

# How do I set permissions (attributes) on a file in a ZIP file using Python's zipfile module?

 [stackoverflow.com/questions/434641/how-do-i-set-permissions-attributes-on-a-file-in-a-zip-file-using-pythons-zip](https://stackoverflow.com/questions/434641/how-do-i-set-permissions-attributes-on-a-file-in-a-zip-file-using-pythons-zip)

When I extract files from a ZIP file created with the Python `zipfile` module, all the files are not writable, read only etc.

The file is being created and extracted under Linux and Python 2.5.2.

As best I can tell, I need to set the `ZipInfo.external_attr` property for each file, but this doesn't seem to be documented anywhere I could find, can anyone enlighten me?

## 4 Answers

up vote 30  
down vote  
accepted

+50

This seems to work (thanks Evan, putting it here so the line is in context):

```
buffer = "path/filename.zip" # zip filename to write (or file-like
                              # object)
name = "folder/data.txt"     # name of file inside zip
bytes = "blah blah blah"    # contents of file inside zip

zip = zipfile.ZipFile(buffer, "w", zipfile.ZIP_DEFLATED)
info = zipfile.ZipInfo(name)
info.external_attr = 0777 << 16L # give full access to included file
zip.writestr(info, bytes)
zip.close()
```

I'd still like to see something that documents this... An additional resource I found was a note on the Zip file format: <http://www.pkware.com/documents/casestudies/APPNOTE.TXT>

up vote 17  
down vote

[This link](#) has more information than anything else I've been able to find on the net. Even the zip source doesn't have anything. Copying the relevant section for posterity. This patch isn't really about documenting this format, which just goes to show how pathetic (read non-existent) the current documentation is.

```

# external_attr is 4 bytes in size. The high order two
# bytes represent UNIX permission and file type bits,
# while the low order two contain MS-DOS FAT file
# attributes, most notably bit 4 marking directories.
if node.isfile:
    zipinfo.compress_type = ZIP_DEFLATED
    zipinfo.external_attr = 0644 << 16L # permissions -r-wr--r--
    data = node.get_content().read()
    properties = node.get_properties()
    if 'svn:special' in properties and \
        data.startswith('link '):
        data = data[5:]
        zipinfo.external_attr |= 0120000 << 16L # symlink file
type
    zipinfo.compress_type = ZIP_STORED
    if 'svn:executable' in properties:
        zipinfo.external_attr |= 0755 << 16L # -rwxr-xr-x
    zipfile.writestr(zipinfo, data)
elif node.isdir and path:
    if not zipinfo.filename.endswith('/'):
        zipinfo.filename += '/'
    zipinfo.compress_type = ZIP_STORED
    zipinfo.external_attr = 040755 << 16L # permissions drwxr-xr-x
    zipinfo.external_attr |= 0x10 # MS-DOS directory flag
    zipfile.writestr(zipinfo, '')

```

Also, [this link](#) has the following. Here the low order byte presumably means the rightmost (lowest) byte of the four bytes. So this one is for MS-DOS and can presumably be left as zero otherwise.

#### *external file attributes: (4 bytes)*

The mapping of the external attributes is host-system dependent (see 'version made by'). For MS-DOS, the low order byte is the MS-DOS directory attribute byte. If input came from standard input, this field is set to zero.

Also, the source file `unix/unix.c` in the sources for InfoZIP's zip program, downloaded from [Debian's archives](#) has the following in comments.

```

/* lower-middle external-attribute byte (unused until now):
 *   high bit      => (have GMT mod/acc times) >>> NO LONGER USED! <<<
 *   second-high bit => have Unix UID/GID info
 * NOTE: The high bit was NEVER used in any official Info-ZIP release,
 *       but its future use should be avoided (if possible), since it
 *       was used as "GMT mod/acc times local extra field" flags in Zip
beta
 *       versions 2.0j up to 2.0v, for about 1.5 years.
 */

```

So taking all this together, it looks like only the second highest byte is actually used, at least for Unix.

EDIT: I asked about the Unix aspect of this on Unix.SX, in the question "[The zip format's external file attribute](#)". Looks like I got a couple of things wrong. Specifically both of the top two bytes are used for Unix.

---

[up vote](#) 0 [down vote](#) When you do it like this, does it work alright?

```
zf = zipfile.ZipFile("something.zip")
for name in zf.namelist():
    f = open(name, 'wb')
    f.write(self.read(name))
    f.close()
```

If not, I'd suggest throwing in an `os.chmod` in the for loop with 0777 permissions like this:

```
zf = zipfile.ZipFile("something.zip")
for name in zf.namelist():
    f = open(name, 'wb')
    f.write(self.read(name))
    f.close()
    os.chmod(name, 0777)
```

answered Jan 12 '09 at 7:15



[Evan Fosmark](#)

42.4k2585112

---

---

## Your Answer

### Sign up or [log in](#)

Sign up using Google

Sign up using Facebook

Sign up using Email and Password

### Post as a guest

---

By posting your answer, you agree to the [privacy policy](#) and [terms of service](#).

Not the answer you're looking for? Browse other questions tagged [python](#) [attributes](#) [zip](#) [file-permissions](#) [zipfile](#) or [ask your own question](#).