

## QGIS Application - Bug report #1079 on the fly projections do not work

2008-05-09 06:01 AM - Paolo Cavallini

<b>Status:</b>	Closed	
<b>Priority:</b>	Low	
<b>Assignee:</b>	Magnus Homann	
<b>Category:</b>	Projection Support	
<b>Affected QGIS version:</b>		<b>Regression?:</b> No
<b>Operating System:</b>	All	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>		<b>Resolution:</b> fixed
<b>Crashes QGIS or corrupts data:</b>		<b>Copied to github as #:</b> 11139
<b>Description</b>		
<p>Layers with different projections in some cases are not aligned properly when enabling on the fly projections. This may be due to GDAL, as pointed out by Maciek:</p> <p>QGIS fails to recognize many SRSSes at layer load time, due to a change in GDAL r1.</p> <p>r1<a href="http://www.nabble.com/forum/ViewPost.jtp?post=16592790&amp;framed=y">http://www.nabble.com/forum/ViewPost.jtp?post=16592790&amp;framed=y</a></p>		

### History

#### #1 - 2008-05-09 06:54 AM - leolami -

I tried it in QGIS 0.10 for Windows build by Marco Pasetti and it works fine.

What about gdal version used here?

Leonardo

#### #2 - 2008-05-13 05:36 AM - Maciej Sieczka -

AFAIK, QGIS fails to recognize only SRSSs which contain an **explicit k parameter** in the SRS proj4 string (which are several hundred SRSSs anyway). It does recognize other properly though. The nature of the problem depends on the **particular** coordinate system **and GDAL version**, and has it's background in the fact that QGIS uses proj4 faromat-based, string-wise SRS matching.

Example:

EPSG:2180 has a following proj4 representation in QGIS srs.db:

```
+proj=tmerc +lat_0=0 +lon_0=19 +k=0.999300 +x_0=500000 +y_0=-5300000 +ellps=GRS80 +units=m +no_defs
```

Against GDAL 1.4.4 QGIS fails to recognize this SRS because GDAL 1.4.4 interprets EPSG:2180 as:

```
+proj=tmerc +lat_0=0 +lon_0=19 +k=0.9993 +x_0=500000 +y_0=-5300000 +ellps=GRS80 +units=m +no_defs
```

Note the difference in k above - 0.999300 against 0.9993 is the cause of string-wise matching failure. The reason of this problem is a change in GDAL <http://trac.osgeo.org/gdal/changeset/12625>.

Against GDAL > 1.4.4 QGIS fails to recognize this SRS because these GDAL versions interpret EPSG:2180 as:

```
+proj=tmerc +lat_0=0 +lon_0=19 +k=0.9993000000000001 +x_0=500000 +y_0=-5300000 +ellps=GRS80 +units=m +no_defs
```

Again, k differ in both cases. This one is due to a bug in GDAL <https://trac.osgeo.org/gdal/ticket/2036>. It is fixed in GDAL 1.6 trunk but the fix caused side-effects, which have to be taken care of before the fix can be backported to stable 1.5 branch.

Other SRS which do not contain an explicit k parameter, eg. EPSG:32633, which in proj4 representation is:

```
+proj=utm +zone=33 +ellps=WGS84 +datum=WGS84 +units=m +no_defs
```

do not pose problems.

Back to the point - if QGIS fails to recognize the SRS of the added layer, by default it assigns it:

```
+proj=longlat +ellps=WGS84 +datum=WGS84
```

Now, when one tries to reproject such layer on the fly, he might come to wrong conclusion that the functionality is broken. However, the real culprit is that the **layer's source SRS is wrong**, so the reprojection cannot work properly. Can you Leonardo and Paolo please verify if this is the case for you - that for Paolo the reprojection seems broken and for Leonardo it's OK?

### #3 - 2008-05-21 09:03 AM - Paolo Cavallini

This may be related to locale: in LOCALE=it\_IT, points are replaced with commas in the description of projections, thus qgis does not understand the projections, and grass plugin crashes.

exporting LANG=C solves the problem. Apparently the same as <https://trac.osgeo.org/qgis/ticket/1080>

### #4 - 2008-05-25 06:09 AM - hamish -

From the above linked nabble archive thread:

*This is Frank Warmerdam's answer :*

*The problem is that if EPSG offers more than one transformation to WGS84 for a datum, the automated translation gives up and offers none of them preferring for the user to make the decision themselves.*

This bug has just been noticed on the grass-users list:

<http://thread.gmane.org/gmane.comp.gis.grass.user/23928/focus=23948>

I notice with my pet projection (NZ Map Grid epsg:27200) that the datum transform is missing. (see screenshot attached to this bug; those are two GRASS vector layers loaded with the GRASS plugin and taken from both WGS84 LL and NZMG locations)

I am working with QGIS 0.8.1 for Debian/etch from backports.org package (latest build available so far). Proj 4.6.0 and GDAL 1.4.4.0 from backports.org too.

I notice that +datum=nzgd49 is missing from the srs.db version of the NZMG projection versus what is in the EPSG file. The above note from Frank explains that(?): there are typically 3 datum transform options used for NZMG- the 3 term, the 7 term, and a NTV2 grid file (grid file distributed with GRASS and PROJ.4).

I tried downloading the latest srs.db from SVN/trunk and replacing the one in /usr/share/qgis/resources/. No change.

I next opened up the new srs.db in sqlitebrowser and edited the entry\* and added +towgs84=59.47,-5.04,187.44,0.47,-0.1,1.024,-4.5993 which is the standard 7-term transform used for +datum=nzgd49. No change. After that I also threw +datum=nzgd49 into the string with no change. The new terms shows up in the dialog if I go to the layer's "P"roperties projection info, but in the terminal I see:

```
Debug: /tmp/build/qgis-0.8.1/src/gui/qgsproject.cpp:1065 4 properties read
Debug: current properties:
Debug: name: properties
Debug: key: <SpatialRefSys> subkey: <SpatialRefSys>
Debug:   name: [[SpatialRefSys]]
Debug:   key: <ProjectSRSProj4String> value: +proj=nzmg +lat_0=-41 +lon_0=173 +x_0=2510000 +y_0=6023150 +ellps=intl +units=m
+no_defs
Debug:   key: <ProjectionsEnabled> value: 1
Debug:   key: <ProjectSRSID> value: 1555
Debug: key: <Gui> subkey: <Gui>
...
```

ie the binary seems to be pulling the SRS data from somewhere else as well, and that somewhere else is not updated. And what is shown from srs.db is just cosmetic? (at least at runtime)

[\*] sqlitebrowser edit: Browse Data tab; Table: tbl\_srs; Go to: 1555; double click on parameters; Apply

[\*\*] bonus: Firefox sqlite DB management GUI plugin:

<https://addons.mozilla.org/en-US/firefox/addon/5817>

It doesn't solve the general problem, but for NZMG I'd be happy with hardcoding the 7-term transform above until the general problem is solved. It isn't as good as using the grid file, but is a lot better than the current 100m+ shift. I can upload a new srs.db binary to someone with commit rights if it helps. (but is there more places to edit it?)

Plus if this gets fixed I can complain about [[ArcGIS]] 9.3's on the fly reprojection ignoring the datum transform again. I feel guilty about doing that while QGIS gets it wrong too. ;)

Hamish

**#5 - 2008-05-25 06:50 PM - hamish -**

see also bugs #1037 and #1035

**#6 - 2008-05-25 07:32 PM - hamish -**

ok, the srs.db file is badly missing +datum= and +towgs84= parameters from many SRS codes. So anything that should do a datum transform won't. It should be regenerated from latest **SVN** GDAL+PROJ scripts + EPSG db IMO.

see new comments in bugs #1035 and #1037.

Hamish

**#7 - 2008-07-10 11:14 AM - Maciej Sieczka -**

Guys

An updated srs.db has been committed to SVN trunk. Please verify how the issues you mention in this discussion look now. Let me know if the ticket can be closed.

**#8 - 2008-08-02 01:20 AM - Tim Sutton**

- *Status changed from Open to Closed*

- *Resolution set to fixed*

Ticket closed as solution has been implemented by msieczka and rduif and is pending user feedback for almost a month. If the issue still exists please reopen and provide fresh info.

**#9 - 2008-08-21 03:05 AM - leolami -**

Hi all,

the problem seems solved, I tested using a projection like Gauss Boaga standard (EPSG code=3003)

```
+proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +units=m +no_defs
```

I try to reproject with a UTM wgs84 fuse 32N layer and it works well.

A little correlate problem is that if I load a layer from a Location of GRASS with this parameters of projection:

```
+proj=tmerc +lat_0=0 +lon_0=9 +k=0.999600 +x_0=1500000 +y_0=0  
+ellps=intl +units=m +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68
```

when I look the projection associated to this layer I can see that QGIS doesn't know this projection and that it associates to the layer the projection setted as default.

But if I create a custom projection like this and I set it as the default projection when I load the GRASS layer I see that QGIS associates to the layer the standard projection:

```
+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

but I remember that before QGIS readed the projection with parameters like the same projection without them and it associated to the layer the standard Gauss Boaga projection (EPSG code=3003)

Do you think we have to reopen the ticket?

Thank you

Leonardo

**#10 - 2008-08-21 04:46 AM - Markus Neteler**

Replying to [comment:10 leolami]:

*Hi all,  
the problem seems solved, I tested using a projection like Gauss Boaga standard (EPSG code=3003)*

```
+proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +units=m +no_defs
```

Note that this string is incomplete. It lacks +towgs84=...

See also #377 and especially #418. The problem persists and needs to be solved.

Markus

#### #11 - 2008-08-21 04:52 AM - Maciej Sieczka -

Replying to [comment:10 leolami]:

*But if I create a custom projection like this and I set it as the default projection when I load the GRASS layer I see that QGIS associates to the layer the standard projection:*

```
+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

*but I remember that before QGIS readed the projection with parameters like the same projection withot them and it associated to the layer the standard Gauss Boaga projection (EPSG code=3003)*

Please post a sample GRASS location with a single layer for testing.

Maciek

#### #12 - 2008-08-21 04:57 AM - Marco Pasetti -

Replying to [comment:11 neteler]:

*Replying to [comment:10 leolami]:*

*Hi all,  
the problem seems solved, I tested using a projection like Gauss Boaga standard (EPSG code=3003)*

```
+proj=tmerc +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +ellps=intl +units=m +no_defs
```

*Note that this string is incomplete. It lacks +towgs84=...*

*See also #377 and especially #418. The problem persists and needs to be solved.*

*Markus*

yes. when I work with GB projections in QGIS I define a custom projection.

For example, for Gauss Boaga Roma40 West, peninsular part, I define:

```
+proj=tmerc +ellps=intl +lat_0=0 +lon_0=9 +k=0.9996 +x_0=1500000 +y_0=0 +no_defs +a=6378388 +rf=297
```

+towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68 +to\_meter=1

the epsg code 3003 is too generic for that purpose

Marco

**#13 - 2008-08-21 05:36 AM - leolami -**

Hi all,

I posted a sample GRASS location with towgs84 parameters with a single layer for testing

**#14 - 2008-08-21 06:31 AM - Maciej Sieczka -**

Replying to [comment:10 leolami]:

*A little correlate problem is that if I load a layer from a Location of GRASS with this parameters of projection:*

```
+proj=tmerc +lat_0=0 +lon_0=9 +k=0.999600 +x_0=1500000 +y_0=0  
+ellps=intl +units=m +towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68
```

*when I look the projection associated to this layer I can see that QGIS doesn't know this projection and that it associates to the layer the projection setted as default.*

*But if I create a custom projection like this and I set it as the default projection when I load the GRASS layer I see that QGIS associates to the layer the standard projection:*

```
+proj=longlat +ellps=WGS84 +datum=WGS84 +no_defs
```

*but I remember that before QGIS readed the projection with parameters like the same projection without them and it associated to the layer the standard Gauss Boaga projection (EPSG code=3003)*

The problem is that QGIS produces an exact, particular proj4 string for your layer, and tries to find a match for it in it's CRS database. In order to let it find it, you need to define a custom CRS using a prj4 exactly the same QGIS could be looking for.

You can find the string QGIS is looking for in it's terminal output when it's loading the layer. In your case the very string is:

```
+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
```

The string **you** quoted is slightly different:

```
+proj=tmerc +lat_0=0 +lon_0=9 +k=0.999600 +x_0=1500000 +y_0=0 +ellps=intl +units=m  
+towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68
```

i.e.: different order of parameters, trailing zeros in 'k', missing 'no\_defs' - enough for QGIS to make it not recognize it.

As long as QGIS depends on string-wise proj4 definitions matching, we need to keep an eye on such subtle differences.

If you specify your custom CRS the way QGIS expects it, matching should work for you (it does for me).

Maciek

#### #15 - 2008-08-23 07:25 AM - Markus Neteler

- Status changed from Closed to Feedback

- Resolution deleted (fixed)

I confirm that QGIS does not understand data in a GRASS location with towgs parameter:

```
Warning: [[QgsCoordinateReferenceSystem]]::getRecord failed : select * from tbl_srs where
parameters='+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'
```

```
GRASS 6.4.svn (pat): > g.proj -w
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]],
  PROJECTION[[Transverse_Mercator]],
  PARAMETER[[latitude_of_origin]],
  PARAMETER[[central_meridian]],
  PARAMETER[[scale_factor]],
  PARAMETER[[false_easting]],
  PARAMETER[[false_northing]],
  UNIT[[Meter]]
```

```
g.proj -wf > /tmp/gb1.prj
testepsg /tmp/gb1.prj
Validate Succeeds.
WKT[/tmp/gb1.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]],
  PROJECTION[[Transverse_Mercator]],
  PARAMETER[[latitude_of_origin]],
  PARAMETER[[central_meridian]],
  PARAMETER[[scale_factor]],
  PARAMETER[[false_easting]],
  PARAMETER[[false_northing]],
  UNIT[[Meter]]
```

```
Simplified WKT[/tmp/gb1.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]],
  PROJECTION[[Transverse_Mercator]],
  PARAMETER[[latitude_of_origin]],
  PARAMETER[[central_meridian]],
  PARAMETER[[scale_factor]],
  PARAMETER[[false_easting]],
  PARAMETER[[false_northing]],
  UNIT[[Meter]]
```

```
Old Style WKT[/tmp/gb1.prj] = PROJCS["Transverse Mercator",GEOGCS["international",DATUM["Monte_Mario",
SPHEROID[[International_1924"6378388297]]PRIMEM["Greenwich]],
UNIT[[degree"00174532925199433]]PROJECTION["Transverse_Mercator]],
PARAMETER[[latitude_of__origin"0]PARAMETER["central_meridian]],
PARAMETER[[scale_factor"09996]PARAMETER["false_easting]],
PARAMETER[[false_northing"0]UNIT["Meter]]
```

```
ESRI'ified WKT[/tmp/gb1.prj] =
PROJCS["Transverse Mercator",
  GEOGCS["international",
    DATUM["D_Monte_Mario",
      SPHEROID[[International_1924]],
      PRIMEM[[Greenwich]],
      UNIT[[Degree]],
    PROJECTION[[Transverse_Mercator]],
    PARAMETER[[latitude_of_origin]],
    PARAMETER[[central_meridian]],
    PARAMETER[[scale_factor]],
    PARAMETER[[false_easting]],
    PARAMETER[[false_northing]],
    UNIT[[Meter]]
```

```
PROJ.4 rendering of [/tmp/gb1.prj] = +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
```

Since GDAL considers this projection **valid**, it must be a QGIS problem. I take liberty to reopen the ticket.

Markus

**#16 - 2008-08-25 01:36 AM - Magnus Homann**

Aha, that explains that my custom projection with a towgs stopped working go the wrong resulta. Presumably the towgs options was ignored? I do not think this has anything to do with GDAL.



**#17 - 2008-08-29 10:35 PM - Markus Neteler**

Here another one (S-JTSK\_Krovak):

Warning: [[QgsCoordinateReferenceSystem]]::getRecord failed : select \* from tbl\_srs where parameters='+proj=krovak +lat\_0=49.5 +lon\_0=24.833333333333333 +alpha=30.28813975277778 +k=0.9999 +x\_0=0 +y\_0=0 +ellps=bessel +units=m +no\_defs'

Not sure why it fails to be detected.

**#18 - 2008-08-30 01:41 AM - Maciej Sieczka -**

Replying to [comment:16 neteler]:

*I confirm that QGIS does not understand data in a GRASS location with towgs parameter:*

```
> Warning: [[QgsCoordinateReferenceSystem]]::getRecord failed : select * from tbl_srs where
> parameters='+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'
> <snip>
```

QGIS doesn't recognize this CRS, because epsg\_tr.py (a GDAL utility, used in the QGIS CRS database creation script) will not create an exactly the same string for any EPSG code. I checked that with all EPSG codes against my GDAL 1.5.2 instalation. Markus, what EPSG code is this proj4 string supposed to represent?

*Since GDAL considers this projection **valid**, it must be a QGIS problem. I take liberty to reopen the ticket.*

True, the CRS definition is valid, in general, according to testepsg (another GDAL utility), yet it is not an exact representation of any EPSG code according to GDAL, as checked with epsg\_tr.py.

Looks like the bottom line of the problem is that QGIS uses an overly strict CRS matching. QGIS should support any CRS definition that GDAL considers valid, not only those which are exact equivalents of some EPSG codes.

QGIS devs - can it? The testepsg from GDAL seems to provide the required functionality.

**#19 - 2008-08-30 01:51 AM - Maciej Sieczka -**

Replying to [comment:18 neteler]:

*Here another one (S-JTSK\_Krovak):*

*Warning: [[QgsCoordinateReferenceSystem]]::getRecord failed : select \* from tbl\_srs where  
parameters='+proj=krovak +lat\_0=49.5 +lon\_0=24.833333333333333 +alpha=30.28813975277778  
+k=0.9999 +x\_0=0 +y\_0=0 +ellps=bessel +units=m +no\_defs'*

*Not sure why it fails to be detected.*

For the same reason as explained in comment:19.

QGIS devs please tell what you think about the idea outlined in the last 2 paragraphs there.

**#20 - 2008-08-31 02:15 AM - Markus Neteler**

Replying to [comment:19 msieczka]:

*Replying to [comment:16 neteler]:*

*I confirm that QGIS does not understand data in a GRASS location with towgs parameter:*

```
> > Warning: [[QgsCoordinateReferenceSystem]]::getRecord failed : select * from tbl_srs where
> > parameters='+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'
> > <snip>
```

*QGIS doesn't recognize this CRS, because epsg\_tr.py (a GDAL utility, used in the QGIS CRS database creation script) will not create an exactly the same string for any EPSG code. I checked that with all EPSG codes against my GDAL 1.5.2 instalation. Markus, what EPSG code is this proj4 string supposed to represent?*

GRASS 6.4.svn (pat):~> g.proj -j | tr '\n'

...

```
+proj=tmerc +lat_0=0.0000000000 +lon_0=9.0 +k_0=0.9996000000 +x_0=1500000.0000000000 +a=6378388 +rf=297 +no_defs
+towgs84=-104.1,-49.1,-9.9,0.971,-2.917,0.714,-11.68 +to_meter=1
```

It is EPSG 3003 + towgs84 (since PROJ4's epsg file still omits towgs84 when multiple datums are available as in many European countries).

*Since GDAL considers this projection **valid**, it must be a QGIS problem. I take liberty to reopen the ticket.*

*True, the CRS definition is valid, in general, according to testepsg (another GDAL utility), yet it is not an exact representation of any EPSG code according to GDAL, as checked with epsg\_tr.py.*

*Looks like the bottom line of the problem is that QGIS uses an overly strict CRS matching. QGIS should support any CRS definition that GDAL considers valid, not only those which are exact equivalents of some EPSG codes.*

Right. Currently it is a showstopper which renders QGIS' on the fly projection unusable as I cannot afford shifts of >100m just due to the omission of towgs84.

*QGIS devs - can it? The testepsg from GDAL seems to provide the required functionality.*

to compile that:

```
cd gdal/apps/
make testepsg
```

**#21 - 2008-10-18 07:40 AM - Magnus Homann**

What is the status of this bug? Do we need to fix the CRS recognition logic yet?

**#22 - 2009-03-28 03:17 PM - Paolo Cavallini**

Still a serious bug: see also #377

**#23 - 2009-07-30 11:11 AM - Paolo Cavallini**

- Status changed from *Feedback* to *Open*

**#24 - 2009-08-13 01:35 AM - Giovanni Manghi**

Replying to [comment:22 homann]:

| What is the status of this bug? Do we need to fix the CRS recognition logic yet?

Hi,

I don't know if someone else noticed, but the problems described in this ticket (together with #418) probably makes that the projections of the vectors in the qgis sample dataset are not recognized by qgis itself. This is very puzzling for new users.

I noticed also the following:

\*) sample data is declared to be in epsg 2964

<http://www.qgis.org/en/download/sample-data.html>

but the string I see in the "general" tab after loading a vector, is different from the 2964 one in the qgis internal db.

\*) When I try to select 2964 from the list of available projections, it doesn't change and remain with the original string. This doesn't happen if I choose any other projection from the available list.

\*) If I create a custom CRS with the string I see in the "general" tab, then it works and is recognized, but I have to take care to strip the blank space at the end of it, otherwise it doesn't work.

**#25 - 2009-08-13 06:21 AM - Magnus Homann**

- Status changed from *Open* to *In Progress*

Pcav promised to attach a test case that I could use. Then I'll have a look.

#418 is probably a feature extensions and text string change, so I don't think that will happen to 1.2.0.

If timlinux doesn't mind, I'll reassign it to myself. :-)

**#26 - 2009-08-13 08:08 AM - Giovanni Manghi**

Replying to [comment:27 homann]:

| Pcav promised to attach a test case that I could use.

I may be wrong, but the title of this ticket seems misleading to me. After all qgis do not fails to reproject, fails to recognize (and use?) the right projections. So if this is the case, no particular case is needed to test this behaviour. Just try the qgis sample dataset as I pointed in my previous comment.

#27 - 2009-08-13 08:45 AM - Magnus Homann

I prefer that we keep only one issue per bug ticket, instead of adding everythinhg that seems wrong with projection to one ticket. This ticket has now become very confusing.

#28 - 2009-08-16 11:38 AM - Magnus Homann

- Resolution set to fixed
- Status changed from In Progress to Closed

Closing this, as it seems we have finally solved the issue. See #1875.

#29 - 2009-08-19 04:38 AM - hamish -

see also comments in bugs #418, #1487, and #1875

Files

qgis_fly_nodatum.png	51.1 KB	2008-05-25	hamish -
Gauss_Boaga.tar.gz	38.3 KB	2008-08-21	leolami -