

QGIS Application - Bug report #14578

Geojson coordinates round to int value

2016-03-28 10:26 AM - Alain Delplanque

Status:	Closed	
Priority:	Normal	
Assignee:		
Category:	Unknown	
Affected QGIS version:	2.14.0	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: up/downstream
Crashes QGIS or corrupts data:	No	Copied to github as #: 22551

Description

When opening a geojson file, coordinates are truncated to int values.
Seems to be a locale bug.

I am wokng with ubuntu, by default :

```
$ echo $LANG
fr_FR.UTF-8
```

geojson exemple :

```
{
  "features": [
    {
      "geometry": {
        "coordinates": [
          [
            [ 0.3, 0.0 ],
            [ 1.0, 0.3 ],
            [ 0.7, 1.0 ],
            [ 0.0, 0.7 ],
            [ 0.3, 0.0 ]
          ]
        ],
        "type": "Polygon"
      },
      "properties": {
        "name": "Test"
      },
      "type": "Feature"
    }
  ],
  "type": "FeatureCollection"
}
```

opening with

```
qgis single_poly.geojson
```

```
==> BAD
LANG=fr_FR qgis single_poly.geojson
==> GOOD
LANG=fr_FR.UTF-8 qgis single_poly.geojson
==> BAD
```

I do not think gdal directly involved, because

```
ogr2ogr single_poly.shp single_poly.geojson
```

result in a correct shapefile

This affect QGIS 2.14, but seems to affect older version to.

History

#1 - 2016-03-29 10:59 AM - Maximilian Krumbach

I could not replicate the issue here (ubuntu, QGIS 2.14.1, manually set LANG=fr_FR.UTF-8). Seems to be an other setting that is off.

#2 - 2016-03-29 02:45 PM - Alain Delplanque

I have also :

```
LANGUAGE=fr_FR
```

As I can understand :

QGIS call gdal, and gdal call json-c.

In json-c, json is parse throuth :

```
struct json_object* json_tokenizer_parse_ex(struct json_tokenizer *tok,
                                             const char *str, int len)
```

```
{
    // ... Short init code
```

```
#ifdef HAVE_SETLOCALE
    char *oldlocale=NULL, *tmplocale;
```

```
    tmplocale = setlocale(LC_NUMERIC, NULL);
    if (tmplocale) oldlocale = strdup(tmplocale);
    setlocale(LC_NUMERIC, "C");
#endif
```

```
    // ... Large main parser code is here
```

```
#ifdef HAVE_SETLOCALE
```

```

setlocale(LC_NUMERIC, oldlocale);
if (oldlocale) free(oldlocale);
#endif

// ... Short error code
}

```

json-c use sscanf to convert string to int which is local sensitive.

```

int json_parse_double(const char *buf, double *retval)
{
    return (sscanf(buf, "%lf", retval)==1 ? 0 : 1);
}

```

Can there be interference with QGIS QgsLocaleNumC ?

(Used in QgsOgrProvider::addFeature and QgsOgrProvider::changeAttributeValues)

I don't think so (need QGIS to change locale in another thread, and ALL coordinates are truncated to int)

#3 - 2016-04-03 06:49 AM - Alain Delplanque

I think I found the problem. QGIS does not seem involved, but gdal and ubuntu.

First, even if ubuntu build of gdal require json-c, it seems to use gdal internal source.

In build log: https://launchpadlibrarian.net/220452462/buildlog_ubuntu-wily-amd64.gdal_1.11.2+dfsg-3ubuntu3_BUILDING.txt.gz (when configure of gdal)
:

```

checking for json_object_set_serializer in -ljson-c... no
using internal libjson-c code

```

According to ldd :

```

$ ldd /usr/lib/libgdal.so | grep json
libjson-c.so.2 => /lib/x86_64-linux-gnu/libjson-c.so.2 (0x00007f899fc82000)

```

But according to readelf, libjson isn't used :

```

$ readelf -d /usr/lib/libgdal.so | grep NEEDED
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libarmadillo.so.5]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libproj.so.9]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libpoppler.so.52]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libfreexl.so.1]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libgeos_c.so.1]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libwebp.so.5]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libepsilon.so.1]
0x0000000000000001 (NEEDED)      Bibliothèque partagée: [libodbc.so.2]

```

0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libodbcinst.so.2]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libkmlbase.so.1]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libkmlcore.so.1]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libkmlengine.so.1]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libexpat.so.1]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libxerces-c-3.1.so]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libopenjp2.so.7]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libjasper.so.1]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libnetcdf.so.7]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libhdf5_serial.so.10]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libmfhdf.so.0]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libdf.so.0]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libogdi.so.3.2]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libgif.so.4]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libjpeg.so.8]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libpng12.so.0]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libpq.so.5]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [liblzma.so.5]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libdapclient.so.6]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libdap.so.17]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libspatialite.so.7]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libpcr.so.3]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libcurl-gnutls.so.4]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libxml2.so.2]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libmysqlclient.so.18]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libpthread.so.0]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libz.so.1]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libdl.so.2]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libstdc++.so.6]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libm.so.6]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libc.so.6]
0x0000000000000001 (NEEDED)	Bibliothèque partagée: [libgcc_s.so.1]

Secondly, gdal don't configure locale properly and don't set HAVE_SETLOCALE. Without this, in json-c internal code, the change of locale before call to sscanf is skip.

I open 2 tickets:

- gdal: <https://trac.osgeo.org/gdal/ticket/6438>
- ubuntu: <https://bugs.launchpad.net/ubuntu/+source/gdal/+bug/1565475>

#4 - 2016-04-09 08:59 AM - Giovanni Manghi

- Resolution set to up/downstream
- Status changed from Open to Closed

#5 - 2017-09-22 10:05 AM - Jürgen Fischer

- Category set to Unknown

Files

qgis-geojson.png	73 KB	2016-03-28	Alain Delplanque
------------------	-------	------------	------------------