

How can I compile a MATLAB Engine application using Microsoft Visual Studio 9.0 or 10.0 ?

Date Last Modified: Wednesday, April 3, 2013

Solution ID:	1-78077S
Product:	MATLAB
Reported in Release:	No Release
Platform:	Windows
Operating System:	Windows Any

Subject:

How can I compile a MATLAB Engine application using Microsoft Visual Studio 9.0 or 10.0 ?

Problem Description:

I would like to call MATLAB from my C/C++ application using the Engine interface. However, I would like to compile my project in the Microsoft Visual Studio 9.0 (2008) or Visual Studio 10.0 (2010) environment, instead of compiling it in the MATLAB environment.

Solution:

To compile a MATLAB engine application in the Microsoft Visual Studio 9.0 (2008) or Microsoft Visual Studio 10.0 (2010) environments, perform the steps outlined below:

1. Open Microsoft Visual Studio and select

File->New->Project

and then select:

Visual C++ -> General -> Empty Project

Following this, enter the name of the project and its location in the indicated text areas and click OK.

2. Right-click on the SOURCE FILES folder in the SOLUTION EXPLORER and click Add -> "New Item...". Choose "C++ file (.cpp)", enter the name of the file as enginedemo.cpp. Click OK.

3. Copy the code from the source file: engwindemo.c, and paste it into this file enginedemo.cpp. The file engwindemo.c may be obtained from the following location:

`$MATLABROOT/extern/examples/eng_mat`

where \$MATLABROOT is the MATLAB root directory, and may be determined by entering the command:

```
matlabroot
```

at the MATLAB command prompt.

4. Right click on the project name and select PROPERTIES in the solution explorer to open the project properties. Once this window has opened, make the following changes:

a. Under C/C++ General, add the following directory to the field ADDITIONAL INCLUDE DIRECTORIES:

`$MATLABROOT\extern\include`

b. Under C/C++ Precompiled Headers, select "Not Using Precompiled Headers".

c. Under Linker General, add the directory to the field ADDITIONAL LIBRARY DIRECTORIES:

(For 32-bit Windows)

`$MATLABROOT\extern\lib\win32\microsoft`

(For 64-bit Windows)

`$MATLABROOT\extern\lib\win64\microsoft`

d. Under Configuration Properties -> Debugging

Add the following Target path in the Command ("Environment" slot):

(for 32 bit MATLAB)

`PATH=$matlabroot\bin\win32`

(for 64-bit MATLAB)

`PATH=$matlabroot\bin\win64`

7. Under Linker Input, add the following names to the field marked ADDITIONAL DEPENDENCIES:

`libmx.lib`

`libmat.lib`

`libeng.lib`

8. If compiling on 64-bit Windows, change the target to x64 as mentioned on the following link:

[http://msdn2.microsoft.com/en-us/library/9yb4317s\(VS.80\).aspx](http://msdn2.microsoft.com/en-us/library/9yb4317s(VS.80).aspx)

You should now be able to build the executable.

9. Make sure that the following directory is in the PATH:

(For 32-bit Windows)

`$MATLABROOT\bin\win32`

(For 64-bit Windows)

`$MATLABROOT\bin\win64`

To do this, check the Windows system path by clicking on Start -> Right click on Computer -> Properties -> Advanced System Settings -> Environment variables -> System Variables -> Open "Path" for editing.

Note: If the machine has more than one MATLAB installed, the directory for the target platform must be ahead of any other MATLAB directory (for instance, when compiling a 64-bit application, the directory in the MATLAB 64-bit installation must be the first one on the PATH).

10. Build and execute the application.

 [Please provide feedback to help us improve this Solution](#)