IT Technology Humor Programming Web Dev CS Design

Home

Simple C++ DLL Loading a Message Box

Nov 17, 2008 Baran Ornarli @ 3:02 am GMT C++ Win32 API

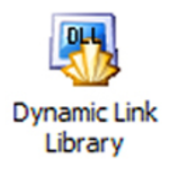
This tutorial shows you how to create a **C++ Win32 DLL** (Dynamic Link Library) which you can then use to call functions from that **DLL** in your main **C++** program. I will be calling a **MessageBox** from the DLL and load it inside my main **C++** program using **LoadLibrary** and **GetProcAddress**. I will also exchange integers between the **DLL** and the main program.

YouTube Tutorial

I also have a video tutorial for this on youtube:

DLL Project

I first start by creating our **DLL**, using CodeBlocks you can create a new project as a Dynamic Link Library (**DLL**). It may be different in other IDEs or editors. Somewhere you have to specify it's a **DLL** project.



DLL (Dynamic Link Library)

Create 2 files, one **C++** file (.cpp or .cxx) and one header file (.h).

DLL Header File

This will be our **DLL** header file:

```
#ifndef __MAIN_H__
#define __MAIN_H__

#include <windows.h>

/*  To use this exported function of dll, include this header
 *  in your project.
 */

#define DLL_EXPORT __declspec(dllexport)


#ifdef __cplusplus
extern "C"
{
#endif

int DLL_EXPORT MsgBox(int x);

#ifdef __cplusplus
}
```

Forum Community

Follow me on

51 readers BY FEEDBURNER

Search

Popular content

All Time:

- 45 Incredible Futuristic Scifi 3D City Illustrations
- C++ Win32 API Tutorial
- Singleton C++
- How to Make a Wireframe Render in Cinema 4D (C4D)
- Harvard scientists discover protein GDF-11 that reverses heart disease in old mice

Get Free Updates

Enter email to receive new posts:

Subscribe

We will never distribute your email address.

Updated Content

- C++ Tutorials

Sign Up! | Login

```
#endif

#endif // __MAIN_H__
```

Basically, there is a definition called __MAIN_H__ which will allow our header file to only be called once, because if it is not defined, I define it and include the code, otherwise I don't repeat the same code.

I include **windows.h** which has our **C++ Win32 API** functions and then define DLL_EXPORT as __declspec(dllexport) which just means that the function can be exporting via **DLL**.

Then I check if it's a C++ compile or a C compile, if it is C++ the preprocessor will use extern "C" so that it works in C programs as well.

Then I define an int DLL_EXPORT function called MsgBox with a parameter (int x). Thus, it will return an int and take an int as an argument.

Main DLL Program

This is our main.cpp file. Our header file was called main.h.

```
#include "main.h"

// a sample exported function
int DLL_EXPORT MsgBox(int x = 0){
    MessageBox(0, "Join us at Inferno Dev!", "DLL Message", MB_OK | MB_ICONINFORMATION);
    return x;
}

BOOL WINAPI DllMain(HINSTANCE hinstDLL, DWORD fdwReason, LPVOID lpvReserved)
{
    switch (fdwReason)
    {
        case DLL_PROCESS_ATTACH:
            // attach to process
            // return FALSE to fail DLL load
            break;

        case DLL_PROCESS_DETACH:
            // detach from process
            break;

        case DLL_THREAD_ATTACH:
            // attach to thread
            break;

        case DLL_THREAD_DETACH:
            // detach from thread
            break;
    }
    return TRUE; // successful
}
```

I have included the same function in the header file, with integer parameter and return value as a DLL_EXPORT type.

The default value for x is 0.

Then MessageBox which is a **Win32 API** function will display a message box with no owner (the first parameter), the message is then written, then the title of the message box, and finally MB_OK which means the messagebox will display an OK button, and MB_ICONINFORMATION which means messagebox will have an ICONINFORMATION icon.

Then I return the integer.

DllMain function is the standard main function for all DLLs. I wasn't doing anything special so I didn't put any real code here.

Now compile this dll as "InfernoDevelopment.dll" (or name your project as InfernoDevelopment) because that is the **DLL** we call in the code below.

Main C++ Program

This will be our main program that will call our DLL function and load our DLL as well.

```
#include <iostream>
#include <windows.h>

using namespace std;

typedef int (*MsgFunction)(int);

HINSTANCE hinstDLL;

int main(){
  MsgFunction MsgBox(0);
  hinstDLL = LoadLibrary("InfernoDevelopment.dll");
  if(hinstDLL != 0){
      MsgBox = (MsgFunction)GetProcAddress(hinstDLL, "MsgBox");
  }
  if(MsgBox == 0)cout << "MsgBox is NULL\n";
  int x = MsgBox(5);
  if(x == 5){
    cout << "Message displayed!\n";
  }
  FreeLibrary(hinstDLL);
  return 0;
}
```

I declare my includes, iostream for cout and cin. Windows.h for LoadLibrary, GetProcAddress, and FreeLibrary.

I typedef (define a type) an int returning (*MsgFunction) (the name of the type) and an integer argument. This is my definition for the function I am about to call.

Then I declare MsgFunction type called MsgBox, set it equal to zero so that I can check if the function was actually loaded properly.

HINSTANCE hinstDLL is our Win32 instance of that DLL. I set LoadLibrary("InfernoDevelopment.dll") as hinstDLL, calling our DLL file.

Note: DLL should be in the SAME folder as this exe.

I check if hinstDLL is not 0, and is working. Then I cast the GetProcAddress(hinstDLL, "MsgBox"); as a (MsgFunction) type and set that to MsgBox. The parameter in GetProcAddress will take the instance and find the process address where there is a function called "MsgBox" in the DLL.

Finally we call the function and check if we got the exact same integer as a return or a zero. I also checked if MsgBox was 0, and if so, an error will be displayed noting that the MsgBox wasn't properly loaded by GetProcAddress.

Then I FreeLibrary(hinstDLL) to avoid memory leaks and to let the DLL go, because we already called our function.

If you have any problems, please visit our **forums**.

Bookmark/Search this post with:

Quote

Hi Man. Very Very Nice, But

Aug 1, 2009
unknown (not verified)

hi man.
very very nice, but i've a question:
the DLL function ("MsgBox") needs to be called by the program where it's injected.
how can i code a simple function (the same "MsgBox" into a DLL) that doesn't needs to be called by the programm?

sorry for my bad english

Reply Quote

Finally A Tutorial That "Just

Aug 2, 2009
Anonymous (not verified)

Finally a tutorial that "just works" and makes some degree of sense! Many thanks!

Reply Quote

This Is A Pretty Good

Aug 4, 2009
Shaun (not verified)

This is a pretty good tutorial, but it seems that you're over-complicating things. The CodeBlocks DLL project works so that you can copy the header file and the .a file you get after compiling, and then include the header in your implementation and link it against the .a file. This way you can just use the function in your program without having to call any APIs.

Reply Quote

How Did You Get The Ccp

how did you get the ccp project tell me pls

Dec 13, 2009
Anonymous (not verified)

Reply Quote

I Can Not Run This

Mar 11, 2010
kanak_bit0105

I can not run this program.when I run this program the following notice occurred.
"You must select a host application to "run" a library"
what can i do?
what is host application?
plz help me.

Reply Quote

I Can Not Run This

Mar 11, 2010
kanak_bit0105

I can not run this program.when I run this program the following notice occurred.
"You must select a host application to "run" a library"
what can i do?
what is host application?
plz help me.

Reply Quote

Thanks, It Worked Out!,

May 24, 2010
Anonymous (not verified)

thanks, it worked out!, kanak_bit0105, u can create another app(lets say app2.cpp),?and copy the dll file into the same folder where your app.exe .

Reply Quote

What Is Host Application?

Sep 14, 2010
basscleff

what is host application?

Reply Quote

Post new comment

Your name:

Anonymous

E-mail:

The content of this field is kept private and will not be shown publicly. If you have a **Gravatar** account associated with the e-mail address you provide, it will be used to display your avatar.

Homepage:

Comment: *

Type the text

Privacy & Terms

Save Preview

Home | Forum | Contact | About