

Forum:[Hack Forums](#)

Thread: [Runtime dependency problems](#)

ApexLead - December 9, 2018, 10:47 pm

Hello,

I followed some solutions that I was given here about how to get rid of the missing vcruntime140.dll and msvcp140.dll errors but it didn't solve my problem.

I changed the linker to /MT for release and /MTd for debug, but when I try it on different windows 10 machines it says the spoken DLLs are missing, Ive searched for a solution everywhere but all I saw was changing the linker to be static, like I did.

If someone can help me solve this issue it will be much appreciated, thanks in advance :)

Cød3ing - December 10, 2018, 6:12 pm

What are you trying to compile? what language is it in? what compiler are you using? what compiler flags have you set? What machine did you compile this on?

ApexLead - December 10, 2018, 11:57 pm

(Dec 10, 2018 07:12 PM)Cød3ing Wrote: [\[CITE\]](#) What are you trying to compile? what language is it in? what compiler are you using? what compiler flags have you set? What machine did you compile this on?

A code I wrote, those are all of the headers & libs:

```
#include "stdafx.h"
#include <iostream>
#include <windows.h>
#include <Shlobj.h>
```

It's written in c++, visual studio 2015, /MT for release and /MTd for debug, compiled on a windows 10 machine

RCE - December 24, 2018, 1:42 pm

Make sure you execute the output file from 'release' into your other machines , if the problem is still then u maybe misconfigured ur settings , try to compile with G++/ mingw via command line and check again

0xDEAD10CC - December 24, 2018, 8:19 pm

(Dec 24, 2018 02:42 PM)Shad0Byte Wrote: [\[CITE\]](#) Make sure you execute the output file from 'release' into your other machines , if the problem is still then u maybe misconfigured ur settings , try to compile with G++/ mingw via command line and check again

He's not using mingw... Not sure what good this will do, the options aren't even the same.

@OP - are you sure you understand how to use the configuration dropdown?

Then you can see visibly the difference of the static linking by looking at the filesizes:

And/or check the imports to make sure that you don't see the dll in the IAT. If you just change the project settings btw, but your code doesn't change, the IDE may not build it, so you'll have to do a

rebuilt instead to force it.

This is easy though, so I'm not sure what your problems would be otherwise.

HiddenEyE - December 24, 2018, 9:12 pm

OP surprise that you have not been able to solve this when the solution is filled up in google. anyway if you select /MT their are libs you should link it with. google for the libs.

0xDEAD10CC - December 24, 2018, 9:21 pm

(Dec 24, 2018 10:12 PM)HiddenEyE Wrote: [\[CITE\]](#) OP surprise that you have not been able to solve this when the solution is filled up in google.

anyway if you select /MT their are libs you should link it with. google for the libs.

If you select /MT you don't have to link it with anything, this option will take care of the linking for you.

ApexLead - December 25, 2018, 2:25 am

(Dec 24, 2018 09:19 PM)0xDEAD10CC Wrote: [\[CITE\]](#) He's not using mingw... Not sure what good this will do, the options aren't even the same.

@OP - are you sure you understand how to use the configuration dropdown?

Then you can see visibly the difference of the static linking by looking at the filesizes:

And/or check the imports to make sure that you don't see the dll in the IAT. If you just change the project settings btw, but your code doesn't change, the IDE may not build it, so you'll have to do a rebuilt instead to force it.

This is easy though, so I'm not sure what your problems would be otherwise.

(Dec 24, 2018 10:12 PM)HiddenEyE Wrote: [\[CITE\]](#) OP surprise that you have not been able to solve this when the solution is filled up in google.

anyway if you select /MT their are libs you should link it with. google for the libs.

Thanks for your replies, but I know how to make /MT and /MTd for release and debug, but for some reason it still shows a runtime dll error on some machines, even if I link statically :/

0xDEAD10CC - December 25, 2018, 11:59 am

(Dec 25, 2018 03:25 AM)ApexLead Wrote: [\[CITE\]](#) Thanks for your replies, but I know how to make /MT and /MTd for release and debug, but for some reason it still shows a runtime dll error on some machines, even if I link statically :/

Then you're doing it wrong. Show me the imports table.

El Lobo - December 25, 2018, 9:05 pm

(Dec 24, 2018 02:42 PM)Shad0Byte Wrote: [\[CITE\]](#) Make sure you execute the output file from 'release' into your other machines , if the problem is still then u maybe misconfigured ur settings , try to compile with G++/ mingw via command line and check again

What a bunch of shit that you wrote. You cant even read properly and yet you are selling hvnc coded in c++.

To the op, you are not using ide properly. Just like this guy above me said, just look at the iat and see if you have dependencies.

HiddenEyE - December 25, 2018, 9:37 pm

am not posting any further. solution given.

ApexLead - December 26, 2018, 1:43 am

(Dec 25, 2018 12:59 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Then you're doing it wrong. Show me the imports table. Sorry for the newbie question, but where can I find it?

(Dec 25, 2018 10:05 PM)El Lobo Wrote: [\[CITE\]](#) What a bunch of shit that you wrote. You cant even read properly and yet you are selling hvnc coded in c++.

To the op, you are not using ide properly. Just like this guy above me said, just look at the iat and see if you have dependencies.
IAT?

(Dec 25, 2018 10:37 PM)HiddenEyE Wrote: [\[CITE\]](#) am not posting any further. solution given.
The static linking solution didn't solve the problem tho..

El Lobo - December 26, 2018, 2:05 am

(Dec 26, 2018 02:43 AM)ApexLead Wrote: [\[CITE\]](#) IAT?

Import address table. Open compiled file with peexplorer and go to imports tab and you will see all dlls that your exe depends on.

ApexLead - December 26, 2018, 7:34 am

(Dec 26, 2018 03:05 AM)El Lobo Wrote: [\[CITE\]](#) Import address table. Open compiled file with peexplorer and go to imports tab and you will see all dlls that your exe depends on.

Opened with PE Explorer, those are the imports:

kernel32.dll

user32.dll

shell32.dll

ole32.dll

api-ms-win-core-path-l1-1-0.dll (On this one I have "Path not found" error)

Hoodie+ - December 26, 2018, 9:15 am

(Dec 25, 2018 10:05 PM)El Lobo Wrote: [\[CITE\]](#) What a bunch of shit that you wrote. You cant even read properly and yet you are selling hvnc coded in c++.

To the op, you are not using ide properly. Just like this guy above me said, just look at the iat and see if you have dependencies.

Good point, lol.

I asked a few question about how he coded the HVNC and he didn't answer them on his thread.
Imagine buying a \$800 bot and it requiring a preinstalled C++ Runtime library ?

(Dec 26, 2018 08:34 AM)ApexLead Wrote: [\[CITE\]](#) Opened with PE Explorer, those are the imports:

kernel32.dll

user32.dll

shell32.dll

ole32.dll

api-ms-win-core-path-l1-1-0.dll (On this one I have "Path not found" error)

api-ms-win-core-path-l1-1-0.dll is a part of [Visual C++ Redistributable for Visual Studio 2012](#) if I'm not mistaking.

Vanilla - December 26, 2018, 1:02 pm

These are cancerous responses. Simply lookup 'static linking visual studio' on Google.

These are what you should see in the differences. Btw, if the issue is on that dll, that almost seems like corruption in the OS somewhere. Try a different machine to verify; quite easy to spin up a VM on Hyper-V.

<https://answers.microsoft.com/en-us/windows/forum/all/api-ms-win-core-path-l1-1-1ddl-is-missing/b7afd65f-b4f6-4868-8433-587a543532e5>

ApexLead - December 27, 2018, 12:38 am

(Dec 26, 2018 10:15 AM)Hoodie+ Wrote: [\[CITE\]](#) Good point, lol.

I asked a few question about how he coded the HVNC and he didn't answer them on his thread. Imagine buying a \$800 bot and it requiring a preinstalled C++ Runtime library ?

api-ms-win-core-path-l1-1-0.dll is a part of [Visual C++ Redistributable for Visual Studio 2012](#) if I'm not mistaking.

(Dec 26, 2018 04:04 PM)0xDEAD10CC Wrote: [\[CITE\]](#)

These are what you should see in the differences. Btw, if the issue is on that dll, that almost seems like corruption in the OS somewhere. Try a different machine to verify; quite easy to spin up a VM on Hyper-V.

<https://answers.microsoft.com/en-us/windows/forum/all/api-ms-win-core-path-l1-1-1ddl-is-missing/b7afd65f-b4f6-4868-8433-587a543532e5>

This is the DLL that is missing when I try running the static linked EXE on different machines, it works on the original machine although it says "Path not found" on PE Explorer.

Are you saying I should compile the EXE somewhere else?

Hoodie+ - December 27, 2018, 3:34 am

(Dec 27, 2018 01:38 AM)ApexLead Wrote: [\[CITE\]](#) This is the DLL that is missing when I try running the static linked EXE on different machines, it works on the original machine although it says "Path not found" on PE Explorer.

Are you saying I should compile the EXE somewhere else?

No dude.

You have a few options.

1. While sending your executable also include the dll with it.
2. Ask the user to install the required C++ framework
3. Statically link the library.

Since you're most likely developing malware, the 3rd options would be the best pick for you.

(If you were to develop quality malware, you wouldn't even use the fucking CRT...)

ApexLead - December 27, 2018, 5:39 am

(Dec 27, 2018 04:34 AM)Hoodie+ Wrote: [\[CITE\]](#) No dude.

You have a few options.

1. While sending your executable also include the dll with it.
2. Ask the user to install the required C++ framework
3. Statically link the library.

Since you're most likely developing malware, the 3rd options would be the best pick for you.

(If you were to develop quality malware, you wouldn't even use the fucking CRT...)

I do not develop any malware, just playing around with different languages and trying new things. I already statically linked the dlls but for some reason still get the error, so I don't really understand how to solve this problem.

And what do you mean by "wouldn't use the CRT"? Explanations might help me a bit more than "wouldn't use the fucking CRT" ?

El Lobo - December 27, 2018, 5:47 am

(Dec 27, 2018 06:39 AM)ApexLead Wrote: [\[CITE\]](#) I do not develop any malware, just playing around with different languages and trying new things.

I already statically linked the dlls but for some reason still get the error, so I don't really understand how to solve this problem.

And what do you mean by "wouldn't use the CRT"? Explanations might help me a bit more than "wouldn't use the fucking CRT" ?

What else have you statically linked? My guess is that its not some weird compiler/os issue and you are doing something wrong with the settings dont know how to properly use VS. My advice is to read more about the subject and get yourself more familiar with visual studio.

About why is bad using crt in the malware is that it just bloats your final exe and aids in reversing.

ApexLead - December 27, 2018, 7:58 am

(Dec 27, 2018 06:47 AM)El Lobo Wrote: [\[CITE\]](#) What else have you statically linked? My guess is that its not some weird compiler/os issue and you are doing something wrong with the settings dont know how to properly use VS. My advice is to read more about the subject and get yourself more familiar with visual studio.

About why is bad using crt in the malware is that it just bloats your final exe and aids in reversing.

I read a lot about the subject and statically linked the dependencies using the option /MT, which means I'm not really suppose to get missing DLL errors (as far as I know) but I still get them.

How can I remove the CRT and still be able to run the EXE as a standalone?

Dem?n - December 27, 2018, 9:08 am

(Dec 27, 2018 08:58 AM)ApexLead Wrote: [\[CITE\]](#) I read a lot about the subject and statically linked the dependencies using the option /MT, which means I'm not really suppose to get missing DLL errors (as far as I know) but I still get them.

How can I remove the CRT and still be able to run the EXE as a standalone?

Instead of static linking it, dynamically load it. First, open a handle to the dll you care about. If it's shell32.dll for example use LoadLibrary, then use GetProcAddress with the handle to find the exported function you want to call. You then cast this return value to its function prototype. There's a good example on msdn -> <https://docs.microsoft.com/en-us/cpp/build/getprocaddress?view=vs-2017>

0xDEAD10CC - December 27, 2018, 12:16 pm

(Dec 27, 2018 08:58 AM)ApexLead Wrote: [\[CITE\]](#) I read a lot about the subject and statically linked the dependencies using the option /MT, which means I'm not really suppose to get missing DLL

errors (as far as I know) but I still get them.

How can I remove the CRT and still be able to run the EXE as a standalone?

/MT doesn't mean you don't get missing dll errors, it is just a static linking option for the CRT.

(Dec 27, 2018 10:08 AM)Dem?n Wrote: [CITE](#) Instead of static linking it, dynamically load it. First, open a handle to the dll you care about. If it's shell32.dll for example use LoadLibrary, then use GetProcAddress with the handle to find the exported function you want to call. You then cast this return value to its function prototype. There's a good example on msdn ->

<https://docs.microsoft.com/en-us/cpp/build/getprocaddress?view=vs-2017>.

This does not help at all, you don't even know what the dll he's missing on the remote machine is for.

For all those interested, it appears his issue is with Shlwapi.lib from the Shlwapi.h header he's using. The Shlwapi is deprecated, although no static linking probably either.

[https://msdn.microsoft.com/en-us/library/windows/desktop/bb759845\(v=vs.85\).aspx](https://msdn.microsoft.com/en-us/library/windows/desktop/bb759845(v=vs.85).aspx)

Means it probably doesn't exist on the machine he's trying to use the program on.

If he's trying to call functions like PathRemoveFileSpec, they are deprecated and docs say to update this to PathCchRemoveFileSpec which uses a different library (Pathcch.lib).

api-ms-win-core-path-l1-1-0.dll is not part of the CRT though.