QGIS Application - Bug report #14640
PyQGIS Creating empty geometry from WKT does not work as expected

2016-04-08 04:04 AM - Stephen Beattie

Status:	Open		
Priority:	Normal		
Assignee:			
Category:	Python bindings / sipify		
Affected QGIS version:	3.4.5	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	22604

Description

As I understand from section 7 of http://www.opengeospatial.org/standards/sfa, empty geometries in WKT format should be specified as, for example "MultiLineString EMPTY".

Using QgsGeometry.fromWkt() works as expected in QGIS 2.8.3, but not in QGIS 2.14.1. See Python console output below:

QGIS 2.8.3 Wien:

> > > *g = QgsGeometry.fromWkt('MultiLineString EMPTY')*
> > > *print g*

<qgis._core.QgsGeometry object at 0x0000000018DC0BF8>

> > > *g.exportToWkt()*

u'MULTILINESTRING'

> > > *g = QgsGeometry.fromWkt('MultiLineString')*
> > > *print g*

None

> > > *g.exportToWkt()*

Traceback (most recent call last):
  File "<input>", line 1, in <module>
AttributeError: 'NoneType' object has no attribute 'exportToWkt'

QGIS 2.14.1 Essen:

> > > *g = QgsGeometry.fromWkt('MultiLineString EMPTY')*
> > > *print g*

None

> > > *g.exportToWkt()*

Traceback (most recent call last):
  File "<input>", line 1, in <module>
AttributeError: 'NoneType' object has no attribute 'exportToWkt'

> > > *g = QgsGeometry.fromWkt('MultiLineString')*
> > > *print g*

<qgis._core.QgsGeometry object at 0x0000000018EA3E18>

> > > *g.exportToWkt()*

u'MultiLineString ()'

Associated revisions

Revision 7d648e5b - 2018-12-18 09:11 AM - Nyall Dawson

Improve Python *repr* handling for null geometries

Also avoid massive long *repr* strings for complex geometries,
as these can flood the Python console (and first aid plugin),
and aren't useful for debugging anyway.

Refs #14640

Revision c339c360 - 2018-12-20 05:33 AM - Nyall Dawson

Improve Python *repr* handling for null geometries

Also avoid massive long *repr* strings for complex geometries,
as these can flood the Python console (and first aid plugin),
and aren't useful for debugging anyway.

Refs #14640

(cherry picked from commit 7d648e5b51d90a2e559dd97abc5682a51b2a74a1)

History

#1 - 2016-04-09 07:27 AM - Giovanni Manghi

- Category set to Python plugins

#2 - 2017-05-01 01:05 AM - Giovanni Manghi

- Regression? set to No
- Easy fix? set to No

#3 - 2018-12-07 01:41 PM - Johannes Kroeger

Different but still weird in QGIS 3:

```
>>> g = QgsGeometry.fromWkt('MultiLineString EMPTY')
>>> print(g)
<QgsGeometry: >
>>> g.asWkt()
''
>>> g.type()
3

>>> g = QgsGeometry.fromWkt('MultiLineString')
>>> print(g)
<QgsGeometry: MultiLineString ()>
>>> g.asWkt()
'MultiLineString ()'
>>> g.type()
1
```

#4 - 2018-12-07 01:49 PM - Johannes Kroeger

And different for Points (empty string twice) but not for LineStrings. Weird!

```
>>> p = QgsGeometry.fromWkt('POINT EMPTY')
>>> p.isEmpty()
True
>>> p.asWkt()
''

>>> p = QgsGeometry.fromWkt('POINT')
>>> p.isEmpty()
True
>>> p.asWkt()
''

>>> p = QgsGeometry.fromWkt('LineString')
>>> p.isEmpty()
True
>>> p.asWkt()
'LineString ()'

>>> p = QgsGeometry.fromWkt('LineString EMPTY')
>>> p.isEmpty()
True
>>> p.asWkt()
''
```

I am pretty sure that the case of "GeometryType" without an "EMPTY" or coordinates should not be allowed at all. Instead this should fail.

#5 - 2018-12-10 11:54 AM - Alessandro Pasotti

- Description updated

What exactly do you mean with "it should fail"? Should it return True from isNull() ?

(note that we are not throwing exceptions from core classes in QGIS API).

#6 - 2018-12-10 01:10 PM - Giovanni Manghi

- Status changed from Open to Feedback

#7 - 2018-12-17 06:38 PM - Johannes Kroeger

I haven't used the API much so no idea about how methods react if they are not successful.

If isNull is what QGIS uses to signal fail, then that should return true. But as a Python user I would expect a ValueError or something more specific (ParserError?) for QgsGeometry.fromWkt('LineString').

#8 - 2018-12-17 11:48 PM - Nyall Dawson

> *But as a Python user I would expect a ValueError or something more specific (ParserError?) for QgsGeometry.fromWkt('LineString').*

That's the plan for 4.0 (See https://github.com/qgis/qgis4.0_api/issues/85). But it's an API break (existing plugins may depend on the current behaviour of not throwing exceptions), so has to wait for 4.0.

#9 - 2018-12-18 12:03 AM - Nyall Dawson

To explain your 3.0 testing:

> > > *g = QgsGeometry.fromWkt('MultiLineString EMPTY')*
> > > *print(g)*

<QgsGeometry: >

The invalid WKT here has resulted in a null geometry - I'm going to improve the repr string here to reflect that, as this should be something more helpful like <QgsGeometry: null>

> > > *g.asWkt()*

''

> > > *g.type()*

3

QgsGeometry.type() returns a QgsWkbTypes.GeometryType, not a WKB type. From the api docs a GeometryType of 3 = UnknownGeometry. So this is correct.

> > > *g = QgsGeometry.fromWkt('MultiLineString')*
> > > *print(g)*

<QgsGeometry: MultiLineString ()>

> > > *g.asWkt()*

'MultiLineString ()'

> > > *g.type()*

1

This looks correct to me, except that the WKT output should really be "MultiLineString EMPTY".

#10 - 2018-12-19 07:10 PM - Stephen Beattie

If the WKT output for an empty MultiLineString should be "MultiLineString EMPTY", should QgsGeometry.fromWKT('MultiLineString EMPTY') not return a valid MultiLineString geometry, as it did back in QGIS 2.8.3?

My understanding of the WKT specification is still that it requires "EMPTY" to be included in these cases, and would therefore expect "MultiLineString EMPTY" to return a valid geometry, even if "MultiLineString" (without the " EMPTY") is also accepted.

In any case, to work around the original problem reported, my plugin first attempts to create a geometry with WKT 'MultiLineString EMPTY'. If None is returned, it then tries 'MultiLineString'.

#11 - 2018-12-19 11:29 PM - Nyall Dawson

Right - so ultimately I think there's two remaining issues here:

1. QGIS WKT parser needs to accept the EMPTY notation (#20753)

and

2. QGIS WKT exporter needs correction for EMPTY lines/collections/...

#12 - 2018-12-20 12:25 AM - Stephen Beattie

Agreed.

Although it seems that #20753 was originally created for the exporter rather than the parser.

#13 - 2019-03-09 03:08 PM - Giovanni Manghi
- Status changed from Feedback to Closed
- Resolution set to end of life

End of life notice: QGIS 2.18 LTR

Source:
http://blog.qgis.org/2019/03/09/end-of-life-notice-qgis-2-18-ltr/

QGIS 3.4 has recently become our new Long Term Release (LTR) version. This is a major step in our history – a long term release version based on the massive updates, library upgrades and improvements that we carried out in the course of the 2.x to 3x upgrade cycle.

We strongly encourage all users who are currently using QGIS 2.18 LTR as their preferred QGIS release to migrate to QGIS 3.4. This new LTR version will receive regular bugfixes for at least one year. It also includes hundreds of new functions, usability improvements, bugfixes, and other goodies. See the relevant changelogs for a good sampling of all the new features that have gone into version 3.4

Most plugins have been either migrated or incorporated into the core QGIS code base.

We strongly discourage the continued use of QGIS 2.18 LTR as it is now officially unsupported, which means we'll not provide any bug fix releases for it.

You should also note that we intend to close all bug tickets referring to the now obsolete LTR version. Original reporters will receive a notification of the ticket closure and are encouraged to check whether the issue persists in the new LTR, **in which case they should reopen the ticket**.

If you would like to better understand the QGIS release roadmap, check out our roadmap page! It outlines the schedule for upcoming releases and will help you plan your deployment of QGIS into an operational environment.

The development of QGIS 3.4 LTR has been made possible by the work of hundreds of volunteers, by the investments of companies, professionals, and administrations, and by continuous donations and financial support from many of you. We sincerely thank you all and encourage you to collaborate and support the project even more, for the long term improvement and sustainability of the QGIS project.

#14 - 2019-03-11 11:22 PM - Stephen Beattie

- Status changed from Closed to Reopened

Re-test in QGIS 3.4.5. Results as follows:

> > > 1. *first test with non-empty multilinestring*
> > >   *y = QgsGeometry.fromWkt('MultiLineString((1 1, 2 2), (3 3, 4 4))')*
> > >   *y*

<QgsGeometry: MultiLineString ((1 1, 2 2),(3 3, 4 4))>

> > > *y.type()*

1

> > > *y.asWkt()*

'MultiLineString ((1 1, 2 2),(3 3, 4 4))'

> > > *print(y)*

<QgsGeometry: MultiLineString ((1 1, 2 2),(3 3, 4 4))>

> > > *g = QgsGeometry.fromWkt('MultiLineString EMPTY')*
> > > *print(g) # expect 'MultiLineString EMPTY', or 'MultiLineString ()' to be consistent with 'h' below*

<QgsGeometry: null>

> > > *g.asWkt() # expect 'MultiLineString EMPTY'*

''

> > > *g.isEmpty() # expect True*

True

> > > *g.type() # expect 1 to match example 'y' above*

3

> > > *h = QgsGeometry.fromWkt('MultiLineString')*
> > > *print(h)*

<QgsGeometry: MultiLineString ()>

> > > *h.asWkt() # expect 'MultiLineString EMPTY'*

'MultiLineString ()'

> > > *h.isEmpty() # expect True*

True

> > > *h.type() # expect 1*

1

> > > *f = QgsGeometry.fromWkt('MultiLineString ()')*
> > > *print(f)*

<QgsGeometry: MultiLineString ()>

> > > *f.asWkt() # expect 'MultiLineString EMPTY'*

'MultiLineString ()'

> > >

> > > *f*

<QgsGeometry: MultiLineString ()>

> > > *f.isEmpty() # expect True*

True

> > > *f.type() # expect 1*

1

#15 - 2019-03-12 11:25 AM - Giovanni Manghi

- Affected QGIS version changed from 2.14.0 to 3.4.5

- Category changed from Python plugins to Python bindings / sipify

- Status changed from Reopened to Open

- Resolution deleted (end of life)