

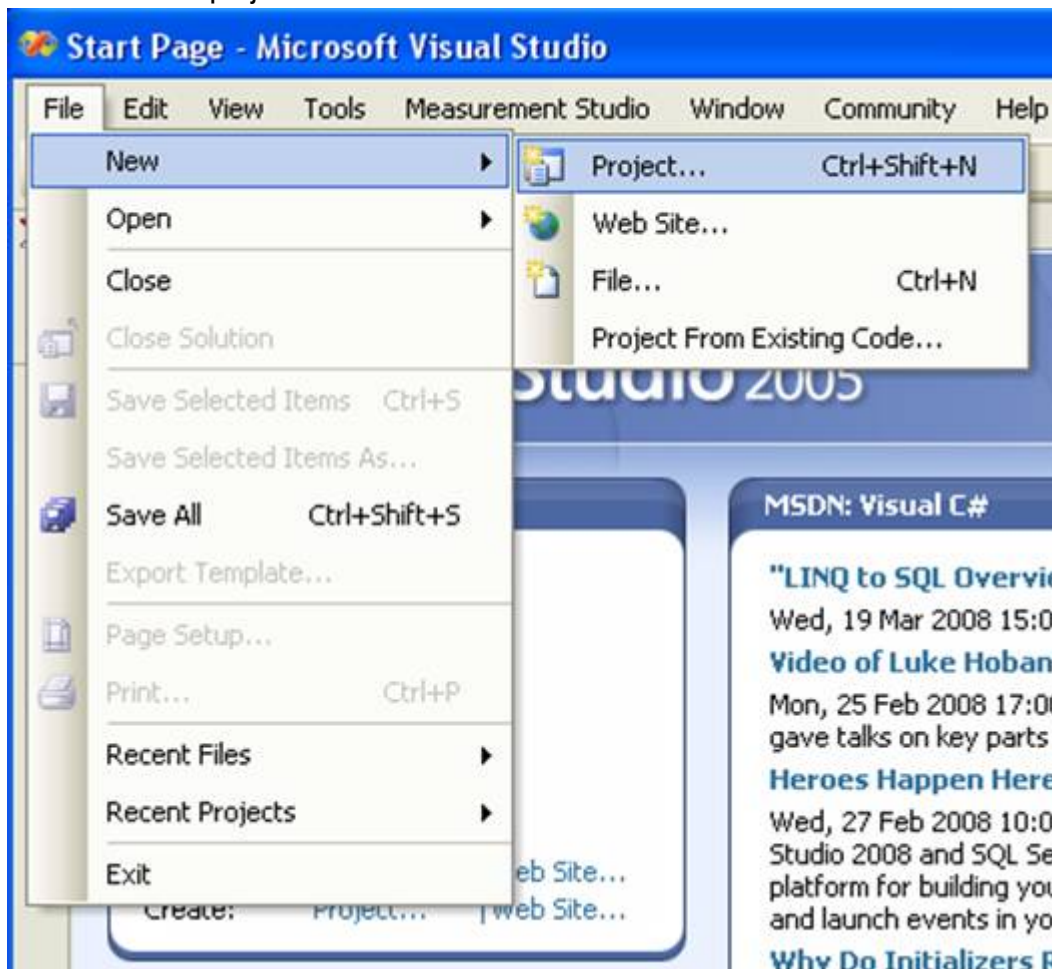
Using Existing C Code or a DLL in LabVIEW

By: Kevin_S 05-12-2008 01:55 PM Last Edited By: Kevin_S 01-27-2017 04:11 AM

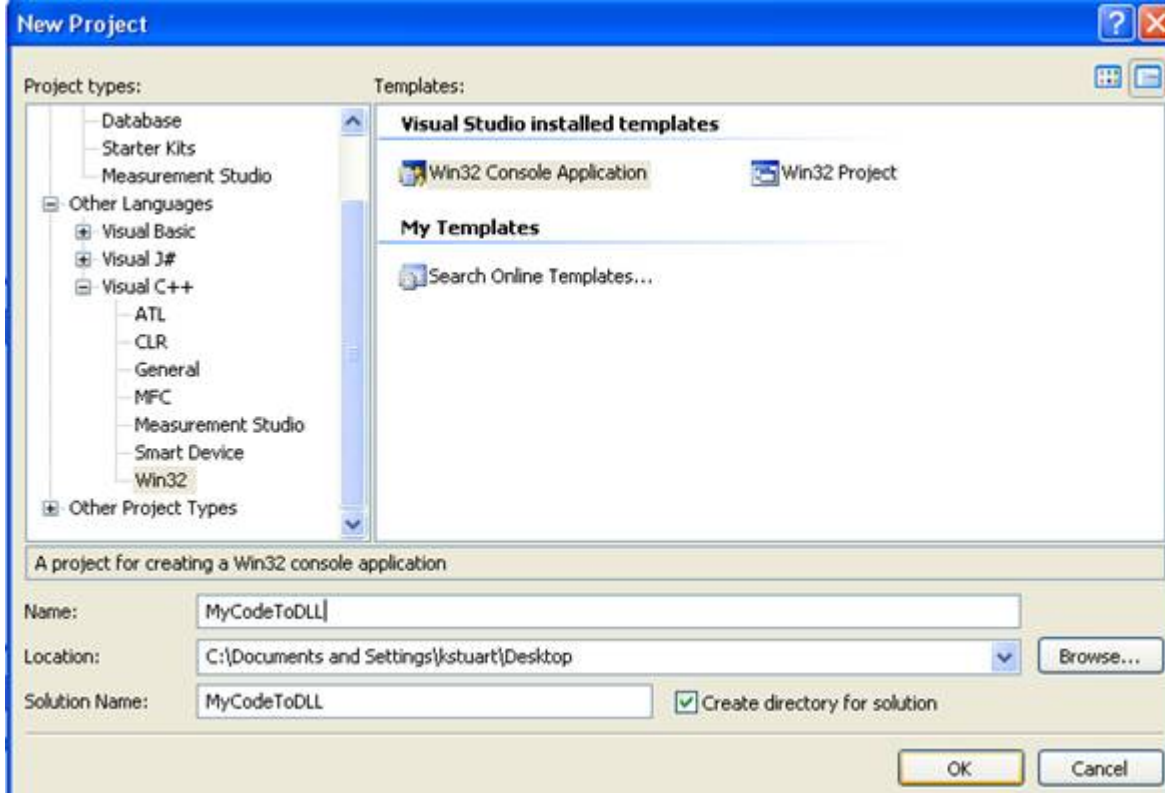


This tutorial explains how to convert existing C code to a DLL (Dynamically Linked Library) and then how to call that DLL from LabVIEW. If you already have a DLL, start on step 10. (Note: This tutorial explains how to create a DLL that can only be used in LabVIEW. If you want to create a DLL out of your C code that will then be downloaded to the FIRST cRIO controller, see this [article](#).)

1. Use an integrated development environment capable of creating DLLs. (This tutorial uses Microsoft Visual Studio 2005.)
2. Create a new project.

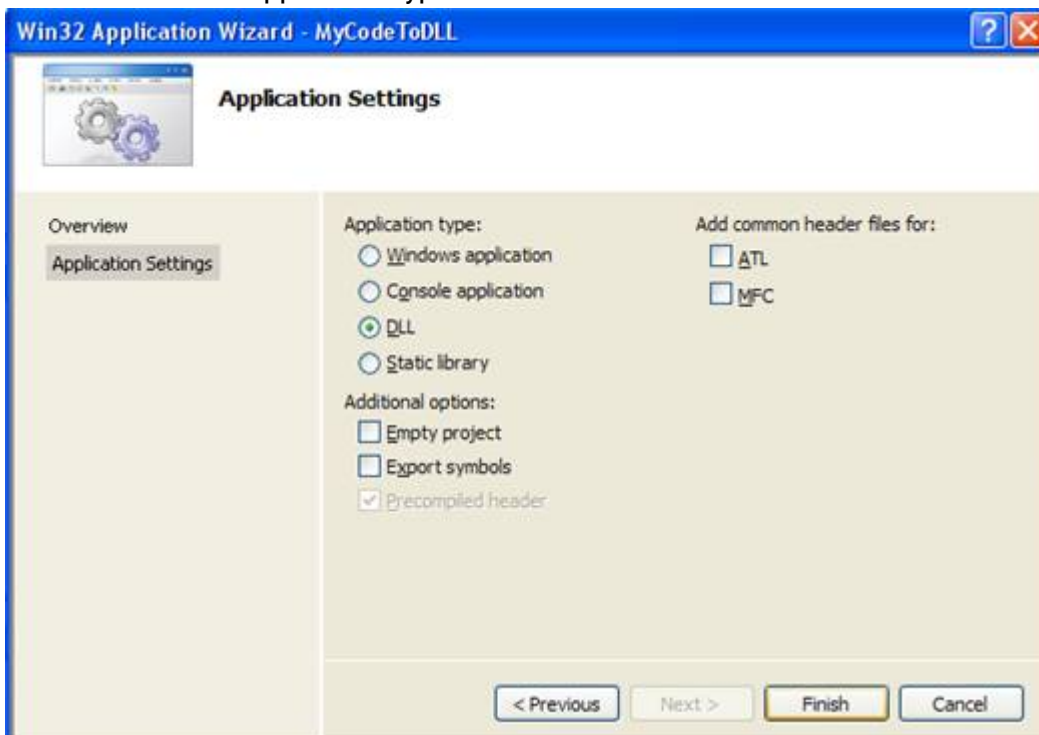


3. Create a C++ Win32 Console Application. Specify a Name. Specify a Location.



4. Click Next for more options.

5. Select DLL from Application type.



6. Add your function to the default program. (See my function in bold.)

```
int Multiply( int a, int b )
{
return a*b;
}
```

```

// MyCodeToDLL.cpp : Defines the entry point for the DLL application.
//

#include "stdafx.h"

#ifdef _MANAGED
#pragma managed(push, off)
#endif

BOOL APIENTRY DllMain( HMODULE hModule,
                      DWORD  ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    return TRUE;
}

int Multiply( int a, int b )
{
    return a * b;
}

#ifdef _MANAGED
#pragma managed(pop)
#endif

```

7. Add the following prototype to the default program. (See my prototype in bold.)
extern "C" __declspec(dllexport) int Multiply(int a, int b);

```

// MyCodeToDLL.cpp : Defines the entry point for the DLL application.
//

#include "stdafx.h"

#ifdef _MANAGED
#pragma managed(push, off)
#endif

extern "C" __declspec(dllexport) int Multiply( int a, int b );

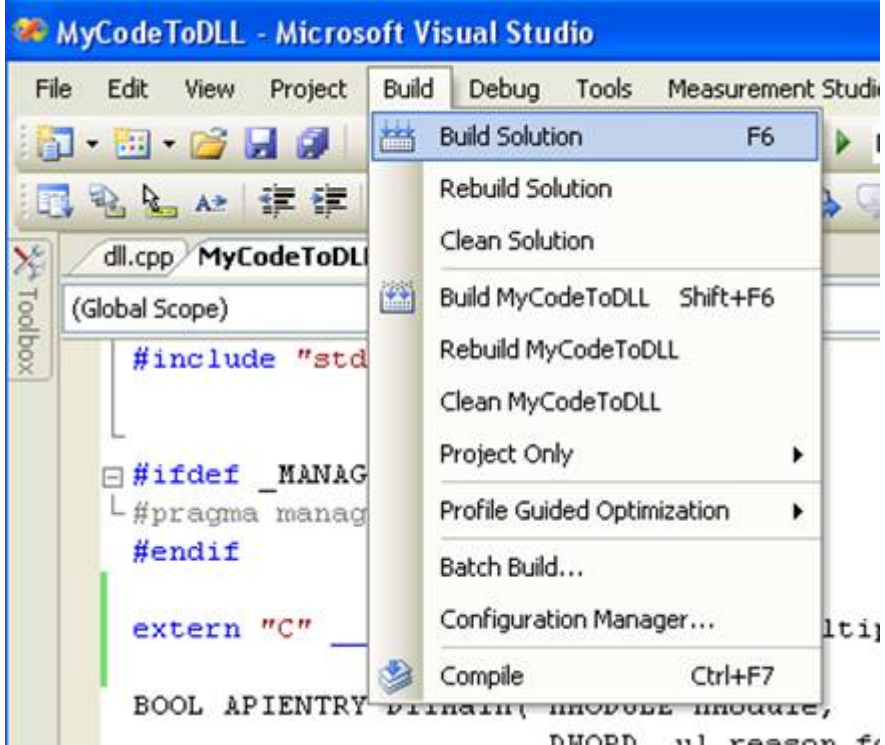
BOOL APIENTRY DllMain( HMODULE hModule,
                      DWORD  ul_reason_for_call,
                      LPVOID lpReserved
                      )
{
    return TRUE;
}

int Multiply( int a, int b )
{
    return a * b;
}

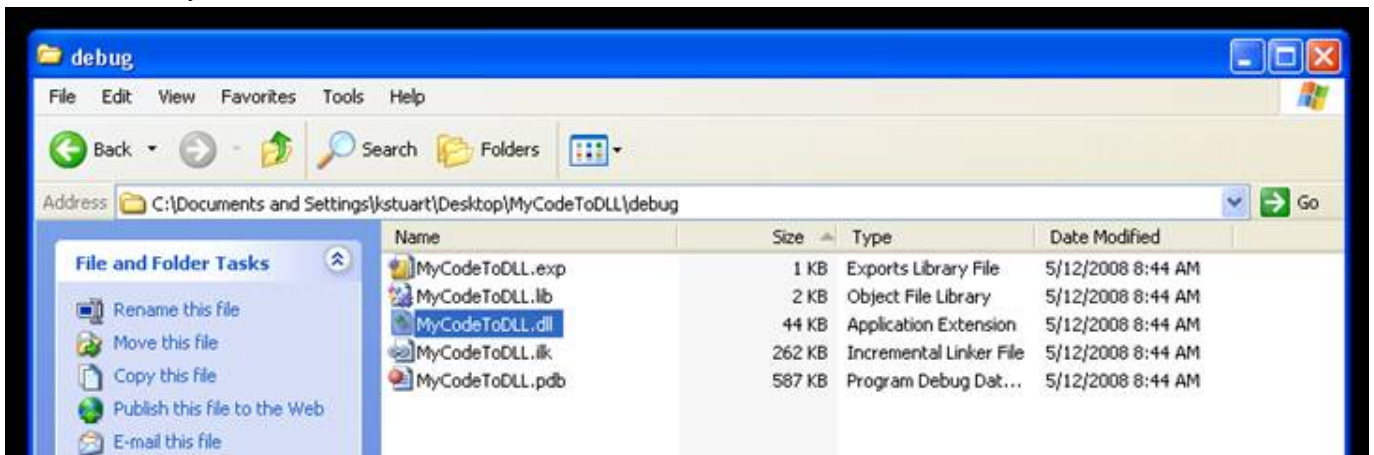
#ifdef _MANAGED
#pragma managed(pop)
#endif

```

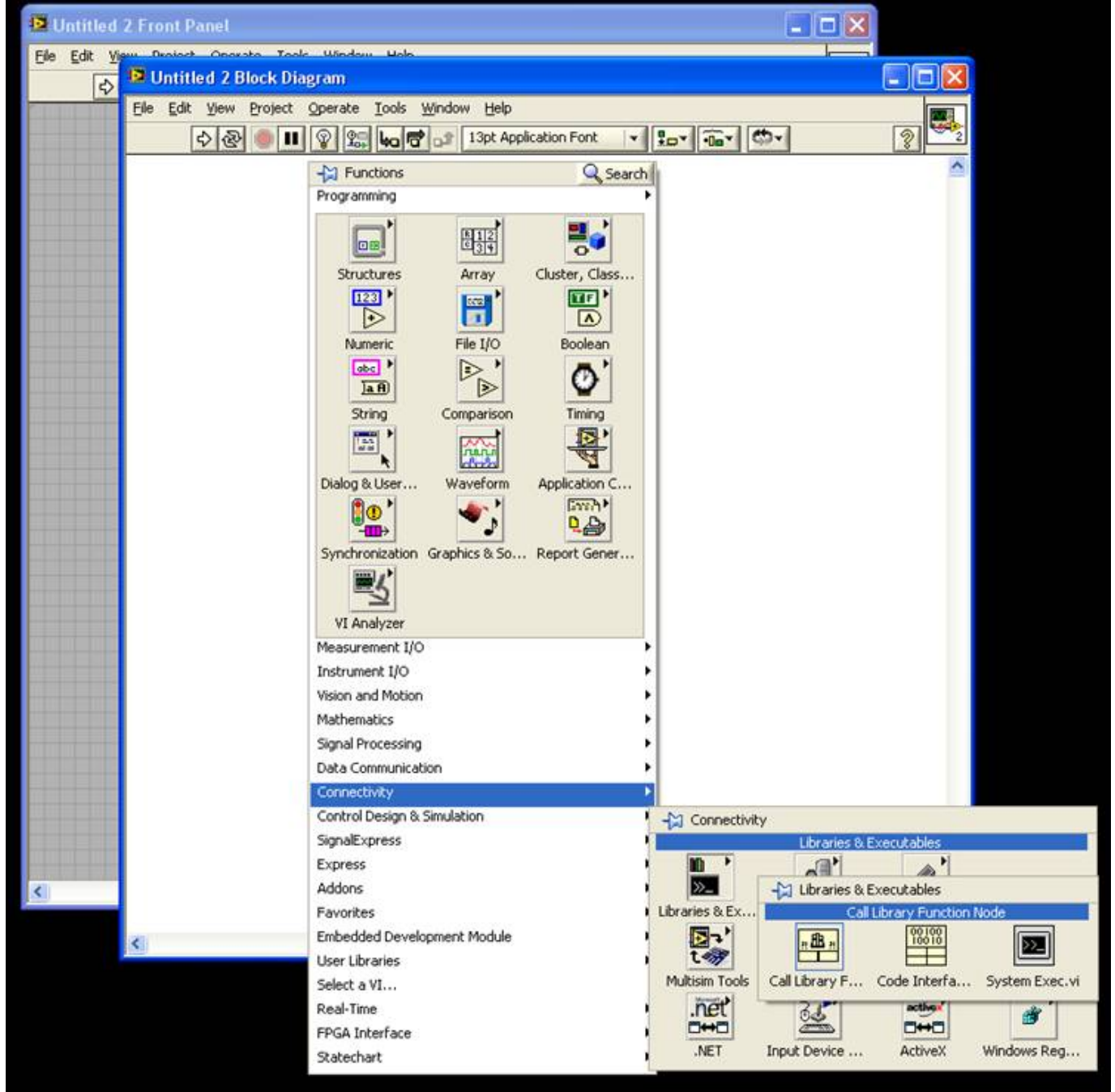
8. Build the DLL.



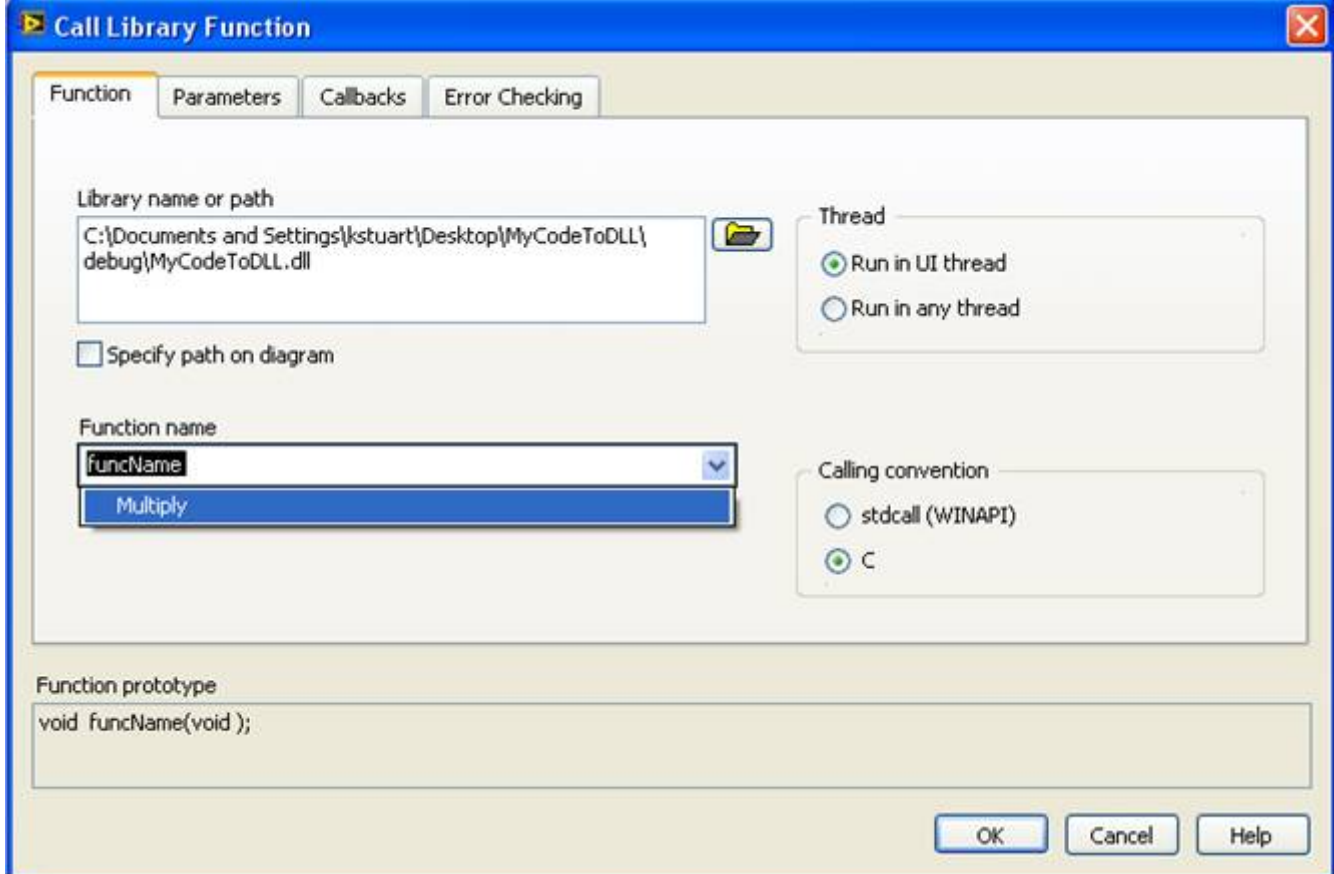
9. Note where your DLL was created.



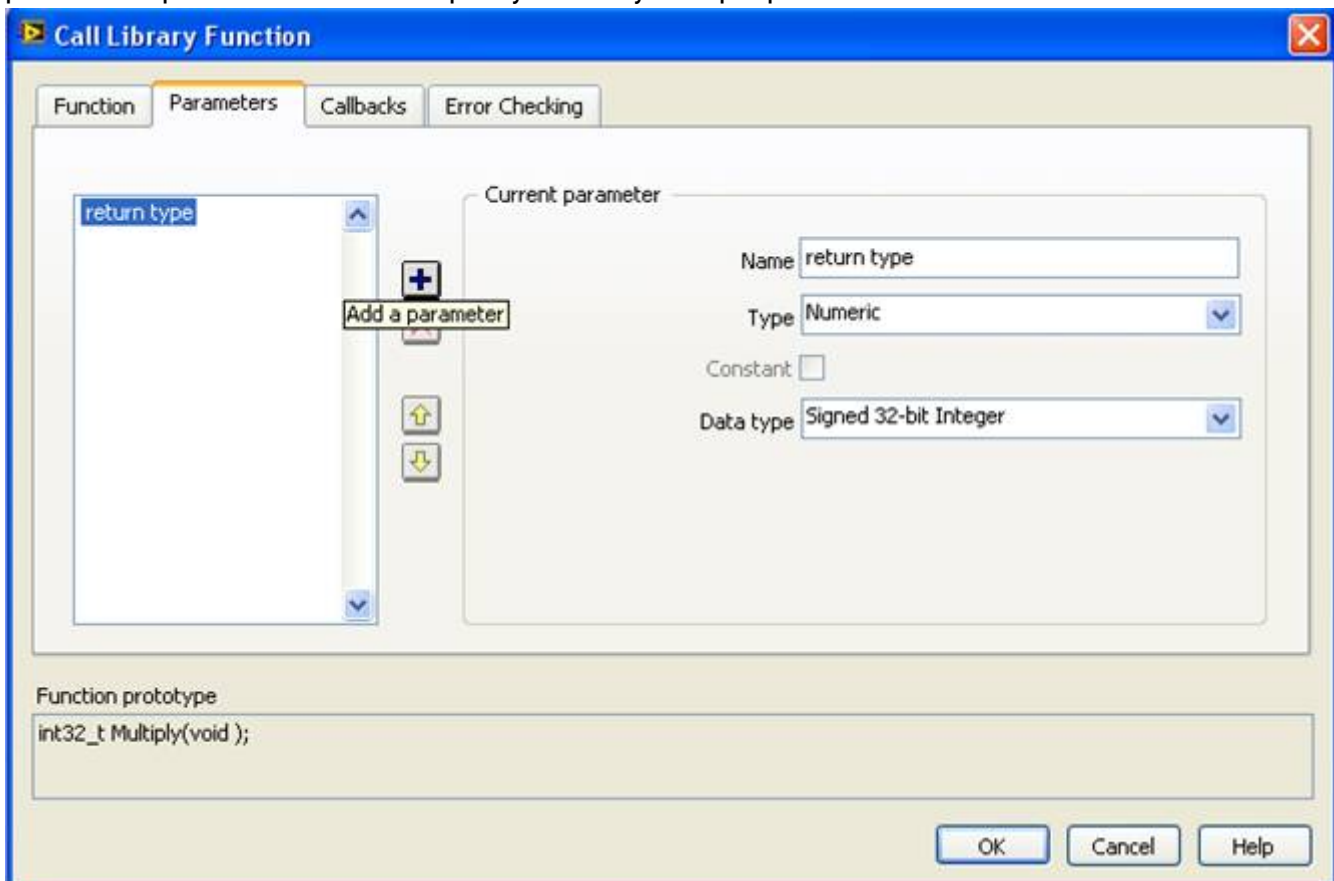
10. Open LabVIEW and place a Call Library Function Node on the Block Diagram.

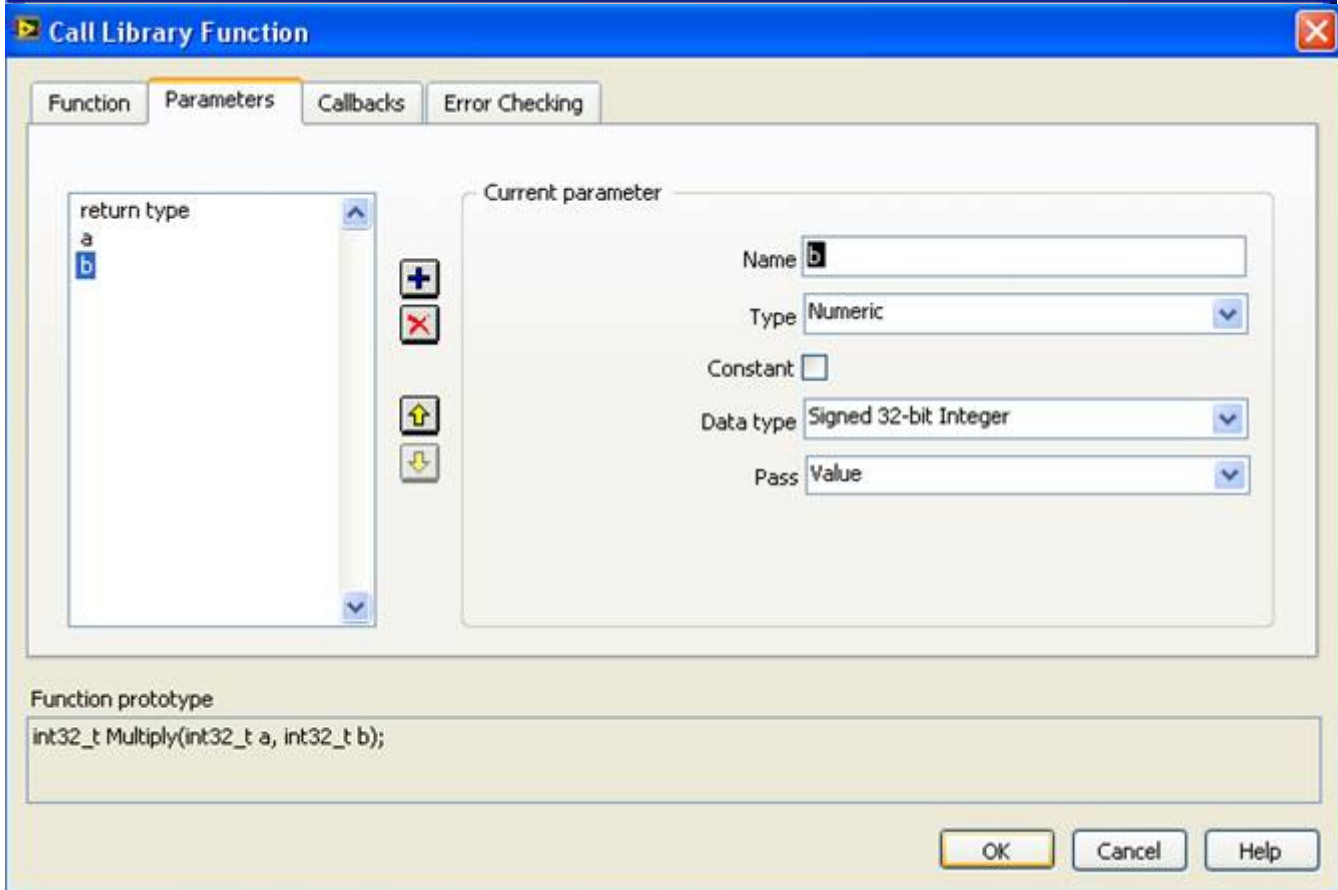
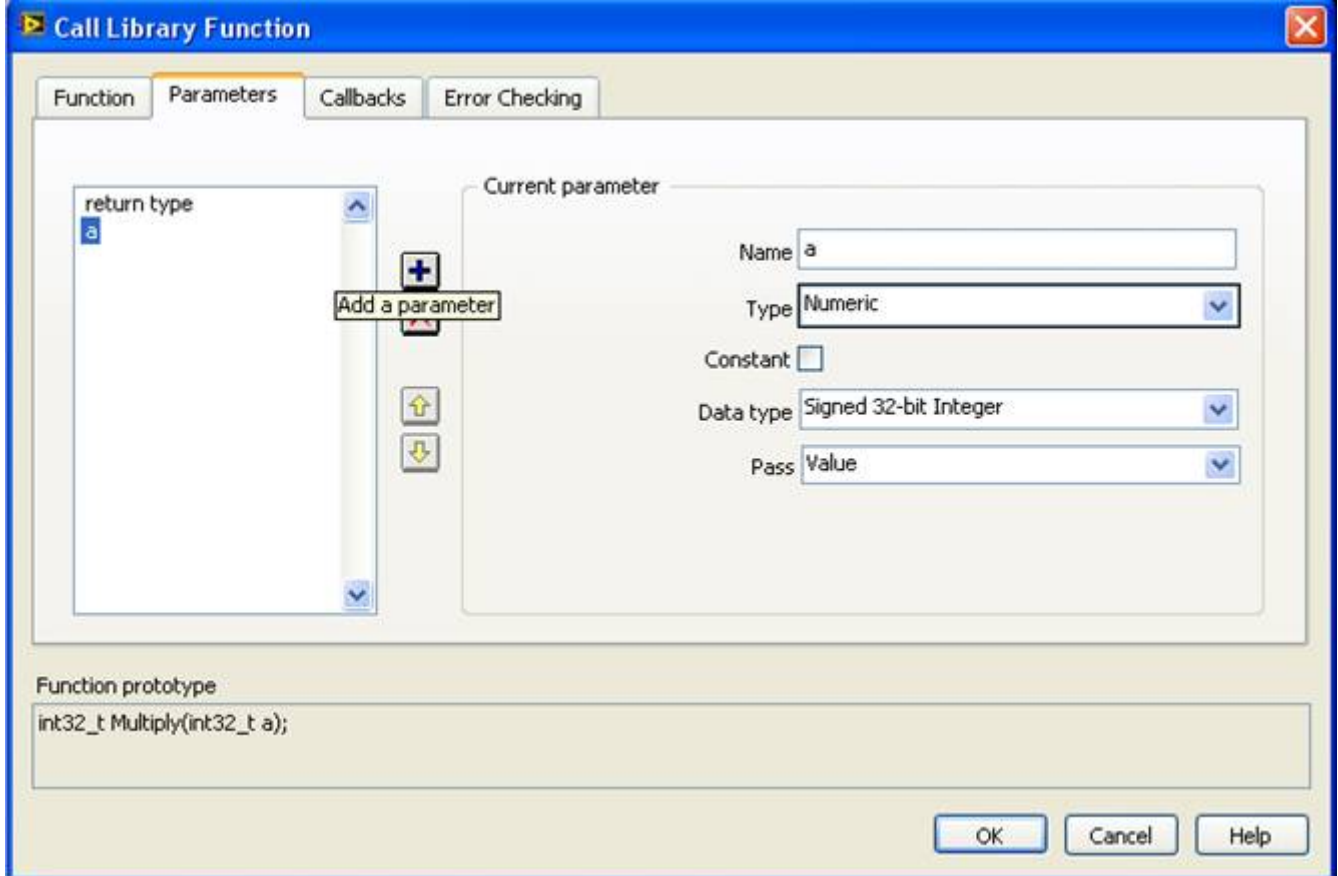


11. Double click on the Node and specify the DLL to load and the function to access.

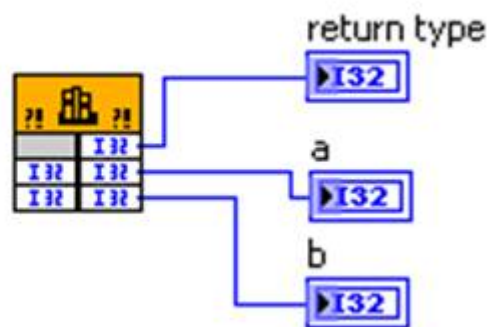
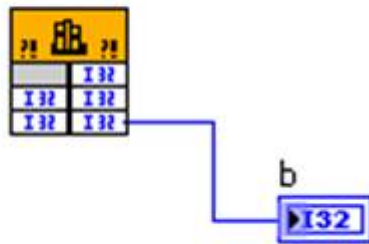
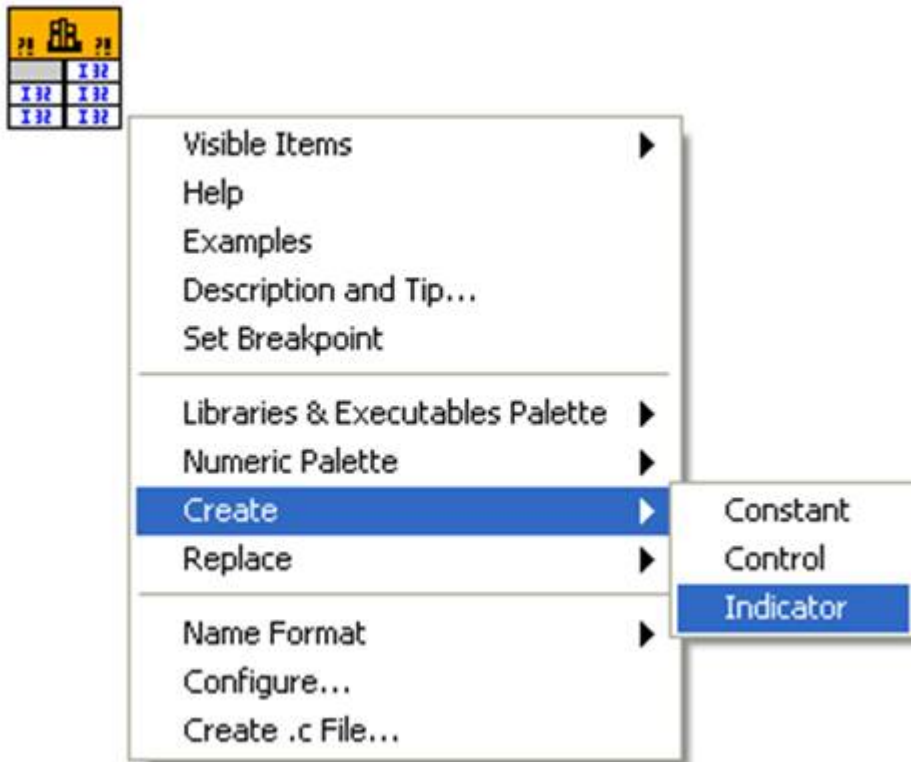


12. Under the Parameters tab, specify the Return Type (the data being passed back from the function) and press Add a parameter in order to specify each of your input parameters.

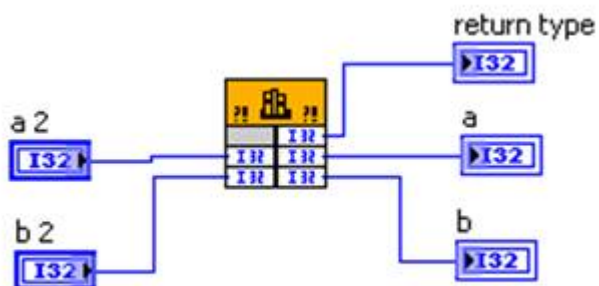
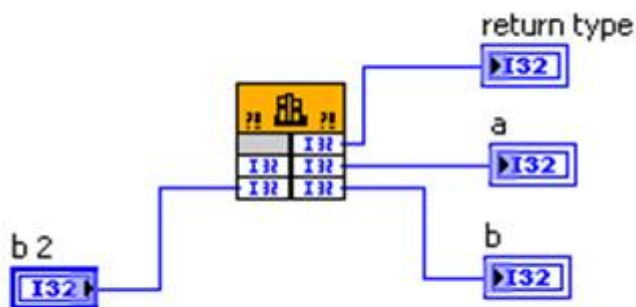
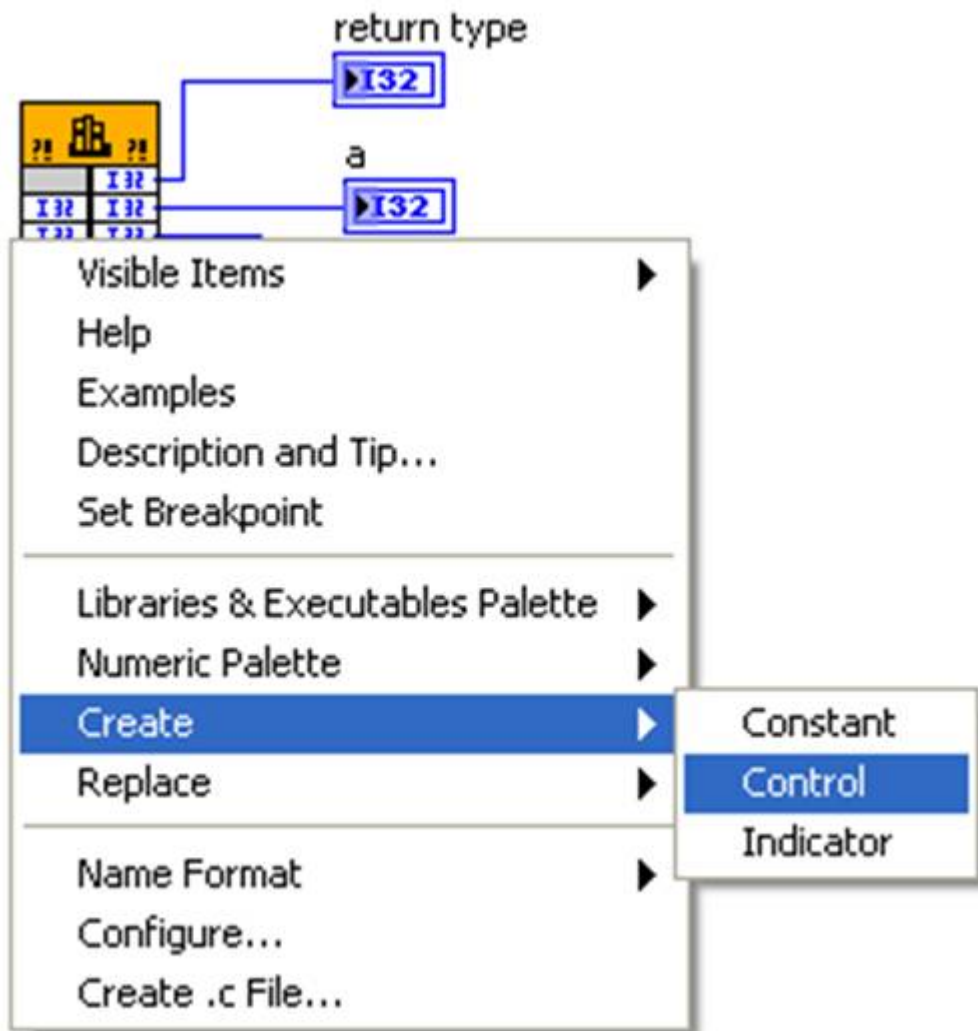




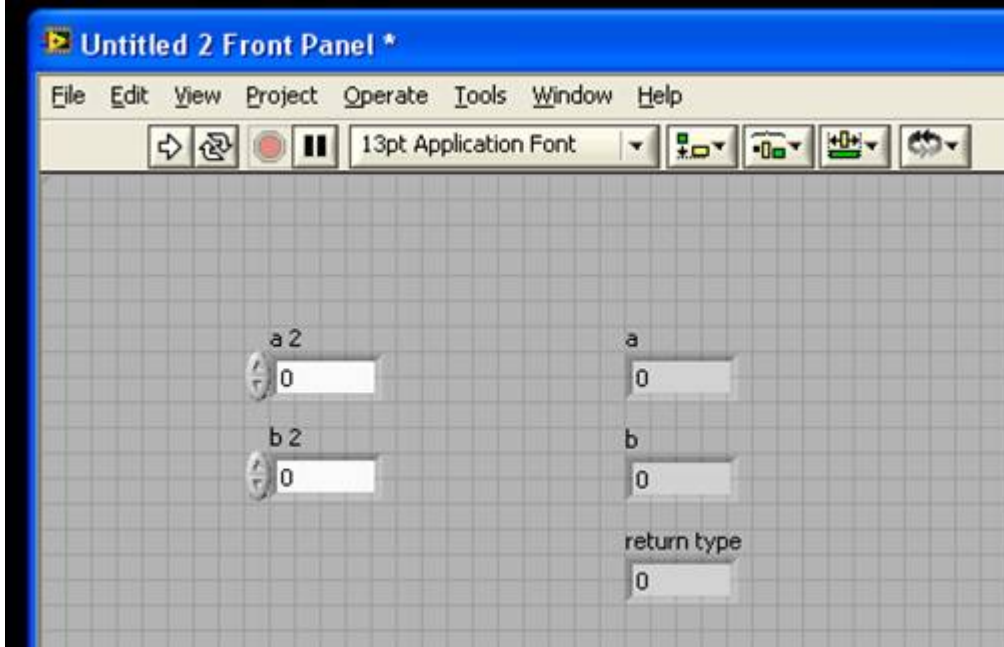
13. Right click on each of the 3 outputs and create an indicator. (Note: You have the option of viewing as outputs each of the input parameters to the DLL.)



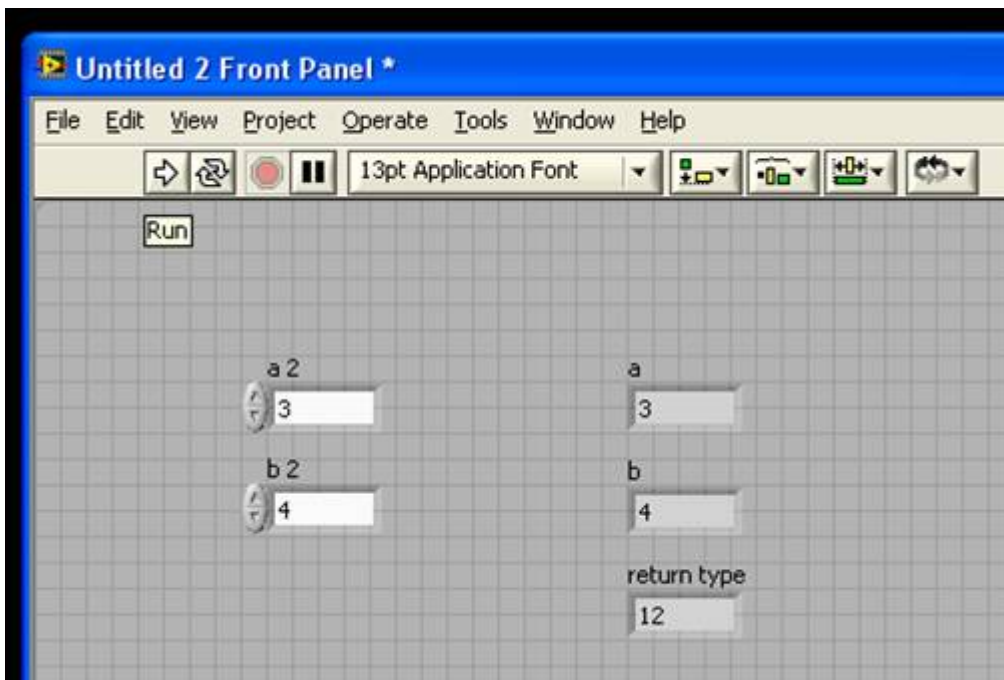
14. Right click on each of the 2 inputs and create controls.



15. Go to the Front Panel. Specify values for the 2 controls (the input parameters).



16. Run the VI in order to execute the DLL.



Example code from the Example Code Exchange in the NI Community is licensed with the MIT license.

★ 7 KUDOS [SHARE](#)

COMMENTS



parkcs MEMBER on 05-18-2012 02:47 PM

05-18-2012 02:47 PM

When I follow this exactly, I get "error LNK2005: _DllMain@12 already defined in dllmain.obj" error. Any help would be appreciated. I need C++ code to read data being collected by LabVIEW and write them into a new file.



0 KUDOS



sym2012go MEMBER on 12-23-2015 01:59 AM

12-23-2015 01:59 AM

In step 6 or 7, some lines of the code are not necessary:

```
BOOL WINAPI DllMain( HMODULE hModule,  
                    DWORD  ul_reason_for_call,  
                    LPVOID lpReserved
```

delete the above code and it will be OK.



0 KUDOS



San_Punt MEMBER on 08-18-2019 08:22 AM

08-18-2019 08:22 AM

nice example, thanks



0 KUDOS