Subject: Using DLI files on Windows
Posted by Novo on Fri, 21 Dec 2012 05:49:38 GMT
View Forum Message <> Reply to Message

It is possible that a DLL exports a name (I can see it in dependency walker) but a call to this function crashes, because, I believe, a stub points to a NULL pointer.

Is there a way in Windows to tell that although function's name is exported it cannot be called?

I have a real world example: OpenCL.dll that supports only OpenCL v.1.1 exports function clRetainDevice, which is defined in OpenCL v.1.2. Call to this function crashes, no wonder why.

TIA

Subject: Re: Using DLI files on Windows
Posted by mirek on Sun, 23 Dec 2012 08:56:24 GMT
View Forum Message <> Reply to Message

Well, in that dli struct, functions are represented as member variable pointers, so you can check the existence before making call.

In any case, I do not think that mere NOP instead of missing function is a good idea in most places (say the function is supposed to return a handle that you use downstream in your code, it would cause havoc).

Mirek

Subject: Re: Using DLI files on Windows
Posted by Novo on Sun, 23 Dec 2012 13:46:22 GMT
View Forum Message <> Reply to Message

mirek wrote on Sun, 23 December 2012 03:56Well, in that dli struct, functions are represented as member variable pointers, so you can check the existence before making call.

In any case, I do not think that mere NOP instead of missing function is a good idea in most places (say the function is supposed to return a handle that you use downstream in your code, it would cause havoc).

Mirek

Thanks. I did check a member for NULL pointer. It looks like this dll is just a proxy to other dlls and I should check version of API instead of a pointer to function because call to clRetainDevice doesn't crash with Intel platform an does crash with NVIDIA although they implement the same version of this API ...

I tried to check content of IMAGE_EXPORT_DIRECTORY.AddressOfFunctions. But everything seemed to be fine.

Sorry for false alarm.