compileall — Byte-compile Source Files

Purpose: Convert source files to byte-compiled version.

The `compileall` module finds Python source files and compiles them to the byte-code representation, saving the results in `.pyc`.

Compiling One Directory

`compile_dir()` is used to recursively scan a directory and byte-compile the files within it.

```
# compileall_compile_dir.py

import compileall
import glob


def show(title):
    print(title)
    for filename in glob.glob('examples/**',
                              recursive=True):
        print('  {}'.format(filename))
    print()


show('Before')

compileall.compile_dir('examples')

show('\nAfter')
```

By default, all of the subdirectories are scanned to a depth of 10. The output files are written to a __pycache__ directory and named based on the Python interpreter version.

```
$ python3 compileall_compile_dir.py

Before
  examples/
  examples/README
  examples/a.py
  examples/subdir
  examples/subdir/b.py

Listing 'examples'...
Compiling 'examples/a.py'...
Listing 'examples/subdir'...
Compiling 'examples/subdir/b.py'...

After
  examples/
  examples/README
  examples/a.py
  examples/subdir
  examples/subdir/__pycache__
  examples/subdir/__pycache__/b.cpython-37.pyc
  examples/subdir/b.py
  examples/__pycache__
  examples/__pycache__/a.cpython-37.pyc
```

Ignoring Files

To filter directories out, use the `rx` argument to provide a regular expression to match the names to exclude.

```
# compileall_exclude_dirs.py

import compileall
import re

compileall.compile_dir(
    'examples',
    rx=re.compile(r'/subdir'),
)
```

This version excludes files in the `subdir` subdirectory.

```
$ python3 compileall_exclude_dirs.py

Listing 'examples'...
Compiling 'examples/a.py'...
Listing 'examples/subdir'...
```

The `maxlevels` argument controls the depth of recursion. For example, to avoid recursion entirely pass `0`.

```
# compileall_recursion_depth.py

import compileall
import re

compileall.compile_dir(
    'examples',
    maxlevels=0,
)
```

Only files within the directory passed to `compile_dir()` are compiled.

```
$ python3 compileall_recursion_depth.py

Listing 'examples'...
Compiling 'examples/a.py'...
```

Compiling sys.path

All of the Python source files found in sys.path can be compiled with a single call to `compile_path()`.

```
# compileall_path.py

import compileall
import sys

sys.path[:] = ['examples', 'notthere']
print('sys.path =', sys.path)
compileall.compile_path()
```

This example replaces the default contents of `sys.path` to avoid permission errors while running the script, but still illustrates the default behavior. Note that the `maxlevels` value defaults to `0`.

```
$ python3 compileall_path.py

sys.path = ['examples', 'notthere']
Listing 'examples'...
Compiling 'examples/a.py'...
Listing 'notthere'...
Can't list 'notthere'
```

Compiling Individual Files

To compile a single file, rather than an entire directory of files, use `compile_file()`.

```
# compileall_compile_file.py

import compileall
import glob


def show(title):
    print(title)
    for filename in glob.glob('examples/**',
                              recursive=True):
        print('  {}'.format(filename))
    print()


show('Before')

compileall.compile_file('examples/a.py')

show('\nAfter')
```

The first argument should be the name to the file, either a full path or a relative path.

```
$ python3 compileall_compile_file.py

Before
  examples/
  examples/README
  examples/a.py
  examples/subdir
  examples/subdir/b.py

Compiling 'examples/a.py'...

After
  examples/
  examples/README
  examples/a.py
  examples/subdir
  examples/subdir/b.py
  examples/__pycache__
  examples/__pycache__/a.cpython-37.pyc
```

From the Command Line

It is also possible to invoke `compileall` from the command line, so it can be integrated with a build system via a Makefile. For example:

```
$ python3 -m compileall -h

usage: compileall.py [-h] [-l] [-r RECURSION] [-f] [-q] [-b] [-d
DESTDIR]
                     [-x REGEXP] [-i FILE] [-j WORKERS]
                     [--invalidation-mode
{checked-hash,timestamp,unchecked-hash}]
                     [FILE|DIR [FILE|DIR ...]]

Utilities to support installing Python libraries.

positional arguments:
  FILE|DIR              zero or more file and directory names to
compile; if
                        no arguments given, defaults to the
equivalent of -l
                        sys.path

optional arguments:
  -h, --help            show this help message and exit
  -l                    don't recurse into subdirectories
  -r RECURSION          control the maximum recursion level. if
`-l` and `-r`
```

```
  -t and -f
                        options are specified, then `-r` takes
precedence.
  -f                    force rebuild even if timestamps are up
to date
  -q                    output only error messages; -qq will
suppress the
                        error messages as well.
  -b                    use legacy (pre-PEP3147) compiled file
locations
  -d DESTDIR            directory to prepend to file paths for
use in compile-
                        time tracebacks and in runtime
tracebacks in cases
                        where the source file is unavailable
  -x REGEXP             skip files matching the regular
expression; the regexp
                        is searched for in the full path of each
file
                        considered for compilation
  -i FILE               add all the files and directories listed
in FILE to
                        the list considered for compilation; if
"-", names are
                        read from stdin
  -j WORKERS, --workers WORKERS
                        Run compileall concurrently
  --invalidation-mode {checked-hash,timestamp,unchecked-hash}
                        How the pycs will be invalidated at
runtime
```

To recreate the earlier example, skipping the `subdir` directory, run:

```
$ python3 -m compileall -x '/subdir' examples

Listing 'examples'...
Compiling 'examples/a.py'...
Listing 'examples/subdir'...
```

See also

- Standard library documentation for compileall

tabnanny — Indentation validator

pyclbr — Class Browser

Quick Links
Compiling One Directory
Ignoring Files
Compiling sys.path
Compiling Individual Files
From the Command Line

This page was last updated 2018-12-09.

Navigation
tabnanny — Indentation validator
pyclbr — Class Browser

The output from all the example programs from PyMOTW-3 has been generated with Python 3.7.1, unless otherwise noted. Some of the features described here may not be available in earlier versions of Python.

Looking for examples for Python 2?

This Site

Module Index
Index

Other Writing

Blog
The Python Standard Library By Example