

QGIS Application - Bug report #1875

A few problems with crs definitions with TOWGS parameters

2009-08-14 06:58 AM - Giovanni Manghi

Status:	Closed	
Priority:	Low	
Assignee:	Magnus Homann	
Category:	Projection Support	
Affected QGIS version:		Regression?: No
Operating System:	All	Easy fix?: No
Pull Request or Patch supplied:		Resolution: fixed
Crashes QGIS or corrupts data:		Copied to github as #: 11935
Description		
Hi, I'm making some extensive test to try break down the problems described in the ticket #1079 and I may have found new issues (or not), one of them is new as changes made r11373. My start point is a layer with a crs definition that includes towgs parameters. You can find an example here https://trac.osgeo.org/qgis/attachment/ticket/1079/Gauss_Boaga.tar.gz a) after r11373, when you open such a layer the towgs parameters are stripped from the original crs string. This doesn't happened before r11373. In the specific case the layer is now recognized as having a "plain" 3003 crs, but it is not true, it misses the original towgs parameters b) if I try to save a layer as a new shapfile, qgis ask em to choose the crs. If a crs with towgs parameters is selected, in the final result the parameters do not appear. You can try doing the following: 1) define a custom crs with the following string <pre>"proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68 +units=m +no_defs"</pre> and now try to save the layer in the above location as shapefile and try apply to it the crs just created. The final result will have a crs like this <pre>"proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +units=m +no_defs "</pre> c) if I have two identical layers, with the only difference that one has towgs parameters in his crs, I can see the shift ONLY if the project is defined in a geographic crs (ex. wgs84). If the project is defined in a projected crs qgis seems to not reproject correctly the layer with towgs parameters. I noticed also that on a copy of qgis 1.2 compiled from source against the dev verison of gdal (1.7), when I open one of the layers of the qgis sample dataset, qgis do not make to recognize as 2964. If then I create a custom crs with the same string, it associates correctly with the custom crs. I noticed also that when creating custom crs's and a trailing space is left at the end of the string, qgis fails to associate the layer crs to it.		

Associated revisions

Revision 4ab570e0 - 2009-08-14 07:59 PM - Magnus Homann

Overwrite the .prj files that GDAL creates for us when saving as SHAPE-file with the WKT from the CRS directly. Might break ESRI-sensitive programs.
Partly fixes #1875.

git-svn-id: <http://svn.osgeo.org/qgis/trunk/qgis@11381> c8812cc2-4d05-0410-92ff-de0c093fc19c

Revision 59f3c8b3 - 2009-08-14 07:59 PM - Magnus Homann

Overwrite the .prj files that GDAL creates for us when saving as SHAPE-file with the WKT from the CRS directly. Might break ESRI-sensitive programs.
Partly fixes #1875.

git-svn-id: <http://svn.osgeo.org/qgis/trunk/qgis@11381> c8812cc2-4d05-0410-92ff-de0c093fc19c

History

#1 - 2009-08-14 07:16 AM - Magnus Homann

b) I discovered just now, GDAL strips the towgs parameter when writing, see also <http://home.gdal.org/projects/opengis/wktproblems.html>

I will investigate a possible workaround.

I think a) and c) has the same root cause.

#2 - 2009-08-14 07:52 AM - Magnus Homann

- Status changed from Open to In Progress

Have added a patch for b).

#3 - 2009-08-14 09:05 AM - Magnus Homann

It appears that GDAL treats a projection with and without towgs as the same.

#4 - 2009-08-14 11:00 AM - Magnus Homann

- Status changed from In Progress to Closed

- Resolution set to fixed

OK, I committed two changes, commit:97ac21e6 (SVN r11381) and that fixes this for me. Re-open if necessary. :-)

#5 - 2009-08-14 11:47 AM - Giovanni Manghi

Hi homann,

trying to test your fixes but right now I'm not able to compile

```
[ 0%] Built target svnversion
```

```
[ 6%] Built target ui
```

```
[ 6%] Building CXX object src/core/CMakeFiles/qgis_core.dir/qgscoordinatereferencesystem.o
```

```
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp: In member function 'bool
[[QgsCoordinateReferenceSystem]]::operator==(const [[QgsCoordinateReferenceSystem]]&)':
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:893: error: invalid conversion from 'char' to 'void*'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:893: error: initializing argument 1 of 'void OGRFree(void*)'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:894: error: invalid conversion from 'char' to 'void*'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:894: error: initializing argument 1 of 'void OGRFree(void*)'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:897: error: invalid conversion from 'char' to 'void*'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:897: error: initializing argument 1 of 'void OGRFree(void*)'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:899: error: invalid conversion from 'char' to 'void*'
/home/gio/Desktop/qgis_unstable/src/core/qgscoordinatereferencesystem.cpp:899: error: initializing argument 1 of 'void OGRFree(void*)'
make: *** [src/core/CMakeFiles/qgis_core.dir/qgscoordinatereferencesystem.o] Error 1
make: *** [src/core/CMakeFiles/qgis_core.dir/all] Error 2
make: *** [all] Error 2
```

#6 - 2009-08-14 11:55 AM - Giovanni Manghi

forget it, I cleaned up and now is compiling. I'll leave feedback soon! thanks!

#7 - 2009-08-14 11:55 AM - Giovanni Manghi

no, spoke to soon. Same error.

#8 - 2009-08-14 12:48 PM - Magnus Homann

jef cleaned it up in commit:7085cf57 (SVN r11384)

#9 - 2009-08-14 01:35 PM - Giovanni Manghi

Hi homann,

thanks! it works everything minus C)

The great news, seems to me, that a number of problems (all but one) described here AND in #1079 have been solved.

This probably lead us back to the original title of #1079, "on the fly projections do not work": as I pointed in C), if two layers differ for towgs parameters in their crs, the shift is visible only if using a geographic projection in the project definitions.

Please let me know if reopen this ticket, open a new one or leave things as now, as seems we are back to #1079 original description.

#10 - 2009-08-14 01:59 PM - Giovanni Manghi

I'm sorry,

but all the confusion of the last days made me look at the problem described in C) in the wrong prospective. It is obvious that if I have two identical layers, that differs just for towgs parameters in the crs definition, and then I enable OTFR and they actually it is right.

Actually I making more tests to find the problem (if one exist yet).

#11 - 2009-08-14 02:00 PM - Giovanni Manghi

... and then I enable OTFR and they actually **align**, it is right.

#12 - 2009-08-14 02:23 PM - Giovanni Manghi

Seems to me that if they align or not depends on the ellipsoid defined in the project crs. So after all it may be a perfect normal thing and it may have no problem at all. But I'm not very deep with projections, so I'll leave others to say the last word.

#13 - 2009-08-14 11:32 PM - Magnus Homann

- Status changed from Closed to Feedback

- Resolution deleted (fixed)

As I don't have GRASS, I'd appreciate if you create two small test layers in shape file format with the two almost identical projections you are using?

#14 - 2009-08-15 12:39 AM - Giovanni Manghi

Hi!

I attached here the example. But look, as I said (maybe not in a very clear way) after all could have no problem at all. Two identical layers that just differ for towgs parameters do align (regardless if the project projection is geographical or projected) depending of the ellipsoid defined in the project crs. This could make sense, but again I'm not very deep yet with projection matters.

If this results ok, I guess that also #1079 can be closed.

#15 - 2009-08-16 10:17 AM - Magnus Homann

It appears to be an intended change in PROJ.4, see this comment from pj_transform.c:

```
---
/* ----- */
/* We cannot do any meaningful datum transformation if either */
/* the source or destination are of an unknown datum type */
/* (ie. only a +ellps declaration, no +datum). This is new */
/* behavior for PROJ 4.6.0. */
/* ----- */
---
```

As there is no datum specified in the +proj string, +towgs84 has no effect. With an old libproj 4.4.9, it seems to work.

#16 - 2009-08-16 10:42 AM - Giovanni Manghi

Replying to [comment:16 homann]:

| As there is no datum specified in the +proj string, +towgs84 has no effect. With an old libproj 4.4.9, it seems to work.

Ummm... so I guess it would be a matter if documenting it and consider the problem "solved"?

After all it appears to be a situation than would have a few easy workarounds.

#17 - 2009-08-16 10:49 AM - Magnus Homann

Well, I can't see what we can do about it. We could guess the datum always is wgs84 when not specified, but I'm not sure that is a good idea at all.

#18 - 2009-08-16 10:54 AM - Giovanni Manghi

Seems to me a particular case. I vote for documenting the new Proj behaviour, using for instance this specific example (with workarounds), and close both this and #1079 tickets.

#19 - 2009-08-16 10:55 AM - Markus Neteler

Replying to [comment:18 homann]:

| Well, I can't see what we can do about it. We could guess the datum always is wgs84 when not specified, but I'm not sure that is a good idea at all.

It is always a bad idea to guess a datum. At least, the user should be told that it is missing.

#20 - 2009-08-16 11:06 AM - Magnus Homann

Not many of the CRS in srs.db have datum specified.

#21 - 2009-08-16 11:37 AM - Magnus Homann

- Status changed from Feedback to Closed
- Resolution set to fixed

Since version 4.6.0 of the PROJ.4 library we use for transformation, the +towgs84 is not handled unless both CRS:es have a +datum specified.

Using proj 4.4.9, there is a notable shift in the layers when using projection.

With 4.6.0, a workaround is to manually define a custom CRS that have a datum specified, e.g. adding a "+datum=WGS84" string. In this case, the layer fauglia.shp can be set to a custom projection, with +datum=WGS84 and a notable shift occur.

NOTE: I haven't verified that the shifts of the layers are correct, only that they occur.

#22 - 2009-08-16 01:30 PM - Giovanni Manghi

Hi homann, thanks again.

Just a small problem yet. I have defined a custom crs with +datum and +towgs parameters and assigned it to a layer, but when loaded qgis shows no sign of the +towgs parameters in the general tab. Qgis seems to strip +towgs parameters and leave +datum. Trying to change by using "specify crs" doesn't change the situation, the crs is correctly defined but qgis ignores or strip the +towgs part.

#23 - 2009-08-16 02:01 PM - Magnus Homann

Can you be more specific? What is the custom CRS you're trying to assign?

#24 - 2009-08-16 02:11 PM - Giovanni Manghi

Hi,

as you suggested I created two custom crs

A) +proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +units=m +no_defs +datum=WGS84

and

B) +proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68 +units=m +no_defs +datum=WGS84

which are basically the same of the two layers I attached to this ticket but with the "+datum" parameter.

I then saved the **same** layer in two version with the above crs, to see if with OTFR enabled and the project defined in WGS84 (+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs) the two align*.

When loading the layer defined with B) qgis strips or ignores

" +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68"

and shows just

" +proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

As said, trying to choose manually the crs doesn't change the situation.

#25 - 2009-08-16 02:22 PM - Markus Neteler

Replying to [comment:25 lutra]:

...

When loading the layer defined with B) qgis strips or ignores

" +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68"

and shows just

" +proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=WGS84 +datum=WGS84 +units=m +no_defs"

So effectively you shift/rotate/scale the map at this point into the wrong direction. If QGIS ignores +towgs84, please don't say "fixed" here because it isn't. I'll make more tests.

In GRASS, we have a dialog which pops up for the definition of the datum and the user can select the datum. And it is not ignored...

#26 - 2009-08-16 03:00 PM - Magnus Homann

Please, can you explain to me what a prj string with both +datum=WGS84 and +towgs84 means? I thought both are datum definitions? See for instance the output of 'cs2cs -ld', where each valid datum is supplied together with the +towgs84 parameters.

Anyway, proj/GDAL seems not to like both of them and uses the former instead.

Remove the +datum=WGS84 from proj B, and test. Either +datum or +towgs84, but not both.

#27 - 2009-08-16 03:39 PM - Giovanni Manghi

Replying to [comment:28 homann]:

Remove the +datum=WGS84 from proj B, and test.

Hi Homann,

If I remove "+datum" from B) the result is the same described earlier in this ticket (see attached files, one defined with +towgs, the other without but also without +datum): the two layer align correctly with OTFR "on" and with certain crs defined for the project. The difference between aligning or not seems to depend if in the project crs are defined the +towgs/+datum/+ellps parameter(s). For example they align if the project has epsg 3003 but not if the project has epsg 4326.

#28 - 2009-08-16 04:17 PM - Giovanni Manghi

Please wait for the tests made by neteler, I'm pretty sure I'm making not necessary confusion.

#29 - 2009-08-17 08:13 AM - Markus Neteler

Ok, I have tested with QGIS trunk from 15 aug 2009. Attached a screenshot, the result of a vector projection from Lat-Long (!OpenStreetMap) and Gauss-Boaga (provincial data) to a provincial 1m DEM which was made in UTM32 look good.

The prj file of the "viapri" Gauss-Boaga file looks like this:

```
# long lines broken for readability here
cat viapri.prj
PROJCS[["Transverse Mercator","GEOGCS[["international","DATUM[["Monte_Mario","SPHEROID[["International_1924]],
TOWGS84[-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68]],PRIMEM[["Greenwich]],
UNIT[["degree","00174532925199433]]PROJECTION[["Transverse_Mercator]],
PARAMETER[["latitude_of_origin","0]]PARAMETER[["central_meridian]],
PARAMETER[["scale_factor","099996]]PARAMETER[["false_easting]],
PARAMETER[["false_northing","0]]UNIT[["Meter]]
```

I did no longer see the error message in the terminal about towgs84 confusion of QGIS. Thanks for having this long term issue fixed.

#30 - 2009-08-18 04:07 AM - Magnus Homann

Great, it seems to be fixed then finally. Most of the projections supplied with QGIS are unfortunately without a +datum clause, and has to be added manually. The reason is that there are sometimes many and sometimes none to chose from.

#31 - 2009-08-18 04:34 AM - Markus Neteler

Replying to [comment:32 homann]:

Great, it seems to be fixed then finally. Most of the projections supplied with QGIS are unfortunately without a +datum clause, and has to be added manually. The reason is that there are sometimes many and sometimes none to chose from.

There is a stabilized and maintained list of datum associations available:

3 param

http://trac.osgeo.org/grass/browser/grass/branches/releasebranch_6_4/lib/gis/datum.table

7 param

http://trac.osgeo.org/grass/browser/grass/branches/releasebranch_6_4/lib/gis/datumtransform.table

Is there any chance to implement a dialog for this as available in GRASS?

http://grass.osgeo.org/screenshots/images/grass64_location_wiz3.png

#32 - 2009-08-18 05:33 AM - hamish -

Hi,

With the latest svn trunk I still see the failure to apply a datum shift as seen in this old screenshot:

http://trac.osgeo.org/qgis/attachment/ticket/1079/qgis_fly_nodatum.png

The problem is that QGIS fails to convert the DATUM string in the WKT shapefile.prj:

```
PROJCS["New Zealand Map Grid",  
  GEOGCS["international",  
    DATUM["New_Zealand_Geodetic_Datum_1949",  
      SPHEROID[[international]],  
      PRIMEM[[Greenwich]],  
      UNIT[[degree]],  
    PROJECTION[[New_Zealand_Map_Grid]],  
    PARAMETER[[latitude_of_origin]],  
    PARAMETER[[central_meridian]],  
    PARAMETER[[false_easting]],  
    PARAMETER[[false_northing]],  
    UNIT[[meter]]
```

If I specifically pick @NZ Map Grid@ from the CRS list it adds that term and the WGS84/LL and NZMG coastlines (more or less) correctly overlap, but I could not find a way to edit the proj4 terms manually in the layer projection settings. If the "Generated CRS" is to be automatically converted into proj4 +terms, it would be nice if it were left in an editable state for those cases, either in the CRS picker GUI or in the layer properties GUI's General tab. Maybe I am missing some easy trick there.

So that gets rid of the 100m+ offset due to the wrong datum (ellipsoid). But a 2-4m shift remains due to the inability to apply +towgs84 or +nadgrids term. For +datum=nzgd49 there are 3 common datum transform options: 3-term +towgs84, 7-term +towgs84, and a +nadgrids NTV2 distortion grid file. One of these maps was reprojected from the other using the grid file.

The +towgs84 datum transform terms are not saved in the shapefile's .prj WKT by GRASS export [IIRC you can add the +proj4 terms as a meta-data extra string in the WKT- that's a todo for grass's v.out.ogr module; somewhere Frank had an example of how that should look. perhaps on the proj4 FAQ or OGR's WKT parsing? I can't find it now...argh]

OT:

I'm finding it very difficult to test these as the map is taking about 2 minutes to render after each adjustment (svn trunk on a brand new amd 64bit quad-core + 8gig RAM). It's a large shapefile with the entire countries coastline as a set of large single-line polygons, but the .shp filesize is only 9MB so it's not that huge. Old versions of QGIS (0.8) had no problem with it. I believe this is happening even before on the fly reprojection is switched on.

If this were GRASS I'd call it the classic "Florida Coastline" problem.

I am happy to send anyone shp.zip's if it would help :)

ramble,
Hamish

#33 - 2009-08-18 05:45 AM - hamish -

slight update:

- on the fly reprojection seems to be the culprit of the huge wait to render after all
- now I see the custom crs menu item and can add new ones. the first time I tried to edit the generated one my edits were lost after I pressed ok (I guess you can only add?). the second time I did the edits and went to click on add button but clicked on the star and it was blanked (tooltips would help there). The third time I successfully added a second custom CRS by hitting the save icon.
- the EPSG file sets +datum=nzgd49 for NZMG <27200>, but no transform options. But in the loading of the shapefile WKT the +datum never arrives. ? (I'm not sure if the epsg file is consulted for that; there is no AUTHORITY:EPSG stuff in this .prj)

?

cheers,
Hamish

#34 - 2009-08-18 05:58 AM - hamish -

perhaps of interest-

?

- comparing CRSs
<http://postgis.refractory.net/pipermail/postgis-users/2004-November/006245.html>



- TOWGS84 as WKT

<http://home.gdal.org/projects/opengis/wktproblems.html>



I still can't find Frank's post about storing +proj terms in a WKT extra field. Frustrating.



Hamish

#35 - 2009-08-18 07:07 AM - hamish -



I still can't find Frank's post about storing +proj terms in a WKT extra field. Frustrating.



Ok, I found him on IRC. the WKT node for the +proj terms is:

```
EXTENSION[[PROJ4""+proj=latlong +datum=WGS84 +wktext]]
```

but a) that isn't ESRI compatible [.prj/map will be rejected] and b) OGR export rips out any WKT terms which are not ESRI compatible. So no luck there.

There is also a WKT node called TOWGS84[] in the CT spec (see wktprobes.html above) but that is also not ESRI compatible so no luck there either.

I am not sure, but I feel that ESRI at least supports AUTHORITY[[EPSG4326]] style, FWIW.

Another thing Frank mentioned is that [[GeoTiff]] has no mechanism to store TOWGS84 parameters.

My reading of all this is that datum transform params will not transfer well from WKT. My perspective is wondering how to improve GRASS's v.out.ogr and r.out.gdal modules to provide as many hints as possible for QGIS et al. to easily pick up. The *only* idea I know of currently is to distribute a .pj4 file containing +proj terms alongside the .prj WKT file, as +proj is the only format which is expressive enough to preserve CRS fidelity.

Hamish

#36 - 2009-08-18 08:12 AM - Magnus Homann

It appears to be something strange about the supplied WKT? If you add a +datum in the Custom CRS and export the file by "Save as shapefile" and supply your changed Custom CRS, a new WKT gets written in the .prj file.

The conversion from WKT to +proj is done by OGR OSRImportFromWkt(). Also the conversion the other way is done with that library.

And the new .prj file is picked up with +datum, and matched against EPSG 27200. Is that correct?

#37 - 2009-08-19 04:36 AM - hamish -

see also bug #1487 (and previously mentioned #1079)

#38 - 2009-08-19 07:05 AM - hamish -

Replying to [comment:38 homann]:

It appears to be something strange about the supplied WKT? If you add a +datum in the Custom CRS and export the file by "Save as shapefile" and supply your changed Custom CRS, a new WKT gets written in the .prj file.

ok, trying that... I notice a message when I do Save as shapefile:

Select the coordinate reference system for the saved shapefile. The data points will be transformed from the layer coordinate reference system.

the output .shp file is binary equal, so I expect that that message might be improved to mention that is the set CRS matches the output CRS no reprojection will happen. Or is that dialog just for creating the .prj WKT and no reprojection of the coordinates will happen if you select wgs84/LL and the wording misleading?

The conversion from WKT to +proj is done by OGR OSRImportFromWkt(). Also the conversion the other way is done with that library.

And the new .prj file is picked up with +datum, and matched against EPSG 27200. Is that correct?

before (GRASS v.out.ogr export):

```
PROJCS["New Zealand Map Grid",  
  GEOGCS["international",  
    DATUM["New_Zealand_Geodetic_Datum_1949",  
      SPHEROID[[international]],  
      PRIMEM[[Greenwich]],  
      UNIT[[degree]],  
    PROJECTION[[New_Zealand_Map_Grid]],  
    PARAMETER[[latitude_of_origin]],  
    PARAMETER[[central_meridian]],  
    PARAMETER[[false_easting]],  
    PARAMETER[[false_northing]],  
    UNIT[[meter]]
```

new .prj after Save as shapefile in QGIS:

```
PROJCS["unnamed",
  GEOGCS["NZGD49",
    DATUM["New_Zealand_Geodetic_Datum_1949",
      SPHEROID["International 1924",6378388,297,
        AUTHORITY[[EPSG""7022]],
      TOWGS84[59.47,-5.04,187.44,0.47,-0.1,1.024,-4.5993],
      AUTHORITY[[EPSG""6272]],
    PRIMEM[[Greenwich]0AUTHORITY["EPSG""8901]],
    UNIT[[degree]001745329251994328AUTHORITY["EPSG""9122]],
    AUTHORITY[[EPSG""4272]],
  PROJECTION[[New_Zealand_Map_Grid]],
  PARAMETER[[latitude_of_origin]],
  PARAMETER[[central_meridian]],
  PARAMETER[[false_easting]],
  PARAMETER[[false_northing]],
  UNIT[[Meter]]
```

if I reload the new shapefile into QGIS it automatically selects the Projected Cood systems -> NZ Map Grid -> NZGD49/NZMG from the CRS list with 27200 in the EPSG column and 2269 in the ID column, and of course +datum=nzgd49.

So the problem is that QGIS's WKT importer is lossy when given WKT created by GRASS's v.out.ogr module. That's kind of weird when both use OGR for the task. (actually I'm not sure if grass's v.out.ogr uses OGR fns() for the WKT creation or if it uses something from elsewhere in the GIS)

As "DATUM["New_Zealand_Geodetic_Datum_1949"," etc makes it into the GRASS-created WKT, I'd consider that sufficient and the info loss happening during the import to QGIS. Nothing of interest is printed to the terminal (how to switch on debug messages?).

Hamish

#39 - 2009-08-19 07:21 AM - Magnus Homann

So, it appears that the WKT your v.out.ogr module wrote isn't translatable by OGR itself, or rather it does not represent EPSG 27200.

"epsg_tr.py -pretty_wkt 27200" results below:

```
PROJCS["NZGD49 / New Zealand Map Grid",
  GEOGCS["NZGD49",
    DATUM["New_Zealand_Geodetic_Datum_1949",
      SPHEROID["International 1924",6378388,297,
        AUTHORITY[[EPSG""7022]],
      TOWGS84[59.47,-5.04,187.44,0.47,-0.1,1.024,-4.5993],
      AUTHORITY[[EPSG""6272]],
    PRIMEM["Greenwich",0,
      AUTHORITY[[EPSG""8901]],
    UNIT["degree",0.01745329251994328,
      AUTHORITY[[EPSG""9122]],
    AUTHORITY[[EPSG""4272]],
  UNIT["metre",1,
    AUTHORITY[[EPSG""9001]],
```

```
PROJECTION[[New_Zealand_Map_Grid]],  
PARAMETER[[latitude_of_origin]],  
PARAMETER[[central_meridian]],  
PARAMETER[[false_easting]],  
PARAMETER[[false_northing]],  
AUTHORITY[[EPSG:"27200]],  
AXIS[[Easting]],  
AXIS[[Northing]]  
---
```

So why do you believe that QGIS is wrong? The supplied WKT is not the same as what OGR thinks EPSG 27200 should be.

#40 - 2009-08-19 08:13 AM - hamish -

Replying to [comment:41 homann]:

*So, it appears that the WKT your v.out.ogr module wrote
isn't translatable by OGR itself, or rather it does not
represent EPSG 27200.*

The export module is not trying to represent EPSG code X (actually by that point it has already forgotten what the seed EPSG code was). It's trying to export a series of non-overlapping coefficients and labels which describe a CRS as best it can. Frank's scripts which create the /usr/share/proj/epsg file only approximate what is really in the EPSG database; some codes are not representable by +proj4 syntax at all (eg multiple datums defined for a single code). In the past there have been library function changes which inadvertently changed the proj4 epsg file's floating point low signif. digit end rounding, causing e.g. msieczyka's exact-match problems in and about the time of #1079.

*So why do you believe that QGIS is wrong? The supplied WKT
is not the same as what OGR thinks EPSG 27200 should be.*

QGIS is passed WKT which says "DATUM["New_Zealand_Geodetic_Datum_1949"," and it fails to convert that to the proj term +datum=nzgd49. That name is the same as "proj -ld" gives for the datum name, it's the Well-Known non-ESRI variant. The CRS import code should be flexible enough to figure that out and not depend on strcmp() of all components.

(my hunch is that "international" vs "International 1924" ellipsoid definition is the culprit in this case)

Is it a fault in the QGIS or OGR codebase? I've no idea. Probably a mix of both or a misassumption of one by the other.

It doesn't matter if the program recognizes that the terms of EPSG 27200 are highly similar to the WKT given, just as long as the +proj4 terms get populated correctly.

<http://spatialreference.org/ref/epsg/27200/>

And FWIW, it's not EPSG-KT, it's WKT. And there are ~5 different flavors of WKT depending on the vendor, so insisting that it matches the current OGR-generated version precisely to work properly ain't going to help when someone shows up with some data from a popular non-OSGeo family GIS. WKT is a very sloppy human-readable definition and always will be. It is much harder to program, but a robust fuzzy-logic system needs to be flexible and smart when handed that slop.

thanks and best regards,
Hamish

#41 - 2009-08-19 08:31 AM - Magnus Homann

Again, the WKT you have supplied is not translated by OGR into the proj-string you think it should be. OGR library is used for that. The proj-string is then used by PROJ4 to do the actual projection. OGR library is used for that translation also.

If you use the GDAL/OGR program "testpsg <your-WKT-definition>" you will see that OGR does not pick up the datum. The error (if any) is reproducible outside QGIS. If you are sure the WKT is correct, please file a bug with OGR, where the problem (if any) can be solved.

Posting more comments here will not fix the problem.

#42 - 2009-08-19 08:57 AM - Frank Warmerdam -

Reviewing the ogr_srs_proj4.cpp code the +datum=nzgd49 is only generated when exporting to PROJ.4 if a 6272 datum authority code is present in the WKT. There is no attempt to recognize this datum by datum name.

If you wish to take exception to this you might want to file a ticket in the GDAL/OGR trac. I'm willing to contemplate improved logic.

#43 - 2009-08-19 10:02 AM - hamish -

Replying to [comment:43 homann]:

Again, the WKT you have supplied is not translated by OGR into the proj-string you think it should be. OGR library is used for that. The proj-string is then used by PROJ4 to do the actual projection. OGR library is used for that translation also.

Ok, got it.

If you use the GDAL/OGR program "testpsg <your-WKT-definition>" you will see that OGR does not pick up the datum.

Well, at least it got the SPHEREOID[] numbers correct...

for the archives, my testpsg setup,

```
GRASS65.svn> gdal/svn/trunk/apps/testpsg "@g.proj -wf@" | less
```

The error (if any) is reproducible outside QGIS. If you are sure the WKT is correct, please file a bug with OGR, where the problem (if any) can be solved.

Posting more comments here will not fix the problem.

It is not my intention to spam the ticket or waste anyone's time, only to smash away at the problem until I/we/them understand it and each other well enough to solve the thing. :)

Replying to [comment:44 warmerdam]:

Reviewing the ogr_srs_proj4.cpp code the +datum=nzgd49 is only generated when exporting to PROJ.4 if a 6272 datum authority code is present in the WKT. There is no attempt to recognize this datum by datum name.

ah,

If you wish to take exception to this you might want to file a ticket in the GDAL/OGR trac. I'm willing to contemplate improved logic.

patch filed as gdal/ogr ticket # 3104 for your consideration.

cheers,
Hamish

Files			
patch_1875_B_1.diff	1.71 KB	2009-08-14	Magnus Homann
fauglia.tar.gz	38 KB	2009-08-14	Giovanni Manghi
qgis_trunk_vector_reprojection.jpg	151 KB	2009-08-17	Markus Neteler
saveasshape2.prj	545 Bytes	2009-08-18	Magnus Homann