

python / cpython Public

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

# Commit 9a23b75



3 people authored 18 hours ago · 69 / 73 · Partially verified

```
[3.11] gh-141707: Skip TarInfo DIRTYE normalization during GNU long name
handling (#145815)

gh-141707: Skip TarInfo DIRTYE normalization during GNU long name handling
(cherry picked from commit 42d754e)

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

3.11 (#98846, #145815)

1 parent 69ddd9b commit 9a23b75

4 files changed +47 -4

↑ 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

Search within code ⚙️

```

Lib/tarfile.py
@@ -1245,6 +1245,20 @@ def _create_pax_generic_header(cls, pax_headers,
type, encoding):
    @classmethod
    def frombuf(cls, buf, encoding, errors):
        """Construct a TarInfo object from a 512 byte bytes object.
+
+       To support the old v7 tar format AREGTYPE headers are
+       transformed to DIRTYPE headers if their name ends in '/'.
+       """
+       return cls._frombuf(buf, encoding, errors)
+
+       @classmethod
+       def _frombuf(cls, buf, encoding, errors, *, dircheck=True):
+           """Construct a TarInfo object from a 512 byte bytes object.
+
+           If ``dircheck`` is set to ``True`` then ``AREGTYPE`` headers will
+           be normalized to ``DIRTYPE`` if the name ends in a trailing slash.
+           ``dircheck`` must be set to ``False`` if this function is called
+           on a follow-up header such as ``GNUTYPE_LONGNAME``.
+           """
+           if len(buf) == 0:
+               raise EmptyHeaderError("empty header")
@@ -1275,7 +1289,7 @@ def frombuf(cls, buf, encoding, errors):
    # Old V7 tar format represents a directory as a regular
    # file with a trailing slash.
    if obj.type == AREGTYPE and obj.name.endswith("/"):
+       if dircheck and obj.type == AREGTYPE and obj.name.endswith("/"):
            obj.type = DIRTYPE
    # The old GNU sparse format occupies some of the unused
@@ -1310,8 +1324,15 @@ def fromtarfile(cls, tarfile):
    """Return the next TarInfo object from TarFile object
    tarfile.
    """
+       return cls._fromtarfile(tarfile)

```

```

1329 + @classmethod
1330 + def _fromtarfile(cls, tarfile, *, dircheck=True):
1331 +     """
1332 +     See dircheck documentation in _frombuf().
1333 +     """
1313 1334         buf = tarfile.fileobj.read(BLOCKSIZE)
1314 -         obj = cls.frombuf(buf, tarfile.encoding, tarfile.errors)
1335 +         obj = cls._frombuf(buf, tarfile.encoding, tarfile.errors,
1336 +                             dircheck=dircheck)
1315 1336         obj.offset = tarfile.fileobj.tell() - BLOCKSIZE
1316 1337         return obj._proc_member(tarfile)
1317 1338
@@ -1369,7 +1390,7 @@ def _proc_gnulong(self, tarfile):
1369 1390
1370 1391         # Fetch the next header and process it.
1371 1392         try:
1372 -             next = self.fromtarfile(tarfile)
1393 +             next = self._fromtarfile(tarfile, dircheck=False)
1373 1394         except HeaderError as e:
1374 1395             raise SubsequentHeaderError(str(e)) from None
1375 1396
@@ -1504,7 +1525,7 @@ def _proc_pax(self, tarfile):
1504 1525
1505 1526         # Fetch the next header.
1506 1527         try:
1507 -             next = self.fromtarfile(tarfile)
1528 +             next = self._fromtarfile(tarfile, dircheck=False)
1508 1529         except HeaderError as e:
1509 1530             raise SubsequentHeaderError(str(e)) from None
1510 1531

```

Lib/test/test\_tarfile.py



```

@@ -1105,6 +1105,25 @@ def test_longname_directory(self):
1105 1105         self.assertIsNotNone(tar.getmember(longdir))
1106 1106
1107 1107         self.assertIsNotNone(tar.getmember(longdir.removesuffix('/')))

```

```

1108 +     def test_longname_file_not_directory(self):
1109 +         # Test reading a longname file and ensure it is not handled as a
           directory
1110 +         # Issue #141707
1111 +         buf = io.BytesIO()
1112 +         with tarfile.open(mode='w', fileobj=buf, format=self.format) as tar:
1113 +             ti = tarfile.TarInfo()
1114 +             ti.type = tarfile.AREGTYPE
1115 +             ti.name = ('a' * 99) + '/' + ('b' * 3)
1116 +             tar.addfile(ti)
1117 +
1118 +             expected = {t.name: t.type for t in tar.getmembers()}
1119 +
1120 +             buf.seek(0)
1121 +             with tarfile.open(mode='r', fileobj=buf) as tar:
1122 +                 actual = {t.name: t.type for t in tar.getmembers()}
1123 +
1124 +             self.assertEqual(expected, actual)
1125 +
1126 +

```

```

1108 1127     class GNUReadTest(LongnameTest, ReadTest, unittest.TestCase):
1109 1128
1110 1129         subdir = "gnu"

```



▼ Misc/ACKS



@@ -1462,6 +1462,7 @@ Dhushyanth Ramasamy

1462 1462 Ashwin Ramaswami

1463 1463 Jeff Ramnani

1464 1464 Bayard Randel

1465 + Eashwar Ranganathan

1465 1466 Varpu Rantala

1466 1467 Brodie Rao

1467 1468 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 ```DIRTYPE``` when parsing
- 2 + GNU long name or link headers.

**Comments** 0