

Forum:[Hack Forums](#)

Thread: [C++ Dependency Explanation](#)

ApexLead - November 28, 2018, 7:11 am

Hello everyone,

As I know C++ is a native language which means it has no dependencies as long as it runs on a windows machine, but whenever I try to run a little code I wrote on different machines than the one I created it on I get a missing DLL error for msvcp140.dll and vcruntime140.dll.

How can I assure my EXE will run on every windows machine without needing to add anything?

Thanks in advance ?

TCP - November 28, 2018, 7:26 am

It's because you don't embed it in the program when you generate the code, C/C++ -> Code Generation -> Runtime Library -> Multithreaded in Release version will solve your problem

HFDone - November 28, 2018, 7:35 am

you using Visual C++ 2015 every version have different runtime
the other os dont have Microsoft Visual C++ 2015 Redistributable installed

ApexLead - November 28, 2018, 7:41 am

(Nov 28, 2018 08:26 AM)Exp500 Wrote: [\[CITE\]](#) It's because you don't embed it in the program when you generate the code, C/C++ -> Code Generation -> Runtime Library -> Multithreaded in Release version will solve your problem

Thanks :)

Is this the only way tho? It pumped my EXE from 12kb to 209kb :/

(Nov 28, 2018 08:35 AM)HFDone Wrote: [\[CITE\]](#) you using Visual C++ 2015 every version have different runtime
the other os dont have Microsoft Visual C++ 2015 Redistributable installed
So how can I make my executable compatible with every windows OS like tool programmers do?

TCP - November 28, 2018, 7:47 am

(Nov 28, 2018 08:41 AM)ApexLead Wrote: [\[CITE\]](#) Thanks :)

Is this the only way tho? It pumped my EXE from 12kb to 209kb :/Either you embed it or load it from an external source

ApexLead - November 28, 2018, 7:49 am

(Nov 28, 2018 08:47 AM)Exp500 Wrote: [\[CITE\]](#) Either you embed it or load it from an external source
What do you mean by external source? Its either I embed it or go back to the old position where I need it available on the PC?

TCP - November 28, 2018, 7:58 am

(Nov 28, 2018 08:49 AM)ApexLead Wrote: [\[CITE\]](#) What do you mean by external source? Its either I embed it or go back to the old position where I need it available on the PC?

It's a runtime module, you need to have it in the application or installed

ApexLead - November 28, 2018, 8:09 am

(Nov 28, 2018 08:58 AM)Exp500 Wrote: [\[CITE\]](#) It's a runtime module, you need to have it in the application or installed
Got you, thanks :)

Vuln - November 28, 2018, 8:41 am

Those are MSVC dependencies, because of Visual Studio's default configuration. But if you want to then you're free to install and use another compiler such as GCC.

ApexLead - November 28, 2018, 11:48 pm

(Nov 28, 2018 09:41 AM)Jailbyte Wrote: [\[CITE\]](#) Those are MSVC dependencies, because of Visual Studio's default configuration. But if you want to then you're free to install and use another compiler such as GCC.

So if ill use GCC the exe will remain like 12kb but without dependencies?

3DES - November 29, 2018, 11:31 am

(Nov 29, 2018 12:48 AM)ApexLead Wrote: [\[CITE\]](#) So if ill use GCC the exe will remain like 12kb but without dependencies?

No.

GCC bloats more than msvc.

Anyway, you are using too many other libraries and shit, to get small stub remove all other libraries and even CRT, use only winapi.

LanMan - November 30, 2018, 7:24 am

Because like other people said, you are linking with the CRT:

- 1: you can load it dynamically (What your 12kb exe does)
- 2: You link it statically (Larger size exe)
- 3: You design your code to work without the CRT and remove it.

If low size and no dependencies are important, remove the CRT

ApexLead - December 2, 2018, 2:20 am

(Nov 29, 2018 12:31 PM)3DES Wrote: [\[CITE\]](#) No.

GCC bloats more than msvc.

Anyway, you are using too many other libraries and shit, to get small stub remove all other libraries and even CRT, use only winapi.

(Nov 30, 2018 08:24 AM)LanMan Wrote: [\[CITE\]](#) Because like other people said, you are linking with the CRT:

- 1: you can load it dynamically (What your 12kb exe does)
- 2: You link it statically (Larger size exe)
- 3: You design your code to work without the CRT and remove it.

If low size and no dependencies are important, remove the CRT

What do you mean by CRT? And how to I remove the urlib and other libs if they are needed for my exe to work?

LanMan - December 2, 2018, 6:31 am

(Dec 2, 2018 03:20 AM)ApexLead Wrote: [\[CITE\]](#) What do you mean by CRT? And how to I remove the urlib and other libs if they are needed for my exe to work?

<https://docs.microsoft.com/en-us/cpp/c-runtime-library/crt-library-features?view=vs-2017>

They are needed if you are using runtime feature otherwise aren't, and with msvc you compile with /NODEFAULTLIB to remove it and all default library

0xDEAD10CC - December 2, 2018, 3:42 pm

(Nov 28, 2018 08:11 AM)ApexLead Wrote: [\[CITE\]](#) Hello everyone,

As I know C++ is a native language which means it has no dependencies as long as it runs on a windows machine, but whenever I try to run a little code I wrote on different machines than the one I created it on I get a missing DLL error for msvcp140.dll and vcruntime140.dll.

How can I assure my EXE will run on every windows machine without needing to add anything?

Thanks in advance ?

Lol, you're heavily misinformed. That first sentence is nowhere near true by itself, but additionally those missing dll issues you're having means you're depending on the C++ redistributables from your project settings. A native language doesn't mean no dependencies... Nevermind how inaccurate the "as long as it runs on a windows machine" statement is.

(Nov 28, 2018 08:41 AM)ApexLead Wrote: [\[CITE\]](#) Thanks :)

Is this the only way tho? It pumped my EXE from 12kb to 209kb :/

So how can I make my executable compatible with every windows OS like tool programmers do? OS compatibility is not a guarantee, which is why some people provide different installers depending on whether you're running Win7 or Win10. The code internally uses the preprocessor so you can write platform or OS specific code for other people to compile in their own environments, otherwise runtime checks are involved since the preprocessor will only solve these issues pre-compile time.

(Nov 28, 2018 08:47 AM)Exp500 Wrote: [\[CITE\]](#) Either you embed it or load it from an external source

Or better yet, don't use it if you care about filesize. You'll have to change the entrypoint and mod a few other things in the way you write your code though because /NODEFAULTLIB means you're excluding access to use a bunch of stuff. There's also a lot of other things you can do such as merge sections of the file to get a smaller size to the point where even UPX creates a larger binary. In most cases, making sure to compile using the default RELEASE build will shrink the output enough for most.

ApexLead - December 3, 2018, 1:09 am

(Dec 2, 2018 04:42 PM)0xDEAD10CC Wrote: [\[CITE\]](#) Lol, you're heavily misinformed. That first sentence is nowhere near true by itself, but additionally those missing dll issues you're having means you're depending on the C++ redistributables from your project settings. A native language doesn't mean no dependencies... Nevermind how inaccurate the "as long as it runs on a windows machine" statement is.

OS compatibility is not a guarantee, which is why some people provide different installers depending on whether you're running Win7 or Win10. The code internally uses the preprocessor so you can write platform or OS specific code for other people to compile in their own environments, otherwise runtime checks are involved since the preprocessor will only solve these issues pre-compile time.

Or better yet, don't use it if you care about filesize. You'll have to change the entrypoint and mod a few other things in the way you write your code though because /NODEFAULTLIB means you're excluding access to use a bunch of stuff. There's also a lot of other things you can do such as merge sections of the file to get a smaller size to the point where even UPX creates a larger binary.

In most cases, making sure to compile using the default RELEASE build will shrink the output enough for most.

(Dec 2, 2018 09:04 PM)Phobophile Wrote: [\[CITE\]](#) You mentioned wanting it to run on all Windows OS. It doesn't look like anyone else mentioned this, but if you want XP compatibility you're going to either want to compile with the XP Toolset in VS and make sure the Subsystem isn't blank in the project settings, or manually go and change the

MajorOperatingSystemVersion/MajorSubsystemVersion in the PE header to 0005 + change the MinorOperatingSystemVersion/MinorSubsystemVersion to 0001. Otherwise it will not run on XP.

Thanks for the answers guys, but I'm confused now and don't know how to proceed. When I write a python code and compile it with pyInstaller I get an output exe that works on Win7 and above (all versions), cant it be achieved with a c++ code? And what method should I use to make it as less dependent as possible? Embed the runtime dlls?