

python / cpython Public

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

Commit 72dde10



3 people authored 18 hours ago · 61 / 62 · Partially verified

```
[3.10] gh-141707: Skip TarInfo DIRTYE normalization during GNU long name
handling (#145816)

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>
```

3.10 (#145816)

1 parent 876858c commit 72dde10

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
@@ -1244,6 +1244,20 @@ 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 DIRTYE 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 ``DIRTYE`` 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")

@@ -1274,7 +1288,7 @@ 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 = DIRTYE

    # The old GNU sparse format occupies some of the unused

@@ -1309,8 +1323,15 @@ def fromtarfile(cls, tarfile):
    """Return the next TarInfo object from TarFile object
    tarfile.
    """
    return cls._fromtarfile(tarfile)

```

```

1328 + @classmethod
1329 + def _fromtarfile(cls, tarfile, *, dircheck=True):
1330 +     """
1331 +     See dircheck documentation in _frombuf().
1332 +     """
1312 1333         buf = tarfile.fileobj.read(BLOCKSIZE)
1313 -         obj = cls.frombuf(buf, tarfile.encoding, tarfile.errors)
1334 +         obj = cls._frombuf(buf, tarfile.encoding, tarfile.errors,
1335 +                             dircheck=dircheck)
1314 1335         obj.offset = tarfile.fileobj.tell() - BLOCKSIZE
1315 1336         return obj._proc_member(tarfile)
1316 1337
@@ -1368,7 +1389,7 @@ def _proc_gnulong(self, tarfile):
1368 1389
1369 1390         # Fetch the next header and process it.
1370 1391         try:
1371 -             next = self.fromtarfile(tarfile)
1392 +             next = self._fromtarfile(tarfile, dircheck=False)
1372 1393         except HeaderError as e:
1373 1394             raise SubsequentHeaderError(str(e)) from None
1374 1395
@@ -1503,7 +1524,7 @@ def _proc_pax(self, tarfile):
1503 1524
1504 1525         # Fetch the next header.
1505 1526         try:
1506 -             next = self.fromtarfile(tarfile)
1527 +             next = self._fromtarfile(tarfile, dircheck=False)
1507 1528         except HeaderError as e:
1508 1529             raise SubsequentHeaderError(str(e)) from None
1509 1530

```

Lib/test/test_tarfile.py



```

@@ -1037,6 +1037,25 @@ def test_longname_directory(self):
1037 1037         self.assertIsNotNone(tar.getmember(longdir))
1038 1038
1039 1039         self.assertIsNotNone(tar.getmember(longdir.removesuffix('/')))

```

```

1040 +     def test_longname_file_not_directory(self):
1041 +         # Test reading a longname file and ensure it is not handled as a
            directory
1042 +         # Issue #141707
1043 +         buf = io.BytesIO()
1044 +         with tarfile.open(mode='w', fileobj=buf, format=self.format) as tar:
1045 +             ti = tarfile.TarInfo()
1046 +             ti.type = tarfile.AREGTYPE
1047 +             ti.name = ('a' * 99) + '/' + ('b' * 3)
1048 +             tar.addfile(ti)
1049 +
1050 +             expected = {t.name: t.type for t in tar.getmembers()}
1051 +
1052 +             buf.seek(0)
1053 +             with tarfile.open(mode='r', fileobj=buf) as tar:
1054 +                 actual = {t.name: t.type for t in tar.getmembers()}
1055 +
1056 +             self.assertEqual(expected, actual)
1057 +
1058 +

```

```

1040 1059     class GNUReadTest(LongnameTest, ReadTest, unittest.TestCase):
1041 1060
1042 1061         subdir = "gnu"

```



▼ Misc/ACKS



@@ -1440,6 +1440,7 @@ Dhushyanth Ramasamy

1440 1440 Ashwin Ramaswami

1441 1441 Jeff Ramnani

1442 1442 Bayard Randel

1443 + Eashwar Ranganathan

1443 1444 Varpu Rantala

1444 1445 Brodie Rao

1445 1446 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