mini-HOWTO: `mex` + Microsoft Visual Studio 2005 Express Edition

William Fox
willfox@mit.edu

1 Aug 2007

Abstract

This document describes how to setup to compile C-language mex files with the free Microsoft Visual C/C++ 2005 Express Edition compiler. You will compile your mex file using the Matlab command line tool `mex`, rather than creating projects and compiling within the Visual Studio IDE. The steps are as streamlined as possible and require no hacking on any of the `mex` setup files.

1 History

version 1.1 8 Dec 2006. Initial Revision. Covered `mex` + MSVC for Matlab R2006b.

version 1.2 1 Aug 2007. Revised to encompass Matlab R2007a. URL to get VC Express has been updated. Added reference to Mathworks website.

2 Introduction

Microsoft now distributes "Visual Studio 2005 Express Edition" for free, which includes the "Express Edition Visual C/C++" compiler (hereafter "VC Express"). This document grew out of an attempt to make VC Express work with Matlab 7.3 (R2006b). This required some trickery, because the Express Edition was not an officially supported compiler; only the Professional Edition ($$) was. With the release of Matlab 7.4, (R2007a), the Express Edition is now an officially supported compiler for `mex`. As a result of its new-found status, the install procedure is slightly different, so this document contains separate instructions based on your version of Matlab.

Why do we need a howto? The hitch is that, out of the box, VC Express can only generate .NET-based programs, not Win32—"regular" Windows—programs, and it does not come packaged with all of the standard windows library (.lib) files (kernel32.lib, user32.lib, etc). Now, even if you don't need to make Win32 calls from within your mex file, `mex` by

default asks the linker to link against several of these standard Win32 libraries, and will fail if it cannot find them. Thus, `mex` will not work with VC Express straight out of the box. Since we'd prefer to do zero hacking on the `mex` setup, we want those libraries.

Fortunately, Microsoft also distributes a "Windows Platform SDK," which contains the necessary library and header files as a free download. Once you have these files, and `mex` can find them, `mex` will be perfectly happy to compile using VC Express.

A final note: Matlab already comes bundled with a compiler (at least on Windows), called `lcc`. Try to use `lcc` first; it should work out of the box for many (most?) normal mex applications. (In our case, we were writing mex routines to control hardware, and for arcane reasons `lcc` could not compile the code. If there are other reasons to prefer VC, please let me know.) Also, the `lcc` distribution that comes with Matlab has the Win32 library and header files, but I was not successful in getting VC Express to use those. Ergo: you have to use the Platform SDK.

3 Instructions for Matlab 7.3 (R2006a, R2006b)

To get VC Express to work with R2006, we will download VC Express and the Platform SDK. The crucial step is installing the Platform SDK in the correct directory so that the whole thing looks to Matlab like a VC *Professional* install.

1. **Obtain and install Visual C++ 2005 Express Edition.** Browse to:

   `http://msdn.microsoft.com/vstudio/express/downloads`

   and choose to download "Visual C++ 2005" under Step 2. This will start the download of the file `vcsetup.exe`. Run this to start the web installer. I installed to the to the default directory (`C:\Program Files\Microsoft Visual Studio 8`)

   During my install, I chose *not* to install any optional features, including the IDE. `mex` needs only the command line tools.

2. **Obtain and install the Windows Platform SDK.**

   The download is called "Microsoft Platform SDK for Windows Server 2003 R2." Browse to:

   `http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/`

   This page is also linked from the bottom of the VC Express homepage. Follow the link on step 2. Get the file `PSDK-x86.exe`, which is the SDK web installer. Run it and choose to do a *Custom install.* This is necessary to install the SDK to the correct directory and to choose only a subset of the components to install.

   CRITICAL: choose the install directory as

C:\Program Files\Microsoft Visual Studio 8\VC\PlatformSDK

(Mind the lack-of-space in "PlatformSDK.") This is a directory where `mex` wants to look for both header files and linking, so installing it here will allow everything else to work out of the box.

Next, choose which components to install. For simplicity, you can include everything. See the end of this document for the minimum set of components to get `mex` to work.

3. `mex` **setup** Finally, we're home free. Within Matlab, run

```
>> mex -setup
```

If you let the routine search for compilers itself, it should find

```
[2] Microsoft Visual C/C++ version 8.0 in C:\Program Files\
Microsoft Visual Studio 8
```

(The number in brackets may vary.) Choose that one; `mex` is now set up to use VC Express.

4 Instructions for Matlab 7.4 (R2007a, and later?)

VC Express is now an officially supported compiler in R2007. As before, we start by downloading VC Express and the Platform SDK.

Note `mex` now knows about both the Professional Edition and the Express Edition. In its list of compilers, it calls the former `Microsoft Visual C++ 2005`, and the latter `Microsoft Visual C++ 2005 Express Edition`.

I have made VC Express work in two ways. One is to follow the directions exactly as for R2006, dropping the Platform SDK in the right place to mimic a Professional install. If you do so, be sure to specify during `mex -setup` that you have `Microsoft Visual C++ 2005` (i.e. the Professional Edition—in this case, we truly are fooling `mex` into thinking we have installed VC Professional.)

The preferred way to install, however, is to place the Platform SDK in its default directory, and define the environment variable `MSSDK` to point to that directory. Then, you tell `mex -setup` that you have installed `Microsoft Visual C++ 2005 Express Edition`. The instructions for this method follow.

1. **Obtain and install Visual C++ 2005 Express Edition.** Browse to:

```
http://msdn.microsoft.com/vstudio/express/downloads
```

and choose to download "Visual C++ 2005 Express Edition" under Step 2. This will start the download of the file `vcsetup.exe`. Run this to start the web installer. I installed to the to the default directory

```
C:\Program Files\Microsoft Visual Studio 8\
```

During my install, I chose *not* to install any optional features, including the IDE. `mex` needs only the command line tools.

2. **Obtain and install the Windows Platform SDK.**

   The download is called "Microsoft Platform SDK for Windows Server 2003 R2." Browse to:

   ```
   http://msdn.microsoft.com/vstudio/express/visualc/usingpsdk/
   ```

   This page is also linked from the bottom of the VC Express homepage. Follow the link on step 2 to get the file `PSDK-x86.exe`, which is the SDK web installer. Run it and choose to do a *Custom install.*

   Unlike for R2006, in R2007a it is not critical where the SDK is installed. For simplicity, choose the default directory. This should be:

   ```
   C:\Program Files\Microsoft Platform SDK for Windows
   Server 2003 R2\
   ```

   Next, choose which components to install. For simplicity, you can include everything. However, the full download is large. See the end of this document for the minimum set of components to get `mex` to work.

3. **Set the `MSSDK` environment variable**

   Set the environment variable by going to

   Start → Control Panel → System → Advanced (tab)
   → Environment (button)

   and then create a New System Variable, with name `MSSDK`, and value

   ```
   C:\Program Files\Microsoft Platform SDK for Windows
   Server 2003 R2\
   ```

   (or whichever directory you installed the Platform SDK to).

4. **`mex` setup**

   First, restart Matlab. It is necessary to do so whenever you change Windows environment variables. To confirm that the environment variable stuck, run:

   ```
   >> !set
   ```

   and check to see that `MSSDK` is on the list with the correct value. If so, you're home free. To setup `mex`, run:

```
>> mex -setup
```

Note, in R2007a, there seems to be a bug in `mex -setup`: it has trouble finding the Express Edition when automatically searching for compilers (at least for me, it misidentified the install as the Professional Edition). The solution is to ask `mex -setup` *not* to search for installed compilers. Then, in the long list choose:

```
[14] Microsoft Visual C++ 2005 Express Edition
```

(The bracketed number may vary.) `mex -setup` may complain that it cannot find the Express Edition; no matter: `mex` is now set up to use VC Express.

A Appendix: Minimum Platform SDK components for a functional mex

For a smaller download (and thus faster install), during the custom Platform SDK install include *only* the following top-level components:

√ Configuration Options

√ Microsoft Windows Core SDK

√ Microsoft Data Access Services SDK

(This was a surprise, but this is how you get `odbc32.lib` and `odbcbcp.lib`)

√ Debugging Tools for Windows

(No key libraries, but we should have these, right?)

All other items (e.g. Microsoft Web Workshop (IE) SDK) got an X at the highest level. Now, drilling down on the chosen components, X out the following sub-components to choose *not* to install

□ Configuration Options

× Register Environment Variables. (This is the default, in fact.)

□ MS Windows Core SDK:

× Tools → AMD 64-bit

× Tools → Intel 64-bit

× Build Environment → AMD 64-bit

× Build Environment → Intel 64-bit

- × Documentation
- × Sample Code
- × Source Code

□ Microsoft Data Access Services:

- × Tools → AMD 64-bit
- × Tools → Intel 64-bit
- × Build Environment → AMD 64-bit
- × Build Environment → Intel 64-bit
- × Documentation
- × Sample Code

This will give you the minimum set of libraries to run mex. Further, within the libraries, you are keeping only code for 32-bit Intels.

References

[1] Mathworks Tech-Notes: Matlab Supported Compilers. `http://www.mathworks.com/support/tech-notes/1600/1601.htm`.