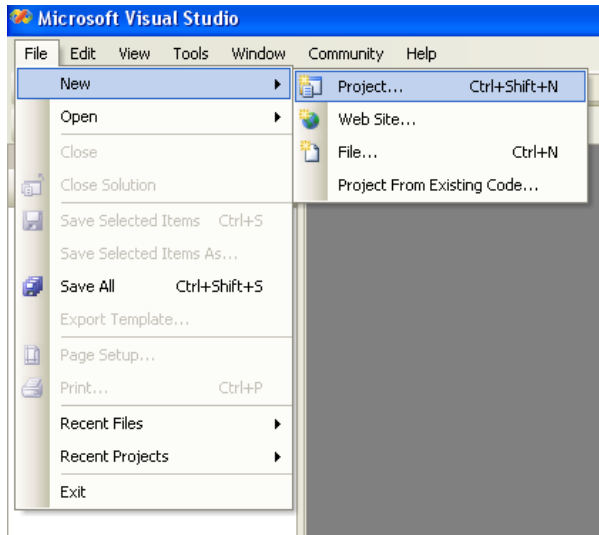


# How to Create a DLL using Visual C++ 2005 and call it from IDL

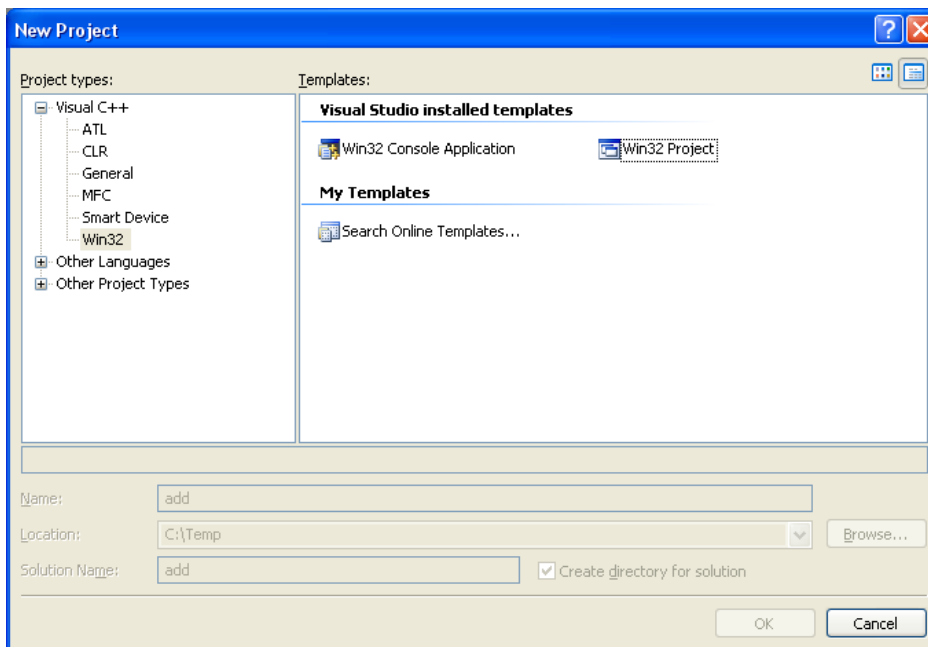
*By Mohammad Al-Sa'd and Munther Gdeisat*

## Step A: Creating a DLL using Visual C++ 2005

1. Launch Visual C++ 2005.
2. In the menu go to **FILE**→**New**→**Project**.

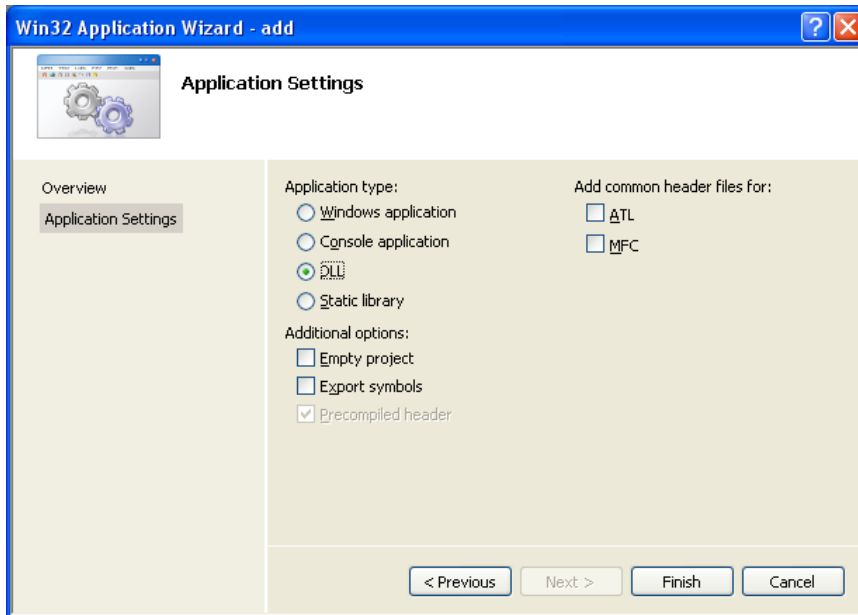


3. In **Project Types** choose **Win32**. In **Visual Studio installed templates** choose **Win32 Project**. Call this project add.

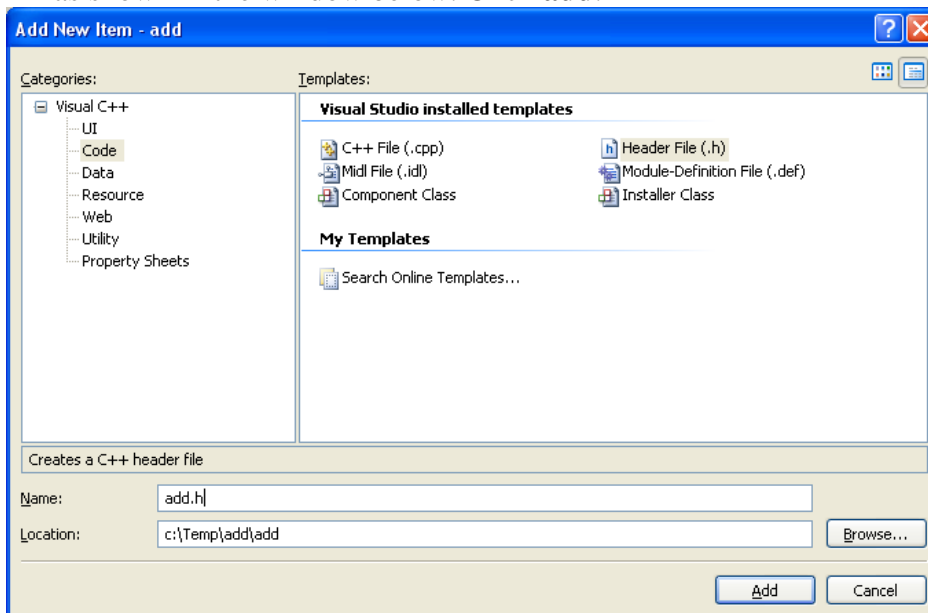


4. Click **OK**. Click **Next**.

5. Choose **DLL**. Click **Finish**.



6. In the menu, go to **Projects→Add New Item...** Add a new header file and call it *add.h* as shown in the window below. Click **add**.



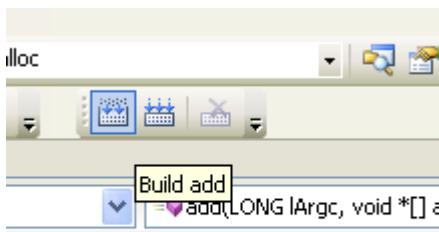
7. In the **Solution pane** click on **add.h**. Add the following code.

```
extern "C" __declspec(dllexport) int add(int argn, void *argv[]);
```

8. In the **Solution pane** click on **add.c**. Delete everything in this file and add the following code.

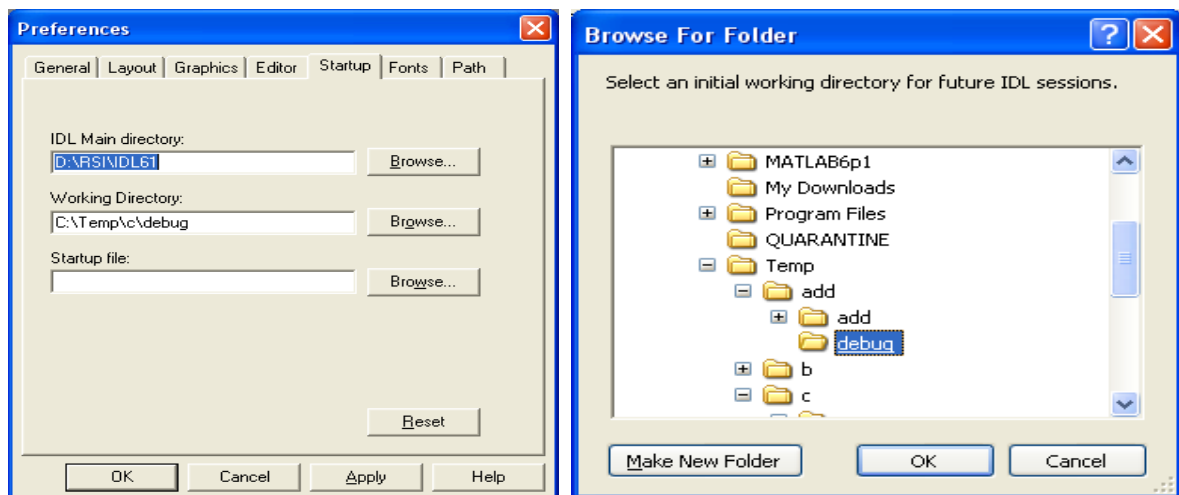
```
#include "stdafx.h"
#include "add.h"
int add(int argn, void *argv[])
{
    int a = *(int *) argv[0];
    int b = *(int *) argv[1];
    int *c = (int *) argv[2];
    c[0] = a;
    c[1] = b;
    return 1;
}
```

9. Using the **Solution pane**, open the file **add.cpp**. Click the **Build** icon to Build the project. The project should be built without errors.



## Step B: Calling the DLL from IDL using call\_external

1. Launch IDL. In the Menu go to **Files** → **Preferences**. Click on **Startup** Tab. Click on **Browse**. Navigate to the **add** directory, then to the **debug** directory. Click **OK**. Click **OK**.



- Open a new file using IDL. Name the file **hello.pro**. Type the following code inside the editor

```
a=1L;
b=2L;
c=lonarr(2);
print, c
r=call_external('add.dll', 'add', a, b, c, /unload)
print, c
end
```

Note that this code creates two long variables **a** and **b**. Also, it creates a two-elements vector **c**. The values of the elements in the vector are zeros. Then the code calls a DLL that sets **c[0]** to **a** and **c[1]** to **b**. If the IDL program prints **1 2**. Then the DLL is working. Please see the table below for the data types used in C++ and their equivalent in IDL.

Data Type in C++	Its equivalent in IDL	Number of bits to represent the data type
unsigned char	byte	8
short	int	16
int	long	32
float	float	32
double	double	64

- Compile and run the code using IDL. The following will result in the output window. If you get this output. Congregation. You have succeeded in calling the DLL from IDL.

