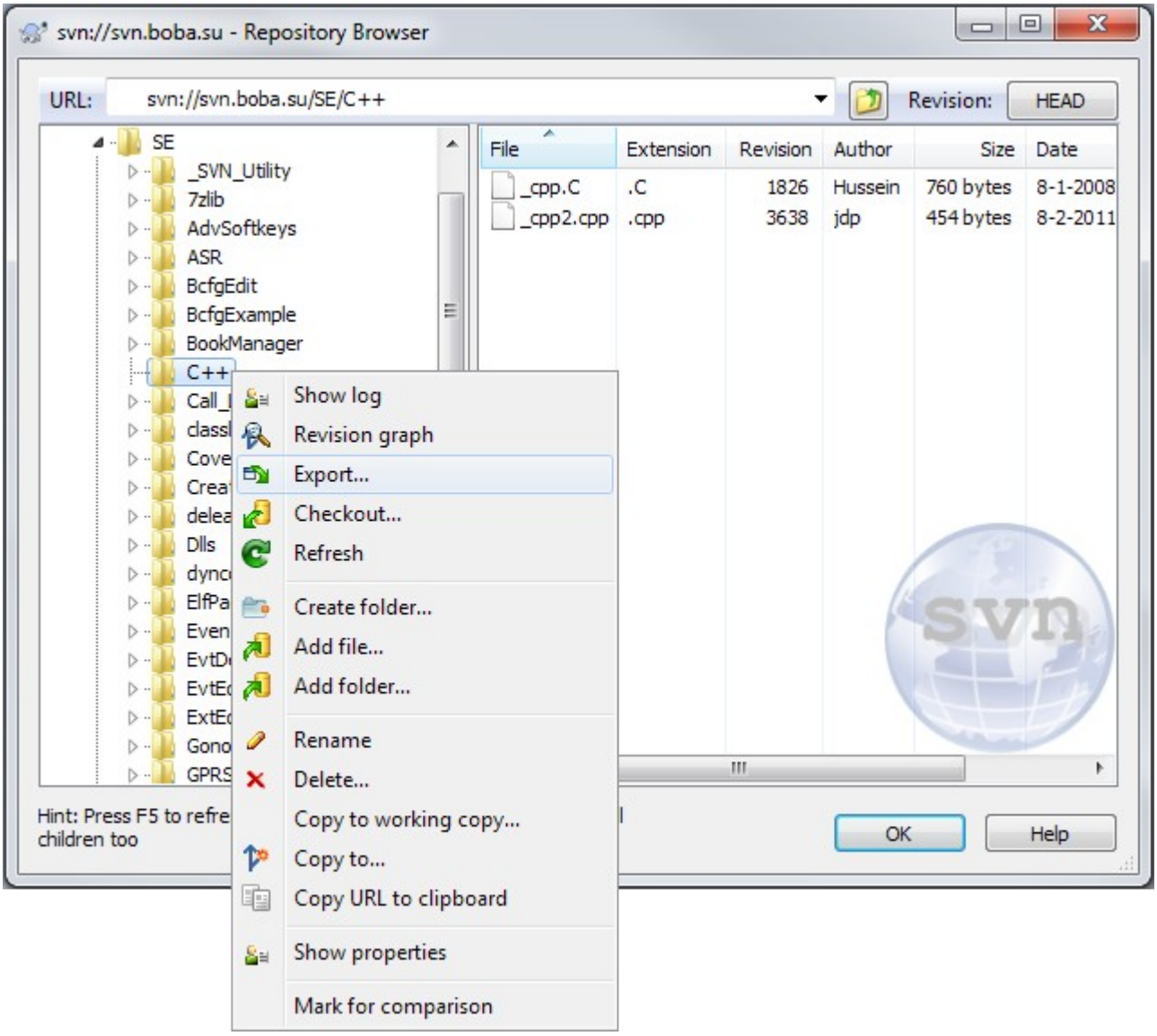
If all is correct, you'll get this screen



We need 4 of these files:

- C++
- ElfPack
- include
- Dlls

Download them by right clicking on them and use Export

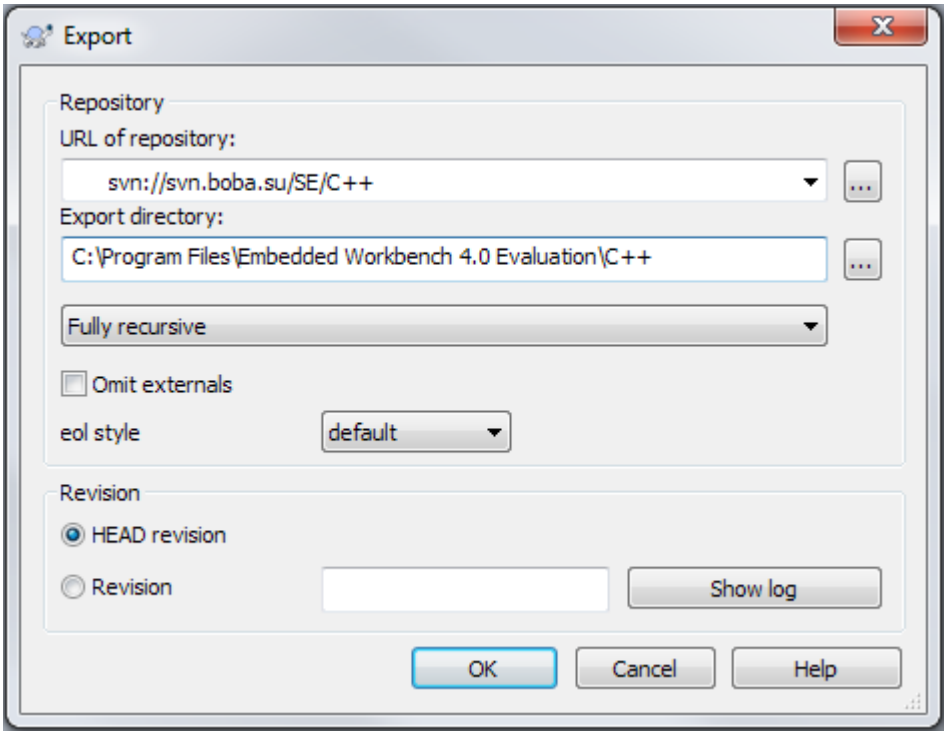


Save them on a location that would seem suitable for you.
How about?

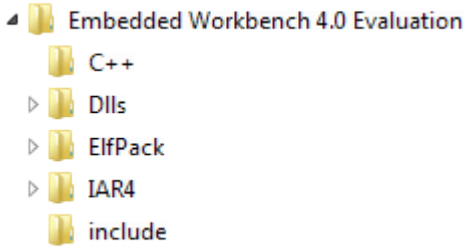
```
C:\Program Files\Embedded Workbench 4.0 Evaluation
```

So where it says "Export directory", we enter (for C++):

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\C++
```



Do the same thing for 'Elfpack', 'include' and 'Dlls'
Now we should have something like this:



Next up, we're going to **open IAR Embedded Workbench**
If you followed the guide, you can copy-paste this directly into explorer to open it. If not, you'll have to find it yourself.

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\IAR4\Embedded Workbench 4.0 Evaluation\common\bin\IarIdePM.exe
```

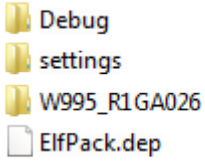
Now, the program will ask us what to do. **Choose the option: Open Existing Workspace**.
The workspace we're looking for is called ElfPack.eww and can be found here:

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\Elfpack
```

Now, if everything went fine, you should be able to see this:
[non-available image posted]
Next, **go to Project -> Edit configuration**
[non-available image posted]

Choose your phone model + firmware (for the purpose of this guide, I'll compile W995_R1GA026)
[non-available image posted]
Next, **choose Project -> Rebuild all**
[non-available image posted]

Now if everything went correct, we've got some new stuff in the ElfPack folder, so go there:
You should now see the new folders:
Debug
Settings
W995_R1GA026 (or your own phone+firmware)
and the file:
ElfPack.dep



Now, go to folder PHONENAME_FIRMWARE\exe. There should be **a file named elfpack.elf**!
We're not done yet, so stick with me.
Create a folder named _Utils in your Elfpack folder.

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\ElfPack\_Utils
```

Place in there the ElfPack.elf file, your phone_app.cxc and phone_emp_app.cxc. (when available)
Now, we need to **get elf2vkpex.exe**. Get that from here or here.
Finally, **create a file named make.bat** in there.
Open make.bat with notepad and in it, you **paste the following** (whatever applies to your platform.)

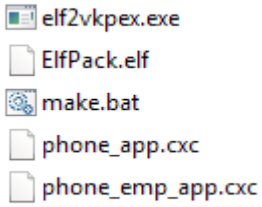
```
elf2vkpex -eELFPACK.ELF -fPHONE_EMP_APP.CXC -fPHONE_APP.CXC

elf2vkpex -eELFPACK.ELF -fPHONE_APP.CXC

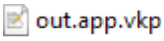
elf2vkpex -fFILE.MBN -eELFPACK.ELF

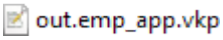
elf2vkpex -fFILE.RAW -eELFPACK.ELF -v
```

When done, your folder should look something like this



Once you're sure you've got the same files as above, **double click make.bat** and see, we've got two new files.





And **we're done**, it's as simple as that.
If you want to publish it, be sure to paste the following lines in the beginning of your patch

```
;PHONE_FIRMWARE
;ELFPack v1.999 (c) Hussein
;SWI_Hook (c) Rst7/CBSIE
;ElfLoader (c)2007 by Rst7/CBSIE, BoBa, Kren
```

So now we're done.
All we now need is the latest library. You could get it here or here, but it is of course way cooler to just get them from svn://svn.boba.su/SE/Library using tortoise.

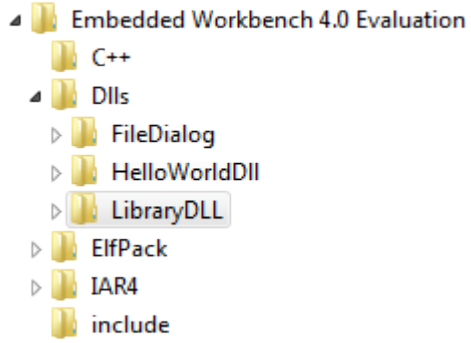
We also need our dyn_const.bin and LibraryDLL.dll.
You could just get them here

```
http://justdanpo.cheb.ru/index.php?link=L3N1cHBvcnQvZWxmcGFjaw==
```

but you can also easily create them yourself.

First up, **LibraryDLL.dll**
You might, or might not know it, but you've already downloaded all the necesary tools. They're all in the Dlls folder.

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\Dlls\LibraryDLL
```



We see a file there named 'LibraryDLL.eww'.
Do you still recognize the .eww extenstion?
It's the IAR workbench extension, so open up IAR.

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\IAR4\Embedded Workbench 4.0
Evaluation\common\bin\IarIdePM.exe
```

Choose 'open existing workspace', go to the 'LibraryDLL folder' and choose 'LibraryDLL.eww'
Choose your own phone + firmware via Project -> Edit Configurations and 'rebuild all'.
If everything is correct you'll find your 'LibraryDLL.dll' in

```
C:\Program Files\Embedded Workbench 4.0
Evaluation\Dlls\LibraryDLL\phone_firmware\Exe
```

Creating dyn_const.bin is very similar, only difference being the fact that we still need to download the necessary files.
To do so, go to svn://svn.boba.su/SE and export the folder called dynconst to

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\dynconst
```

Open IAR again:

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\IAR4\Embedded Workbench 4.0
Evaluation\common\bin\IarIdePM.exe
```

Open the file 'dynconst.eww' in the 'dynconst folder'.
[non-available image posted]

Now here is the tricky part.
When choosing phone + firmware, you'll notice that not all phones are listed.
If your phone is not listed (like my W995), pick one that is from 'the same generation'.
For me, this would be a db3210 and a walkman phone (like W705, W715), so I pick one of those.
To know which one to pick look below the spoiler to find a full list of all phones (as of 16-8-2010)
Spoiler
Once picked, click rebuild all and you will find your dyn_const.bin in

```
C:\Program Files\Embedded Workbench 4.0 Evaluation\dynconst\phone_firmware\Exe
```

That's it. We're done. Compress your Elfpack using winrar or winzip, and upload it in the elfpack topic, for other users to use.

I'd like to say thanks some people for helping me with this guide.
-First off, Boysie. He helped me compile my first ElfPack, answered the most silly questions, and uploaded his IAR. Without him, I wouldn't have been able to do it.
-Next the other people helping me in the A2 Elfpack topic, thanks guys.
-Thomassafca, for being a pain in my ass and pushing me into writing this guide. Thanks.
-Den Po, IronMaster, Husein for releasing Elfpack
-Blacklizard, for being one of the greatest creators/porters out there. (Also for providing Boysie with

the link to IAR, apparantly)
-And last, but certainly not least, awg, over at mobilefree.ru. Without his guide, I would have come nowhere.

Hope you'll enjoy using it. Any questions can be posted here.

Edit:
Notes and advanced tips:
- phone_emp_app.vkp is obviously not needed for db3150 (quote)
- How to remove / modify elf icon. *By Boysie* (Link)
- Or, a more sofisticated way to remove / modify elf icon. *By NKLS* (Link)
- Expanding phone_app.cxc and phone_emp_app.cxc *By Crazysharath / Maxidimov* (Link)
- Advanced make.bat with auto rename function. *By Tobi@s* (Link)
- New Elf2VKP *By Den Po. Thanks to Crazysharath* (Link)

Boysie

Posted 13 August 2010 – 09:49 AM

A little tip: To remove the annoying elf icon, remove these lines from **regfile.c** and compile.

```
#ifdef EP_EXTICON14x14
#include "exticons\14x14.inc"
#endif
#ifdef EP_EXTICON16x20
#include "exticons\16x20.inc"
#endif
#ifdef EP_EXTICON24x24
#include "exticons\24x24.inc"
#endif
#ifdef EP_EXTICON42x32
#include "exticons\42x32.inc"
#endif
#ifdef EP_EXTICON48x36
#include "exticons\48x36.inc"
#endif
#ifdef EP_EXTICON56x42
#include "exticons\56x42.inc"
#endif
```

But what if we want to change the elf icons?

Then you compile ep without the embedded icons as above and change the animal icon it uses by patch.

Elfloader has embedded icons in the patch if compiled without removing the lines I stated above. These

cannot be changed by GFX patch.

Removing those lines means you can replace the icon by GFX patch by replacing "CALE_LUNAR_12ANIMALS_1ST_MOUSE_ICN" (Small) and "CALE_LUNAR_12ANIMALS_10TH_ROOSTER_ICN" (Large) with IMGTool.

---

nkls

Posted 13 August 2010 – 02:20 PM

Definitely not a sophisticated way to do it Boysie! :P

The better way is to go to:
Project > Options... > C/C++ Compiler > Preprocessor (tab)
and remove the EP_EXTICON??x?? from Defined symbols. :)

> But what if we want to change the elf icons?
>
> Then you compile ep without the embedded icons as above and change the animal icon it uses by patch.

Or, open the icon you want to use in HxD (hex editor, others might work too), select File > Export > C and save it somewhere.
Open the file you saved and open ElfPack\exticons*.inc in a text editor, and replace the data "body" (what's between the {}) in *.inc with the one from your file. B-)