


Building on Windows 10 using mingw32 #2751

New issue

Closed petobens opened this issue on Jan 6, 2016 · 10 comments



petobens commented on Jan 6, 2016

ValueError: Unknown MS Compiler version 1900

In order to attempt to solve this problem, following the discussion in [this](#) bug report, I applied the changes/patch in this [pull request](#). However when running again `python setup.py build -c mingw32` I now get the following output:

```
C:\Users\Pedro\Desktop\statsmodels>python setup.py build -c mingw32
Cythonizing sources
statsmodels\nonparametric\linbin.pyx has not changed
statsmodels\nonparametric\smoothers_lowess.pyx has not changed
statsmodels\tsa\kalmanf\kalman_loglike.pyx has not changed
statsmodels\tsa\statespace\statespace.pyx.in has not changed
running build
running build_py
copying statsmodels\version.py -> build\lib.win-amd64-3.5\statsmodels
package init file 'statsmodels\tsa\vector_ar\data\__init__.py' not found (or not a regular file)
running build_ext
Looking for python35.dll
Cannot build msvc library: "vcruntime140d.dll" not found
building 'statsmodels.nonparametric.linbin' extension
C compiler: gcc -g -DDEBUG -DMS_WIN64 -O0 -Wall -Wstrict-prototypes

compile options: '-DNPY_MINGW_USE_CUSTOM_MSVCRT -D_MSVCRT_VERSION_=0x1400 -IC:\prog-lang\python\lib
gcc -g -DDEBUG -DMS_WIN64 -O0 -Wall -Wstrict-prototypes -DNPY_MINGW_USE_CUSTOM_MSVCRT -D_MSVCRT_VERS
Found executable C:\prog-tools\msys2\mingw64\bin\gcc.exe
gcc -g -shared build\temp.win-amd64-3.5\Release\statsmodels\nonparametric\linbin.o -LC:\prog-lang\py
C:\prog-tools\msys2\mingw64\bin\../lib/gcc/x86_64-w64-mingw32/5.3.0/../../../../x86_64-w64-mingw32/t
collect2.exe: error: ld returned 1 exit status
error: Command "gcc -g -shared build\temp.win-amd64-3.5\Release\statsmodels\nonparametric\linbin.o -
```

I checked and there is `ucrtbase.dll` file in `C:\Windows\System32`. I am probably missing something important but I have no idea what is going on. Any help will be greatly appreciated. If, for whatever reason, it is not possible to build `statsmodels` dev on Windows, can someone point me to updated binaries? Thanks in advance!

Assignees

No one assigned

Labels

build

Projects

None yet

Milestone







No milestone


Notifications

Subscribe

You're not receiving notifications from this thread.

6 participants





josef-pkt commented on Jan 6, 2016

I haven't tried mingw with python 3.5 yet.

From your error message, you are missing `vcruntime140d.dll`

You could try Winpython which has a python 3.5 version and includes pymingw and the related dlls. For up to python 3.4 it worked without changes for me to build with the included mingw. Winpython doesn't need to be registered with Windows and just stays a standalone package without affecting other installed python versions.

There is a discussion on the mailing lists (numpy, scipy) that plain mingw32 might not work, and the effort goes into pymingw which will have all the necessary settings.

Related: I managed two days ago to build `statsmodels` master with the Microsoft compiler SDK and @JanSchulz is working on building wheels through appveyor. So, I hope that we soon will get "nightly" wheels back (at maybe irregular times)


Here are my wheels. Those are experimental for now (my first build with Microsoft SDK) and are build against the default numpy in each of the corresponding Winpython versions:
<https://drive.google.com/folderview?id=0B17GgFZDG6XMWVBQNldzZ0lybzg&usp=sharing>

If you try them out, please report back whether there were problems. It's possible that the redistributable runtime libraries are missing by default.

Author

+ @

...



petobens commented on Jan 6, 2016

Hi @josef-pkt thanks for the quick reply, explanation and links.

Following your suggestion and this [comment](#) in Winpython I installed the VC++ 2015 runtime redistributable but I still get the `missing vcruntime140d.dll` error message. According to this [comment](#) in reddit:

That DLL is the Visual C++ 2015 debug runtime. You don't normally link against the debug version of the runtime for things you're going to distribute, because that version is not on end user systems. Whoever built this component did something wrong

something is wrong (but I'm completely out of my league here to figure out what is actually wrong). Any ideas?


In the meantime I will try the wheels that you kindly provided and wait for the nightly appveyor build. I have one question though: since I'm using numpy 1.10.2 and, as far as I understand, your build is against numpy 1.9.2 (or 1.9.1), should I expect some sort of incompatibility?

Once again thanks for the help.

Member

+ @

...



josef-pkt commented on Jan 6, 2016

I also just started to figure this out a few days ago, using Windows 10 in a virtualbox. So, I don't know much of the details either. (I didn't know about the difference between debug and not-debug versions of the dll. I don't know what my wheel uses. Winpython has the non- d version "C:\programs\WinPython-64bit-3.5.1.1\python-3.5.1.amd64\vcruntime140.dll") Because I'm currently only using Winpython installs, I don't run into some of the problems and don't know about them. (I didn't manage to get a working scipy in a plain Python 3.5 install.)

About numpy:


We can use binaries that are compiled against older numpy versions with newer versions, but not the other way around.

Some details are in my answer here <http://stackoverflow.com/questions/17709641/valueerror-numpy-dtype-has-the-wrong-size-try-recompiling/18369312#18369312>

Member

+ @

...




petobens commented on Jan 6, 2016

Ok, thanks for the link! Feel free to close this issue whenever you like.

Author

+ @

...




josef-pkt commented on Jan 6, 2016

We keep this issue open until this is settled. We don't have currently another issue for Python 3.5 or Windows 10 open.


Member

+ @

...



josef-pkt added the build label on Jan 6, 2016



rgommers commented on Jan 6, 2016

mingw32 does not and will never work with Python 3.5, due to Python 3.5 moving to MSVC'15 and mingw32 being a dead project.


Mingw-w64 doesn't work with Python 3.5 now, but may start to work in the coming year. But don't try it now, a simple patch won't get you there. See <http://mingwpy.github.io/> for details if interested.

So for now: use MSVC or Intel compilers.

Member

+ @

...



josef-pkt commented on Jan 6, 2016


@rgommers Thanks for clarifying.

I just checked and saw that Winpython 3.5 does not include the `c/c++` compilers, in contrast to the other python versions.

Member


+ @

...



nonhermitian referenced this issue on Jun 12, 2016

Does not compile under MS Visual C #447



PetrovskYYY commented on Aug 25, 2016


Corrected similar problem by just commenting out all (two) lines with occurrences of `get_msvcrt()` in `cygwincompiler.py` (and one "else:" line also). After this I had successfully compiled and installed and used `marisa-trie` by pip on Windows 10/Python 3.5/mingw32:

```
pip install marisa-trie
Collecting marisa-trie
  Using cached marisa-trie-0.7.2.tar.gz
Installing collected packages: marisa-trie
Running setup.py install for marisa-trie ... done
Successfully installed marisa-trie-0.7.2
```

7

+ @

...




juanluj commented on Dec 20, 2016

PetrovskYYY this worked perfectly for me on Python 3.5 on Windows 7 and mingw32, thanks!

+ @

...




bashtage commented on May 8

Seems to be solved now.


Contributor

+ @

...



bashtage closed this on May 8



Write Preview

AA B I " " < > ↺ ⋮ ⋮ ⋮ @ ↻

Leave a comment

Attach files by dragging & dropping, selecting them.

Comment

© 2019 GitHub, Inc. Terms Privacy Security Status Help Contact GitHub Pricing API Training Blog About