



All Topics, MFC/C++ >> C++ / MFC >> [Unedited Reader Contributions](#)  
<http://www.codeproject.com/useritems/MatlabVS.asp>

## Matlab 7.1+ and Visual Studio 2005

By [abn9c](#).

An article describing how to compile Matlab 7.1-7.4 mexw32 files using Visual Studio 2005

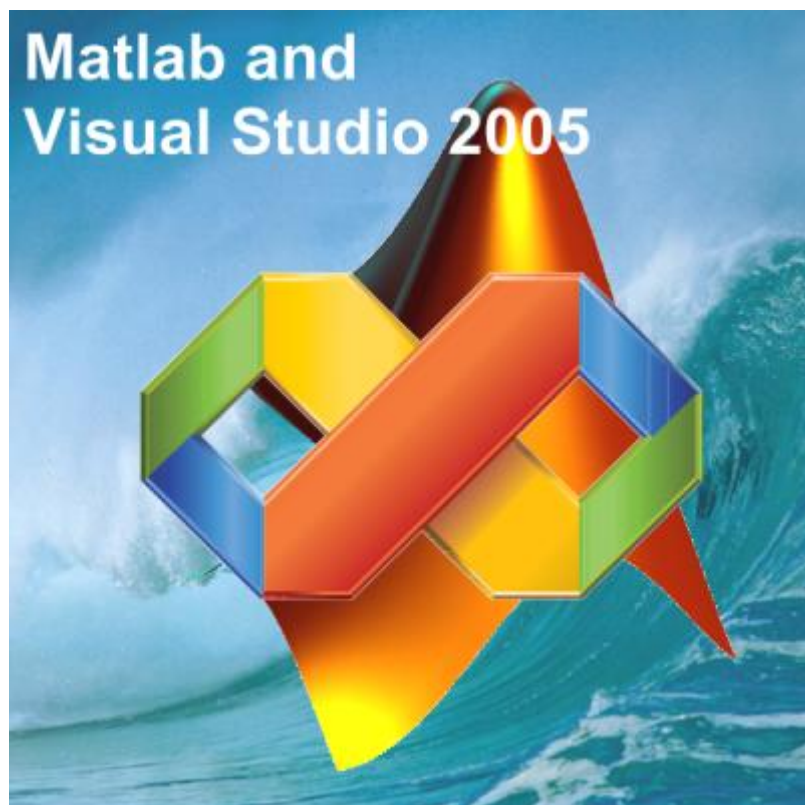
Note: [This is an unedited reader contribution](#)

C++ (VC8.0), C++/CLI, C  
Windows, .NET  
Win32, VS, MFC  
Dev  
Posted : 11 Jul 2007  
Views : 5,405

5 votes for this article.

Popularity: 2.41. Rating: 3.44 out of 5.

[Download MatlabVS.zip - 2,803.4 KB](#)



### Introduction

This program shows how to compile and link the new Matlab mexw32 files using Visual Studio 2005. Matlab 7.1 changed its mex file format to .mexw32 from .dll. All new versions, MATLAB 7.1 (R14SP3), MATLAB 7.2 (R2006a), MATLAB 7.3 (R2006b), and MATLAB 7.4 (R2007a), compile mexw32 files. Previously compiled mex .dll's must be recompiled. Using Visual Studio 2005 makes this process easy. Furthermore, there are many times where one needs to use outside headers and libraries from other C/C++ functions. Visual Studio works much better than Matlab's own compiler in these situations. Mex files allow one to create very powerful Matlab functions, and are able to use all the functionality of .m Matlab script files.

### Background

Why did I learn this in the first place? I needed to control a laser gyroscope from Matlab. I had the C

functions to control the laser from the command prompt, so I needed to implement the C code inside of a Matlab mexw32 file. Not only did I need to add the libraries and headers for the mex file to compile, but I also needed to add in the laser libraries. My example code contains function calls to the laser device, however they are commented out since they will not compile without the correct libraries. You can switch your code, for your own specific function calls, for mine. - A. Bryant Nichols Jr.

## Using the code

You can either open my solution file, and modify the properties to link to your matlab directory, or follow these steps:

- 1) Create a new C++ Win32 console application.
- 1) Add `mexversion.rc` from the MATLAB include directory, `matlab\extern\include`, to the project.
- 2) Create a `.def` file with the following text, and add it to the Solution. Note: change `myFileName` to yours.

```
LIBRARY myFileName
EXPORTS mexFunction
```

- 3) Under C/C++ General Additional Include Directories, add `matlab\extern\include`
- 4) Under C/C++ Preprocessor properties, add `MATLAB_MEX_FILE` as a preprocessor definition.
- 5) Under Linker General Output File, change to `.mexw32`
- 6) Under Linker General Additional Library Directories, add `matlab\extern\lib\win32\microsoft`
- 7) Under Linker Input Additional Dependencies, add `libmx.lib`, `libmex.lib`, and `libmat.lib` as additional dependencies.
- 8) Under Linker Input Module Definition File, add the `(.def)` file you created.
- 9) Under General Configuration Type, choose `.dll`
- 10) Add more links to includes and libraries that you will need for your program. In my case I also added the `\include` and `\lib` dependencies for the laser gyroscope.

## The Code

The headers required by Matlab. Note: include your own headers for more specific functions.

```
#include "stdafx.h"
#include "Windows.h"
#include "mex.h"
#include "matrix.h"
// #include "AerSys.h"
```

This is my own function, used to control the laser.

```
void StartDevice();
```

This is the main entry point for the mex file. `nlhs` is the number of outputs, `*plhs` is the output pointer, `rlhs` is the number of inputs, and `*prhs` is the input pointer.

```
void mexFunction(int nlhs, mxArray *plhs[], int nrhs, const mxArray *prhs[])
```

The first variable read from the input is `iCase` which is used for the case statement. See section Sample for all of the possible input/output options.

[-] Collapse

```
if (nrhs>0)
{
    data1 = mxGetPr(prhs[0]);
    iCase = (int)data1[0];
}
if (nrhs>1)
{
    data1 = mxGetPr(prhs[1]);
    dist1 = data1[0];
    if (iCase==4)
    {
        char buf[128];
        mwSize buflen;
        // Allocate enough memory to hold the converted string.
        buflen = mxGetNumberOfElements(prhs[1]) + 1;
        mxGetString(prhs[1], buf, buflen);
        sprintf_s(szProgram,buf);
    }
}
if (nrhs>2)
{
    data1 = mxGetPr(prhs[2]);
    dist2 = data1[0];
}
StartDevice();
if (iCase==3)
{
    if (nlhs>=2)
    {
        plhs[0] = mxCreateDoubleMatrix(1, 1, mxREAL);
        plhs[1] = mxCreateDoubleMatrix(1, 1, mxREAL);
        data1 = mxGetPr(plhs[0]);
        data1[0] = dXPos/AZCOUNTS;
        data1 = mxGetPr(plhs[1]);
        data1[0] = dYPos/ELCOUNTS;
    }
}
return;
```

This is the call to my function, however it will not compile without the correct laser device libraries.

```
void StartDevice()
```

## Sample Matlab Commands

These are optional commands that could be used in Matlab to run the laser. Note: even though the filename is MatlabVS.m, if one types MatlabVS(1); at the Matlab command prompt, the MatlabVS.mexw32 file gets priority and will be run. This is a nice way to comment your mexw32 files so users can understand its usage format.

```
% function MatlabVS(option, var1, var2)
%
% 0) To Reset...
%     MatlabVS(0);
%
% 1) To Initialize...
%     MatlabVS(1);
%
% 2) To Absolute Move...
%     MatlabVS(2,Xdist,Ydist);
%
% 3) To Return Positions
%     [AZ,EL]=MatlabVS(3);
%
% 4) To Run GCode...
%     MatlabVS(4,FileName);
%     e.g. MatlabVS(4,'Aero.pgm');
%
% 5) To Increment Move...
%     MatlabVS(5,Xdist,Ydist);
%
% 6) To Home...
%     MatlabVS(6);
```

## Points of Interest

Matlab is a great tool for array and matrix calculations, plotting, analysis, etc. The added use of C/C++ in Matlab mexw32 files allows one to acquire realtime data from external C controlled devices and analyze it right away inside of Matlab. With Matlab 7.1+ and Visual Studio 2005, the sky's the limit!

## History

Version 1.1

## About abn9c



I mainly use Visual Studio 2005 for programming. I am an electrical engineer in Washington DC.

Click [here](#) to view abn9c's online profile.

Add this article to:



# THE CODE PROJECT

The Visual Studio .NET Developer's Homepage

## Discussions and Feedback

0 comments have been posted for this article. Visit <http://www.codeproject.com/useritems/MatlabVS.asp> to post and view comments on this article.

Updated: 11 Jul 2007

Article content copyright abn9c, 2007  
everything else Copyright © [CodeProject](#), 1999-2007.