**Yesterday**

**You said:**

I have setup.py like starting like this:from distutils.core import setup, Extension

Since distutils has been deprecated and replaced instead with setuptools, how I should change this line?

**Copilot said:**

Copilot

To update your setup.py file to use setuptools instead of the deprecated distutils, you can modify the import statement. Here's how you can change it:

from setuptools import setup, Extension

This change ensures that your setup script uses setuptools, which is the recommended tool for packaging Python projects. If you have any other questions or need further assistance, feel free to ask!

Edit in Pages

Copy

AI-generated content may be incorrect

**You said:**

Can you produce setup.cfg based on following setup.py file:from setuptools import setup, Extension

setup(name='rdrand',

version='1.5.0',

description="Python interface to Intel hardware rng",

long\_description= "".join(open('rdrandom.rst').readlines()),

author="Chris Stillson",

author\_email="stillson@gmail.com",

url='https://github.com/stillson/rdrand',

license="New BSD license",

ext\_modules=[Extension('\_rdrand', ['rdrand.c'])],

py\_modules = ['rdrand'],

keywords = ['rdrand', 'rdseed', "intel","hardware","random","number","generator","rng"],

data\_files=[('share', ['rdrandom.rst'])],

classifiers = ["Development Status :: 4 - Beta",

"Intended Audience :: Developers",

"Intended Audience :: Science/Research",

"License :: OSI Approved :: BSD License",

"Programming Language :: Python",

"Programming Language :: Python :: 2",

"Programming Language :: Python :: 3",

"Topic :: Security",

"Topic :: Security :: Cryptography",],

extras\_require={

'tests': [],

'docs': [

'sphinx >= 1.4',

'sphinx\_rtd\_theme']}

)

**Copilot said:**

Copilot

Sure! Here's how you can convert your setup.py file into a setup.cfg file:

[metadata]

name = rdrand

version = 1.5.0

description = Python interface to Intel hardware rng

long\_description = file: rdrandom.rst

author = Chris Stillson

author\_email = stillson@gmail.com

url = https://github.com/stillson/rdrand

license = New BSD license

keywords = rdrand, rdseed, intel, hardware, random, number, generator, rng

classifiers =

Development Status :: 4 - Beta

Intended Audience :: Developers

Intended Audience :: Science/Research

License :: OSI Approved :: BSD License

Programming Language :: Python

Programming Language :: Python :: 2

Programming Language :: Python :: 3

Topic :: Security

Topic :: Security :: Cryptography

[options]

py\_modules = rdrand

[options.extras\_require]

tests =

docs =

sphinx >= 1.4

sphinx\_rtd\_theme

[options.data\_files]

share = rdrandom.rst

[options.ext\_modules]

\_rdrand = rdrand.c

This setup.cfg file should be placed in the same directory as your setup.py file. It provides the same information in a declarative configuration format. If you have any further questions or need additional help, feel free to ask!

Edit in Pages

Copy

AI-generated content may be incorrect

**You said:**

IS this syntax correct? You have forgotten that in above:[build]

compiler = mingw32

**Copilot said:**

Copilot

You're right! I missed the [build] section. Here's the corrected setup.cfg file with the compiler option included:

[metadata]

name = rdrand

version = 1.5.0

description = Python interface to Intel hardware rng

long\_description = file: rdrandom.rst

author = Chris Stillson

author\_email = stillson@gmail.com

url = https://github.com/stillson/rdrand

license = New BSD license

keywords = rdrand, rdseed, intel, hardware, random, number, generator, rng

classifiers =

Development Status :: 4 - Beta

Intended Audience :: Developers

Intended Audience :: Science/Research

License :: OSI Approved :: BSD License

Programming Language :: Python

Programming Language :: Python :: 2

Programming Language :: Python :: 3

Topic :: Security

Topic :: Security :: Cryptography

[options]

py\_modules = rdrand

[options.extras\_require]

tests =

docs =

sphinx >= 1.4

sphinx\_rtd\_theme

[options.data\_files]

share = rdrandom.rst

[options.ext\_modules]

\_rdrand = rdrand.c

[build]

compiler = mingw32

This should now include the compiler option under the [build] section. If you have any more questions or need further adjustments, feel free to ask!

Edit in Pages

Copy

AI-generated content may be incorrect

**You said:**

replace with memcpy:bcopy((char\*)&rando, &data[num\_quads \* 8], num\_chars);

**Copilot said:**

Copilot

To replace the bcopy function with memcpy, you can use the following syntax:

memcpy(&data[num\_quads \* 8], (char\*)&rando, num\_chars);

The memcpy function copies num\_chars bytes from the source (char\*)&rando to the destination &data[num\_quads \* 8]. This should achieve the same result as the bcopy function you were using. If you have any more questions or need further assistance, feel free to ask!

Edit in Pages

Copy

AI-generated content may be incorrect

**You said:**

Why this error. I am building rdrand from github source with above changes you suggested:$ pip install .

Processing d:\programs\msys2\home\j1304\downloads\rdrand

Preparing metadata (setup.py) ... done

Building wheels for collected packages: rdrand

Building wheel for rdrand (setup.py) ... error

error: subprocess-exited-with-error

× python setup.py bdist\_wheel did not run successfully.

│ exit code: 1

╰─> [13 lines of output]

running bdist\_wheel

running build

running build\_py

creating build\lib.win-amd64-cpython-313

copying rdrand.py -> build\lib.win-amd64-cpython-313

running build\_ext

building '\_rdrand' extension

creating build\temp.win-amd64-cpython-313\Release

gcc -shared -O -Wall -ID:\Programs\Msys2\home\j1304\venv313\include -ID:\Programs\Python313\include -ID:\Programs\Python313\Include -c rdrand.c -o build\temp.win-amd64-cpython-313\Release\rdrand.o

D:\Programs\Msys2\tmp\ccctsIBk.s: Assembler messages:

D:\Programs\Msys2\tmp\ccctsIBk.s:381: Error: invalid instruction suffix for `push'

D:\Programs\Msys2\tmp\ccctsIBk.s:384: Error: invalid instruction suffix for `pop'

error: command 'D:\\Programs\\Msys2\\ucrt64\\bin\\gcc.EXE' failed with exit code 1

[end of output]

note: This error originates from a subprocess, and is likely not a problem with pip.

ERROR: Failed building wheel for rdrand

Running setup.py clean for rdrand

Failed to build rdrand

ERROR: Failed to build installable wheels for some pyproject.toml based projects (rdrand)

**Copilot said:**

Copilot

The error you're encountering, specifically the "invalid instruction suffix for push and pop," is likely due to an incompatibility between the assembly code in rdrand.c and the assembler being used by your GCC compiler. This issue can arise when the assembler does not recognize certain instruction suffixes, which might be specific to a different architecture or instruction set.

Here are a few steps you can try to resolve this issue:

1. **Check the Compiler and Assembler Compatibility**: Ensure that your GCC compiler and assembler are compatible with the assembly code in rdrand.c. You might need to use a different version of GCC or adjust the compiler flags.
2. **Modify the Assembly Code**: If you are comfortable with assembly language, you can modify the assembly code in rdrand.c to use instructions that are compatible with your assembler. For example, you might need to remove or replace the instruction suffixes that are causing the errors.
3. **Use a Different Build Environment**: Sometimes, using a different build environment can help. For example, you could try building the project on a different system or using a different version of MSYS2 or MinGW.
4. **Disable Assembler Optimizations**: You can try disabling assembler optimizations by adding the --disable-assembler flag to your build configuration. This might help if the issue is related to specific assembler optimizations.
5. **Specify the Correct Build Option**: Ensure that you are specifying the correct build option for your environment. For example, you might need to use --build=x86\_64-w64-mingw32 for a 64-bit MinGW-w64 environment

1

.

If you continue to face issues, you might want to reach out to the maintainers of the rdrand project on GitHub for further assistance. They might have more specific insights or patches for this issue.

1

: SourceForge discussion on invalid instruction suffix errors.

1Trouble building 17.01 from source on x64 - OpenWrt Forum

Edit in Pages

Copy

AI-generated content may be incorrect

**You said:**

Where to specify:Specify the Correct Build Option: Ensure that you are specifying the correct build option for your environment. For example, you might need to use --build=x86\_64-w64-mingw32 for a 64-bit MinGW-w64 environment1.

**Copilot said:**

Copilot

To specify the correct build option for your environment, you can add the --build flag to your setup.py command. Here's how you can do it:

python setup.py build --build=x86\_64-w64-mingw32

Alternatively, if you are using pip to install the package, you can pass the build option like this:

pip install . --global-option=build --global-option=--build=x86\_64-w64-mingw32

This should ensure that the build process uses the correct configuration for your MinGW-w64 environment. If you encounter any further issues or need additional help, feel free to ask!