

A Tutorial on creating DLLs with VC++

How to create a WIN32 DLL from the scratch, A beginners tutorial. This tutorial explains each steps involved in writing a WIN32 Dynamic Link Library, with the help of sample programs. This tutorial also includes directions on how to call the exported functions using Visual Basic.

Visual basic is very fast and easy tool for developing applications with high degree of user friendliness , but it lacks some important functionalities like direct access to hardware, multi threading(Activex servers allows some type of multi threading !). The easy and effective solution for this problem is to write a DLL. But beginners experience problems in understanding how to write a dll. .Here is a simple tutorial on writing DLLs with Visual C++. I assume that you know a little bit Visual Basic and 'C' programming.

What is Static linking and Dynamic linking ?

The libraries can be linked to the executable in two ways . one is static linking and the other is dynamic linking. Static linking is the method used to combine the library routines with the application. When building the program , the linker will search all the libraries for the unresolved functions called in the program and if found, it will copy the corresponding object code from the library and append it to the executable. The linker will keep searching until there are no more unresolved function references. This is the most common method for linking libraries.This method boasts of ease of implementation and speed but will generate bulky executable. In dynamic linking , the required functions are compiled and stored in a library with extension .DLL. Unlike static linking, no object code is copied in to the executable from the libraries.Instead of that, the executable will keep the name of the DLL in which the required function resides. and when the executable is running , it will load the DLL and call the required functions from it. This method yields an executable with small footprint , but have to compromise on speed a little.

Example 1

Objective

Create a Dynamically linkable library which can calculate and return the sum of two numbers passed to it.

Tools required

visual C++ 6 , Visual basic 6

Writing The DLL

Start VC++ IDE , select New from File menu. Then select Win32 Dynamic - link library from Projects tab(picture-1). enter project name as example1 , then click OK button. Now you can see a dialog box with caption Win32 Dynamic - Link library - step 1 of 1 (picture-2).

Picture 1

Picture 2

Select a simple DLL project and click Finish. Now open exaple1.cpp from fileview (picture-3)

Picture 3

add these lines in to example1.cpp

```
int _stdcall sum(int x , int y)
{
return x+y;
}
```

It is a simple function which returns the sum of numbers passed to it.

Now our function is ready. The next step is to create a `.def` file which informs the linker to export our function. Select New from File menu and then select text File from Files tab. Give it a name `example1.def`. Now you can see an empty file . Add these lines in to `example1.def`

```
LIBRARY EXAMPLE1
```

```
EXPORTS
```

```
sum @1
```

Select Build `example1.dll` from build menu. The build process should complete without any errors. If any error occurs , correct it and rebuild again. Once the build process is completed successfully , you can see `example1.dll` in debug directory.

Now we have successfully built the DLL and it is the time to test it. For testing the DLL we have to write a small program in VB. Open a new project in Visual Basic. Place three text box and a command button like in picture 4.

Picture 4

Copy the code given bellow , to the code window

```
Private Declare Function sum Lib "example1.dll" (ByVal x As Long, ByVal y As Long) As Long Private Sub Command1_Click()
```

```
Text3.Text = Str(sum(CInt(Text1.Text), CInt(Text2.Text)))
```

```
End Sub
```

copy example 1.dll to windows or system directory . Then run VB ,enter some values to text1 and text2 and click Calculate button. If you can see the sum in text3, you have completed example1 successfully ! congratulations !.

Download the source code and project files for example1

Example 2

Objective

Create a DLL that contains a function to convert lowercase chars to uppercase and return the result.

Tools required

visual C++ 6 , Visual basic 6

Writing The DLL

Create a new win32 DLL project in VC++ just like in the previous example. insert the statement given below before main

```
#include string.h
```

Insert the following code after main

```
char* _stdcall to_upper(char *lowerstring)
{
    _strupr(lowerstring);
    return lowerstring;
}
```

Create a def file and insert the following code

```
LIBRARY example2
```

```
EXPORTS
```

```
to_upper @1
```

Now save the project, build it and copy example2.dll to windows or system directory.

Create a new project in VB, place a text box and command button the form paste the following code to code window.

```
Private Declare Function to_upper Lib "example2.dll" (ByVal text As String) As String
```

```
Private Sub Command1_Click()
```

```
    If (Text1.text) = "" Then
```

```
        Exit Sub
```

```
    End If
```

```
    MsgBox to_upper(Text1.text)
```

```
End Sub
```

Run the program and enter some lower case characters in textbox. It should display a message box with same characters with case changed.

Download the source code and project files for example2

Example 3

Objective

Create a DLL which can read/write data to parallel port

This DLL is a very useful for interfacing parallel/serial ports. Since Visual Basic does not have any facility to communicate with parallel port directly , programmers will have to write code for accessing parallel port in a DLL and should be called from VB. In this DLL , in this dll , accessing the device registers is done using functions `_Inp()` and `_Outp()`, which is declared in `‘conio.h’` . More details about these functions can be found at MSDN online .

We will export two functions from the DLL. One is for reading from a device register and the other is for writing to the device register. Let these functions be `InPort()` and `OutPort()`. `InPort()` will take one parameter ie; the address of the register to be read. and `Outport` will take two ie; the address of the register to which the data to be written and the data itself.

Note: to do this project , You need to have some knowledge in Parallel port interfacing.

Tools required

visual C++ 6 , Visual basic 6

Writing The DLL

Create a new win32 DLL project in VC++ like in the previous examples. insert the statement given below before main

```
#include conio.h
```

Insert the following code after main

```
short _stdcall InPort(short portaddress)
```

```
{  
return _inp(portaddress);  
}
```

```
void _stdcall OutPort(short portaddress,short data)
```

```
{  
_outp(portaddress,data);  
}
```

create a .def file with name example3.def and insert the following code

```
LIBRARY example3
```

```
EXPORTS
```

```
InPort @1
```

```
OutPort @2
```

now save the project and build example3.dll.

for testing the dll , create a new project in visual basic.Create a form like in picture5.

Picture 5

Add the following code to code window

```
Private Declare Function InPort Lib "example3.dll" (ByVal portaddress As Integer) As Integer  
Private Declare Function OutPort Lib "example3.dll" (ByVal portaddress As Integer, ByVal data As Integer) As Integer
```

```
Private Sub Command1_Click()  
OutPort 888, 10 'Val("&h" + Text2.Text), Val(Text1.Text)  
End Sub
```

```
Private Sub Command2_Click()  
Text3.Text = Str(InPort(Val("&h" + Text2.Text)))  
End Sub
```

Run the program and enter some value in text1. then click Write it button. Again click the Read it button, now if text2 displays the number which you entered in text1, your dll is working successfully.

Download the source code and project files for example3

NOTE: This program will not work on Win NT, 2000, XP systems. Use WIN95/98 for testing

How to use this dll with parallel port ?