

# Building python packages using AIXTOOLS python

Contributed by Michael Felt

My starting point is AIXL 5.3 TL7 - the TL/SP packaging that came out together with AIX 6.1.

There are some key differences - don't ask me what - but applications compiled on AIX 5.3 TL6 and earlier may have issues on AIX 5.3 TL7 and later. This process will also work with later versions of AIX.

## Starting point

So - fresh install, plus xlc compiler. And no "RPM" packages, other than what the rpm.rte fileset installs.

```
*****
*                                     *
*                                     *
* Welcome to AIX Version 5.3!          *
*                                     *
*                                     *
* Please see the README file in /usr/lpp/bos for information pertinent to *
* this release of the AIX Operating System.          *
*                                     *
*                                     *
*****
```

```
root@x065:[/]rpm -qa
AIX-rpm-5.3.7.0-1
root@x065:[/]
```

Initially - nothing from AIXTOOLS - so first I'll get python installed. See <http://www.aixtools.net/index.php/python>

```
+-----+
|           |
| Installing Software... |
|           |
+-----+
```

```
installp: APPLYING software for:
aixtools.python.rte 2.7.12.3
aixtools.python.man.en_US 2.7.12.3
```

Restoring files, please wait.

4438 files restored.

+----- ensurepip: embedded install of pip and setuptools -----+

Ignoring indexes: <https://pypi.python.org/simple>

Collecting setuptools

Collecting pip

Installing collected packages: setuptools, pip

Successfully installed pip-8.1.1 setuptools-20.10.1

Finished processing all filesets. (Total time: 50 secs).

+-----+

Summaries:

+-----+

#### Installation Summary

-----

Name	Level	Part	Event	Result
aixtools.python.rte	2.7.12.3	USR	APPLY	SUCCESS
aixtools.python.man.en_US	2.7.12.3	USR	APPLY	SUCCESS

AIXTOOLS python comes with pip ready to go!

Installing the man pages is optional. BUT, notice that unlike the BULL/Perzl/IBM packaging pip is available.

To build a package such as cifs - the easy starting point is to let pip download the packages.

Note:  
before you begin - make sure your filesystems are large enough as pip  
and

make do not enlarge your filesystems automatically. My defaults are  
very small:

root@x065:[/]df -g

Filesystem	GB	blocks	Free	%Used	lused	%lused	Mounted on
/dev/hd4	0.12	0.11	9%	1104	4%	/	
/dev/hd2	0.88	0.13	86%	13601	31%	/usr	
/dev/hd9var	0.12	0.12	7%	317	2%	/var	
/dev/hd3	0.12	0.12	2%	46	1%	/tmp	
/dev/hd1	0.12	0.12	1%	5	1%	/home	
/proc	-	-	-	-	-	/proc	
/dev/hd10opt	0.25	0.10	59%	5733	19%	/opt	

This generally gives me enough room to build stuff.

```
root@x065:[/]chfs -a size=2G /opt
Filesystem size changed to 4194304
root@x065:[/]chfs -a size=1G /tmp
Filesystem size changed to 2097152
root@x065:[/]chfs -a size=1G /var
Filesystem size changed to 2097152
root@x065:[/]chfs -a size=512M /
Filesystem size changed to 1048576
```

Also make sure your umask is not too strict

```
root@x065:[/]umask
022
```

Ready to start:

First we test that pip is able to talk to the outside and see what python dependancies exist, if any.

```
root@x065:[/]mkdir /tmp/build
root@x065:[/]cd /tmp/build
```

```
root@x065:[/tmp/build]pip download cffi
```

```
Collecting cffi
```

```
Downloading cffi-1.10.0.tar.gz (418kB)
```

```
100% |#####| 419kB 81kB/s
```

```
Saved ./cffi-1.10.0.tar.gz
```

```
Collecting pycparser (from cffi)
```

```
Downloading pycparser-2.18.tar.gz (245kB)
```

```
100% |#####| 256kB 678kB/s
```

```
Saved ./pycparser-2.18.tar.gz
```

```
Successfully downloaded cffi pycparser
```

You are using pip version 8.1.1, however version 9.0.1 is available.

You should consider upgrading via the 'pip install --upgrade pip' command.

```
Upgrade pip
```

Using pip to upgrade itself is a good test that everything is working - and it gets rid of this message!

```
root@x065:[/tmp/build]pip install --upgrade pip
```

```
Collecting pip
```

```
Downloading pip-9.0.1-py2.py3-none-any.whl (1.3MB)
```

```
100% |#####| 1.3MB 158kB/s
```

```
Installing collected packages: pip
```

```
Found existing installation: pip 8.1.1
```

```
Uninstalling pip-8.1.1:
```

```
Successfully uninstalled pip-8.1.1
```

```
Successfully installed pip-9.0.1
```

Well, that did not actually test everything - because the C compiler was not needed.

So, next try - let's first build the cffi dependency - pycparser.

```
root@x065:[/tmp/build]pip build pycparser
```

ERROR: unknown command "build"

Unlike CPAN (think perl) pip(2) does not have a build command - so we just try install.

```
root@x065:[/tmp/build]pip install pycparser
```

Collecting pycparser

Using cached pycparser-2.18.tar.gz

Installing collected packages: pycparser

Running setup.py install for pycparser ... done

Successfully installed pycparser-2.18

Again

- no C compiler needed - so all is good. However, the next step is going to fail because cffi has 'other' dependencies, such as libffi.

But let's pretend we did not know about that!

```
root@x065:[/tmp/build]pip install cffi
```

Collecting cffi

Using cached cffi-1.10.0.tar.gz

Requirement already satisfied: pycparser in /opt/lib/python2.7/site-packages (from cffi)

Installing collected packages: cffi

Running setup.py install for cffi ... error

```
Complete output from command /opt/bin/python -u -c "import setuptools, tokenize;__file__='/tmp/pip-build-3pvK6g/cffi/setup.py';f=getattr(tokenize, 'open', open)(__file__);code=f.read().replace('\r\n', '\n');f.close();exec(compile(code, __file__, 'exec'))" install --record /tmp/pip-eaJIQ0-record/install-record.txt --single-version-externally-managed --compile:
```

```
"_configtest.c",
line 1.1: 1506-731 (S) The '__thread' keyword is not supported on the
target platform. The keyword is ignored.
```

Note: will not use '\_\_thread' in the C code

\*\*\*\*\* The above error message can be safely ignored.

Id: 0711-736 ERROR: Input file \_configtest.o:

XCOFF64 object files are not allowed in 32-bit mode.

Note: will not use '\_\_sync\_synchronize()' in the C code

\*\*\*\*\* The above error message can be safely ignored.

running install

running build

running build\_py

creating build

creating build/lib.aix-5.3-2.7

creating build/lib.aix-5.3-2.7/ffi

copying ffi/vengine\_cpy.py -> build/lib.aix-5.3-2.7/ffi

...

creating build/temp.aix-5.3-2.7/c

```
xc_r -I/opt/include -O3 -qmaxmem=-1 -qarch=pwr5 -q64
-I/opt/builaix/includes -DNDEBUG -I/usr/include/ffi
-I/usr/include/libffi -I/opt/include/python2.7 -c c/_cffi_backend.c -o
build/temp.aix-5.3-2.7/c/_cffi_backend.o
```

"c/\_cffi\_backend.c", line 15.10: 1506-296 (S) #include file <ffi.h> not found.

"c/malloc\_closure.h", line 6.10: 1506-296 (S) #include file <ffi.h> not found.

"c/malloc\_closure.h", line 81.5: 1506-046 (S) Syntax error.

...

"c/\_cffi\_backend.c", line 4923.26: 1506-098 (E) Missing argument(s).

error: command 'xc\_r' failed with exit status 1

-----

Command

```
"/opt/bin/python -u -c "import setuptools,
tokenize;__file__='/tmp/pip-build-3pvK6g/ffi/setup.py';f=getattr(tokenize,
'open', open)(__file__);code=f.read().replace('\r\n',
'\n');f.close();exec(compile(code, __file__, 'exec'))" install --record
/tmp/pip-eaJIQ0-record/install-record.txt
--single-version-externally-managed --compile" failed with error code 1
in /tmp/pip-build-3pvK6g/ffi/
```

What can I learn from all those messages?

OK. This failed. HOWEVER - there are some important bits in the error messages.

-  
ld: 0711-736 ERROR: Input file \_configtest.o:

XCOFF64 object files are not allowed in 32-bit mode.

&diam; The aixtools version of python is compiled in 64-bit mode, so you need to "export OBJECT\_MODE=64" before running 'pip install'

```
- xlc_r -l/opt/include -O3 -qmaxmem=-1 -qarch=pwr5 -q64
-l/opt/buildaix/includes -DNDEBUG -l/usr/include/ffi
-l/usr/include/libffi -l/opt/include/python2.7
```

&diam; the command to build packages is practically hardwired into the packaging. KNOW that you can change these settings in the file /opt/lib/python2.7/\_sysconfigdata.py

-  
"c/\_cffi\_backend.c", line 15.10: 1506-296 (S) #include file <ffi.h> not found.

&diam; libffi is missing. Either find and download, and compile libffi - or install aixtools.libffi (at <http://download.aixtools.net/tools/aixtools.libffi.3.2.1.1.I>)

## Installation Summary

```
-----
```

Name	Level	Part	Event	Result
aixtools.libffi.share	3.2.1.1	USR	APPLY	SUCCESS
aixtools.libffi.rte	3.2.1.1	USR	APPLY	SUCCESS
aixtools.libffi.man.en_US	3.2.1.1	USR	APPLY	SUCCESS

Again man pages are optional.

For "just in case" my packaging has both the 32-bit and 64-bit libraries. And the include file is in /opt/include.

```
root@x065:[/opt/lib/python2.7]slpp -f aixtools.libffi.rte
```

```
Fileset      File
```

```
-----
```

Path: /usr/lib/objrepos

aixtools.libffi.rte 3.2.1.1

```
/opt/lib/pkgconfig/libffi.pc
/opt/include
/opt/lib/pkgconfig
/opt/include/ffitarget.h
/opt/lib/libffi.a
/opt/lib
/opt/include/ffi.h
```

```
root@x065:[/opt/lib/python2.7]ls -l /usr/include/*ffi*
```

```
ls: 0653-341 The file /usr/include/*ffi* does not exist.
```

And,  
as the (nerly) hardwired xlc command does not have /opt/include (it has /opt/buildaix/includes) - I edit and update the file /opt/lib/python2.7/\_sysconfigdata.py

And, now try again...

There are several compiler warnings - I am choosing to ignore - but a loader error catches me:

```
/opt/lib/python2.7/config/ld_so_aix xlc_r
-bl:/opt/lib/python2.7/config/python.exp
build/temp.aix-5.3-2.7/c/_cffi_backend.o -lffi -o
build/lib.aix-5.3-2.7/_cffi_backend.so
```

```
ld: 0706-006 Cannot find or open library file: -l ffi
```

```
ld:open(): A file or directory in the path name does not exist.
```

```
error: command '/opt/lib/python2.7/config/ld_so_aix' failed with exit status 255
```

So,  
again - my first thought is to edit the /opt/lib/python2.7/\_sysconfigdata.py and add /opt/lib to the LDFLAGS definitions. However, that is already there.

Sadly, it seems there is an error in how that ld\_so\_aix command is called. So, to "make it happen" - the quick path is to also export LDFLAGS.



```
root@x065:[/opt/lib/python2.7]export LD_FLAGS="-L/opt/lib"
```

```
root@x065:[/opt/lib/python2.7]pip install cffi
```

```
Collecting cffi
```

```
Using cached cffi-1.10.0.tar.gz
```

```
Requirement already satisfied: pycparser in ./site-packages (from cffi)
```

```
Installing collected packages: cffi
```

```
Running setup.py install for cffi ... done
```

```
Successfully installed cffi-1.10.0
```

Closing

So, I hope this helps - and just enough wandering so that you will be able to resolve future issues on your own!

p.s. - don't forget to verify the install is really there.

```
root@x065:[/opt/lib/python2.7]pip list --format=columns
```

```
Package  Version
```

```
-----
```

```
cffi     1.10.0
```

```
pip      9.0.1
```

```
pycparser 2.18
```

```
setuptools 20.10.1
```