
Subject: Remove unused languages
Posted by [crydev](#) on Mon, 06 Jan 2014 11:00:27 GMT
[View Forum Message](#) <> [Reply to Message](#)

Hello,

When I browse through my executable in the disassembler I see a lot of unused languages being compiled into the application. English is sufficient for me, Dutch, Polish, Swedish, etc. are not necessary and in my situation only waste space in the application.

Is there a way to disable or remove them in a way that they are not linked anymore?

Regards,
crydev

Subject: Re: Remove unused languages
Posted by [mirek](#) on Mon, 06 Jan 2014 14:47:09 GMT
[View Forum Message](#) <> [Reply to Message](#)

crydev wrote on Mon, 06 January 2014 06:00Hello,

When I browse through my executable in the disassembler I see a lot of unused languages being compiled into the application. English is sufficient for me, Dutch, Polish, Swedish, etc. are not necessary and in my situation only waste space in the application.

Is there a way to disable or remove them in a way that they are not linked anymore?

Regards,
crydev

That is an interesting request...

Is it worth it? (Can you check?). If it accounts for more than 50KB for console application, I guess we could consider option or compilation flag to leave only EN-US translations...

Mirek

Subject: Re: Remove unused languages
Posted by [crydev](#) on Mon, 06 Jan 2014 15:20:06 GMT
[View Forum Message](#) <> [Reply to Message](#)

I did some basic calculations, based on the information I found in the disassembler and what I found in files: Core.t and LangInfo.cpp.

The image below shows a few example strings. They are located in Core.t and I assume they are

translations for an error or information message in multiple languages. However, in the disassembler you can see that they are all compiled into the executable. I did not measure this, but an educated guess on Core.t, which is 44 kb (~ 40 kb with margins of non-trivial information) is linked into the executable. If just the english translations are linked into the executable, the size could be reduced by roughly: average string length divided by the number of languages built into U++.

The image below shows the language array in LangInfo.cpp. The big rectangle shows the strings that are linked, which are a hell of a lot and they shouldn't be necessary.

The small rectangle shows the fact that some strings even indicate only very small differences. The compilation loaded into IDA Pro is built in VC10 Optimal with String Pooling enabled. Even if these strings could not be eliminated, it would be nice if they could be concatenated in order to remove the redundancy that is generated.

If you need any more information, please tell me.

Regards,
crydev

File Attachments

- 1) [IDA Strings.PNG](#), downloaded 276 times
 - 2) [Langinfo.PNG](#), downloaded 259 times
-

Subject: Re: Remove unused languages
Posted by [mirek](#) on Mon, 06 Jan 2014 16:45:37 GMT
[View Forum Message](#) <> [Reply to Message](#)

OK, IMO it makes sense to at least add RM issue about this. Alternatively, we could consider also consider compression (zlib) - but that might be hard(er) to achieve.

Subject: Re: Remove unused languages
Posted by [crydev](#) on Mon, 06 Jan 2014 22:24:42 GMT
[View Forum Message](#) <> [Reply to Message](#)

mirek wrote on Mon, 06 January 2014 17:45 OK, IMO it makes sense to at least add RM issue about this. Alternatively, we could consider also consider compression (zlib) - but that might be hard(er) to achieve.

That would be very nice. I have been trying to get my hands on the whereabouts of zlib in U++ GUI applications and why it is being used. I am not sure but I think it has something to do with the

iml compression? If so, it has to be researched / tested whether removing zlib actually results in a decrease of image size when images are not compressed anymore and in situations of how many images you should have in order to gain size improvements.

Could you enlighten the purpose of zlib in U++ GUI applications a bit more?

Thanks,
crydev

Subject: Re: Remove unused languages
Posted by [mirek](#) on Tue, 07 Jan 2014 08:29:59 GMT
[View Forum Message](#) <> [Reply to Message](#)

crydev wrote on Mon, 06 January 2014 17:24mirek wrote on Mon, 06 January 2014 17:45OK, IMO it makes sense to at least add RM issue about this. Alternatively, we could consider also consider compression (zlib) - but that might be hard(er) to achieve.

That would be very nice. I have been trying to get my hands on the whereabouts of zlib in U++ GUI applications and why it is being used. I am not sure but I think it has something to do with the iml compression? If so, it has to be researched / tested whether removing zlib actually results in a decrease of image size when images are not compressed anymore and in situations of how many images you should have in order to gain size improvements.

Could you enlighten the purpose of zlib in U++ GUI applications a bit more?

Well, zlib is quite universally used about everywhere fast moderate compression is required. I do not think I will recall all of them, but

- .imls are indeed compressed
- .brc files can be compressed as well
- .tpp files are compressed
- HttpRequest has to use zlib to decompress responses
- plugin/png is using zlib; png is format of choice for X11 clipboard
- Some of RichObject formats use zlib compression
- RichEdit spellchecker is using zlib compression
-

Mirek
