

python / cpython Public

<> Code Issues 5k+ Pull requests 2.2k Actions Projects Security and q

Commit 7ad3093



3 people authored on Mar 17 · ✖ 81 / 87 · Partially verified

```
[3.14] gh-141707: Skip TarInfo DIRTYE normalization during GNU long name
handling (GH-145819)

(cherry picked from commit 42d754e)

Co-authored-by: Seth Michael Larson <seth@python.org>
Co-authored-by: Eashwar Ranganathan <eashwar@eashwar.com>
```

1 parent [4601007](#) commit 7ad3093

4 files changed +47 -4 lines changed

↑ Top

Filter files...

- Lib
 - tarfile.py
- test
 - test_tarfile.py
- Misc
 - ACKS
 - NEWS.d/next/Library
 - 2025-11-18-06-35-53.gh-issue-141707.DBmQly.rst

4 files changed +47 -4 lines changed

Search within code

```
Lib/tarfile.py
@@ -1278,6 +1278,20 @@ def _create_pax_generic_header(cls, pax_headers,
type, encoding):
```



```

1365 + See dircheck documentation in _frombuf().
1366 + """
1346 1367 buf = tarfile.fileobj.read(BLOCKSIZE)
1347 - obj = cls.frombuf(buf, tarfile.encoding, tarfile.errors)
1368 + obj = cls._frombuf(buf, tarfile.encoding, tarfile.errors,
dircheck=dircheck)
1348 1369 obj.offset = tarfile.fileobj.tell() - BLOCKSIZE
1349 1370 return obj._proc_member(tarfile)
1350 1371
⋮
@@ -1402,7 +1423,7 @@ def _proc_gnulong(self, tarfile):
1402 1423
1403 1424 # Fetch the next header and process it.
1404 1425 try:
1405 - next = self.fromtarfile(tarfile)
1426 + next = self._fromtarfile(tarfile, dircheck=False)
1406 1427 except HeaderError as e:
1407 1428 raise SubsequentHeaderError(str(e)) from None
1408 1429
⋮
@@ -1537,7 +1558,7 @@ def _proc_pax(self, tarfile):
1537 1558
1538 1559 # Fetch the next header.
1539 1560 try:
1540 - next = self.fromtarfile(tarfile)
1561 + next = self._fromtarfile(tarfile, dircheck=False)
1541 1562 except HeaderError as e:
1542 1563 raise SubsequentHeaderError(str(e)) from None
1543 1564
⋮

```

Lib/test/test_tarfile.py

```

⋮
@@ -1234,6 +1234,25 @@ def test_longname_directory(self):
1234 1234 self.assertIsNotNone(tar.getmember(longdir))
1235 1235
self.assertIsNotNone(tar.getmember(longdir.removesuffix('/')))
1236 1236
1237 + def test_longname_file_not_directory(self):
1238 + # Test reading a longname file and ensure it is not handled as a
directory

```

```

1239 + # Issue #141707
1240 + buf = io.BytesIO()
1241 + with tarfile.open(mode='w', fileobj=buf, format=self.format) as tar:
1242 +     ti = tarfile.TarInfo()
1243 +     ti.type = tarfile.AREGTYPE
1244 +     ti.name = ('a' * 99) + '/' + ('b' * 3)
1245 +     tar.addfile(ti)
1246 +
1247 +     expected = {t.name: t.type for t in tar.getmembers()}
1248 +
1249 +     buf.seek(0)
1250 +     with tarfile.open(mode='r', fileobj=buf) as tar:
1251 +         actual = {t.name: t.type for t in tar.getmembers()}
1252 +
1253 +     self.assertEqual(expected, actual)
1254 +
1255 +

```

```

1237 1256 class GNUReadTest(LongnameTest, ReadTest, unittest.TestCase):

```

```

1238 1257

```

```

1239 1258     subdir = "gnu"

```



▼ Misc/ACKS



```
@@ -1535,6 +1535,7 @@ Ashwin Ramaswami
```

```
1535 1535 Jeff Ramnani
```

```
1536 1536 Grant Ramsay
```

```
1537 1537 Bayard Randel
```

```
1538 + Eashwar Ranganathan
```

```
1538 1539 Varpu Rantala
```

```
1539 1540 Brodie Rao
```

```
1540 1541 Rémi Rampin
```



▼ ...5-11-18-06-35-53.gh-issue-141707.DBmQIy.rst



```
... @@ -0,0 +1,2 @@
```

```

1 + Don't change :class:`tarfile.TarInfo` type from ``AREGTYPE`` to ``DIRTYE`` when
  parsing

```

```

2 + GNU long name or link headers.

```

Comments 0