

QGIS Application - Bug report #8487  
manual CRS being overwritten by QGIS

2013-08-19 01:40 PM - Gavin Fleming

<b>Status:</b>	Closed	
<b>Priority:</b>	Severe/Regression	
<b>Assignee:</b>		
<b>Category:</b>	Projection Support	
<b>Affected QGIS version:</b>	master	<b>Regression?:</b> No
<b>Operating System:</b>	Ubuntu	<b>Easy fix?:</b> No
<b>Pull Request or Patch supplied:</b>	No	<b>Resolution:</b> fixed/implemented
<b>Crashes QGIS or corrupts data:</b>	No	<b>Copied to github as #:</b> 17252
<b>Description</b>		
<p>The new series of EPSG codes for South Africa (Hartebeesthoek/LO) with +axis=wsu might be correct for land survey data in South Africa, which is 'south-facing', but it is NOT correct for the way GIS is practised in South Africa, where conventionally-oriented coordinates are used.</p> <p>QGIS is now forcefully overriding my manual CRS definitions so I cannot work in our local projections any more.</p> <p>I define a CRS as such: +proj=tmerc +lat_0=0 +lon_0=21 +k=1 +x_0=0 +y_0=0 +ellps=WGS84 +towgs84=0,0,0,0,0,0 +units=m +no_defs, which QGIS 1.8 and even 1.9 till recently obeyed. Now QGIS replaces this with +proj=tmerc +lat_0=0 +lon_0=21 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0 +units=m +no_defs (note the added +axis=wsu) which is completely unusable since no GIS data in South Africa has 'south-facing' coordinates. The map is flipped upside down and back to front and I have no way of correcting it.</p> <p>QGIS is apparently detecting that my definition is similar to EPSG:2049 and imposing the EPSG:2049 definition on my own.</p>		

History

#1 - 2013-08-22 12:22 PM - Jürgen Fischer

Is that a custom crs or did you modify srs.db?

#2 - 2013-08-23 02:32 AM - Giovanni Manghi

- Operating System deleted (ubuntu)
- OS version deleted (13.04)

Jürgen Fischer wrote:

Is that a custom crs or did you modify srs.db?

it is a custom crs, I tested and qgis does indeed overwrite the crs string.

#3 - 2013-08-26 02:38 AM - Frank Sokolic

I've been able to solve this bug on my system (Ubuntu 13.04, QGIS master): If I open /usr/share/qgis/reources/srs.db using spatialite and then edit the relevant rows in the tbl\_srs table to remove +axis=wsu then Bug #8487 goes away. Is this a possible solution to this bug?

#### #4 - 2013-08-26 02:47 AM - Jürgen Fischer

Giovanni Manghi wrote:

*it is a custom crs, I tested and qgis does indeed overwrite the crs string.*

What exactly happens? I suppose it's not assigning the user CRS to the layer, when loading the shape file and assigns the "standard" crs instead. But it doesn't modify (overwrite) the existing user CRS and manually assigning the user crs works as expected. Right?

#### #5 - 2013-08-26 02:56 AM - Leyan Ouyang

It is due to the changes I made in the custom CRS handling. Previously, it was all manual, then I changed it to make use of `QgsCoordinateReferenceSystem`. When creating a custom CRS, a `QgsCoordinateReferenceSystem` is created using `createFromProj4`, then is saved. The second string is the result of `toProj4` on this CRS. To clarify, the custom CRS is not saved as expected, the modified version is saved instead. Manually changing would not help, but modifying the database can allow to replace with the correct value.

However, `createFromProj4` is trying too hard to match to an existing CRS: it will match with an existing CRS, even if this CRS has additional parameters. It is even documented [here](#)

I think the clean way would be to make smaller utility class to manage CRS (check validity, convert between representations), separated from `QgsCoordinateReferenceSystem` which tries very hard to fit to existing CRS. However, I was very close to the feature freeze when I submitted this, and couldn't make the changes. I was not very happy with this matching to existing CRS, as it can rearrange some parameters, but I did not think it would make some custom CRS unusable ...

#### #6 - 2013-08-26 03:40 AM - Giovanni Manghi

Jürgen Fischer wrote:

*What exactly happens?*

you create the custom CRS with

```
+proj=tmerc +lat_0=0 +lon_0=21 +k=1 +x_0=0 +y_0=0 +ellps=WGS84 +towgs84=0,0,0,0,0,0 +units=m +no_defs
```

then click ok, and when you go edit the same CRS it turns it is

```
+proj=tmerc +lat_0=0 +lon_0=21 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0 +units=m +no_defs
```

#### #7 - 2013-08-26 03:52 AM - Frank Sokolic

Yes, on my system (Ubuntu 13.04, QGIS Master) if I create a custom CRS, e.g:

```
+proj=tmerc +lat_0=0 +lon_0=31 +k=1 +x_0=0 +y_0=0 +ellps=WGS84 +towgs84=0,0,0,0,0,0 +units=m +no_defs
```

and then click ok, and then edit the same CRS then `+axis=wsu` has been added:

```
+proj=tmerc +lat_0=0 +lon_0=31 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0 +units=m +no_defs
```

#### #8 - 2013-08-26 04:00 AM - Frank Sokolic

For me, the issue is that the proj4 definitions in SRS.db for these South African CRSs are not correct and shouldn't contain the +axis=wsu parameter as GIS users in South Africa (and Namibia and Lesotho) don't use south-facing coordinates. If the +axis=wsu parameter was removed from the relevant entries in SRS.db then southern African GIS users wouldn't have to use custom CRSs at all.

#### #9 - 2013-08-27 12:55 PM - Hien TRAN-QUANG

From what Leyan said, it seems that the error occurs when the custom CRS has too few parameters, compared to the found ones in the srs.db.

Shouldn't it only match the exact same number of parameters (apart maybe from +datum?) and not try to find a corresponding one, as it is contradictory to the purpose of a custom crs ?

#### #10 - 2013-08-29 12:08 PM - Hien TRAN-QUANG

I did a pull request [<https://github.com/qgis/Quantum-GIS/pull/838>]

Now createFromProj4 only checks for same parameters (except +datum)

#### #11 - 2013-08-29 11:51 PM - René-Luc ReLuc

- Pull Request or Patch supplied changed from No to Yes
- Category set to Projection Support
- Target version set to Version 2.0.0
- Status info set to Review requested

#### #12 - 2013-09-01 07:46 AM - Gavin Fleming

I found a simpler solution though it wasn't easily apparent. If you specify +axis=enu explicitly (the normal north-oriented axes) in the Custom CRS dialog when creating a new custom CRS then QGIS accepts the definition you provide. It does not show the +axis=enu switch in the resulting definition but does implement it and does not override it with +axis=wsu. This does seem to be a good approach, i.e. to force explicit definition of axis orientation, but does need to be documented somewhere. I am accustomed to not specifying axis orientation but am quite happy to specify it explicitly from now on. If that is the case then +axis=enu should perhaps remain in the CRS definition.

If I edit an existing custom CRS that I had tried to change to north facing when I created it, but where +axis=wsu was forcefully inserted by the bug, the change becomes effective and can be seen in Project Properties but does not show in the Custom CRS dialog.

e.g. to make a north-facing version of EPSG:22287 I edited the CRS I had created (follow the +axis flag in these definitions):

```
+proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753 +towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs.
```

I changed this to:

```
+proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=0 +y_0=0 +axis=enu +a=6378249.145 +b=6356514.966398753 +towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs.
```

In Project properties it now appears as the following and most importantly, it behaves correctly:

```
+proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=0 +y_0=0 +a=6378249.145 +b=6356514.966398753 +towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs
```

But if I open Custom CRS again it still shows:

```
+proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753 +towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs
```

Frank Sokolic wrote:

*For me, the issue is that the proj4 definitions in SRS.db for these South African CRSs are not correct and shouldn't contain the +axis=wsu parameter as GIS users in South Africa (and Namibia and Lesotho) don't use south-facing coordinates. If the +axis=wsu parameter was removed from the relevant entries in SRS.db then southern African GIS users wouldn't have to use custom CRSs at all.*

I believe the definitions are correct and it would be improper to alter a standard definition to suit a historical non-standard practice. GIS users have traditionally worked with negative values but this situation only arose because existing software was unable to implement south facing coordinates. Civil Engineers on the other hand have been correctly using TMSO for years via local software products. If you wish to use TMNO this should be viable by creating a custom CRS.

#### #14 - 2013-09-02 02:00 AM - Tim Sutton

There are a complex of issues, here I try to outline a roadmap to address them:

## Incorrect Proj4 definitions

/usr/share/proj/epsg lists Cape data with the with the easting/northing order switched. In other words in all of theses ``+axis=wsu`` needs to become ``+axis=swu``. There is already a ticket (filed in proj4 for this <http://trac.osgeo.org/proj/ticket/18>).

```
# Cape / Lo15
<22275> +proj=tmerc +lat_0=0 +lon_0=15 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo17
<22277> +proj=tmerc +lat_0=0 +lon_0=17 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo19
<22279> +proj=tmerc +lat_0=0 +lon_0=19 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo21
<22281> +proj=tmerc +lat_0=0 +lon_0=21 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo23
<22283> +proj=tmerc +lat_0=0 +lon_0=23 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo25
<22285> +proj=tmerc +lat_0=0 +lon_0=25 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo27
<22287> +proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo29
<22289> +proj=tmerc +lat_0=0 +lon_0=29 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo31
<22291> +proj=tmerc +lat_0=0 +lon_0=31 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>
# Cape / Lo33
<22293> +proj=tmerc +lat_0=0 +lon_0=33 +k=1 +x_0=0 +y_0=0 +axis=wsu +a=6378249.145 +b=6356514.966398753
+towgs84=-136,-108,-292,0,0,0,0 +units=m +no_defs <>

# Hartebeesthoek94 / Lo15
<2046> +proj=tmerc +lat_0=0 +lon_0=15 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo17
<2047> +proj=tmerc +lat_0=0 +lon_0=17 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo19
<2048> +proj=tmerc +lat_0=0 +lon_0=19 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo21
<2049> +proj=tmerc +lat_0=0 +lon_0=21 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo23
<2050> +proj=tmerc +lat_0=0 +lon_0=23 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo25
<2051> +proj=tmerc +lat_0=0 +lon_0=25 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
```

```
# Hartebeesthoek94 / Lo27
<2052> +proj=tmerc +lat_0=0 +lon_0=27 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo29
<2053> +proj=tmerc +lat_0=0 +lon_0=29 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo31
<2054> +proj=tmerc +lat_0=0 +lon_0=31 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
# Hartebeesthoek94 / Lo33
<2055> +proj=tmerc +lat_0=0 +lon_0=33 +k=1 +x_0=0 +y_0=0 +axis=wsu +ellps=WGS84 +towgs84=0,0,0,0,0,0,0 +units=m +no_defs <>
```

Until such time as an upstream fix happens, it would be good to fix these locally in the srs.db

## There are no north facing CRS definitions for the Lo Hartebeeshoek and the Lo Cape CRSs

---

There is currently not a ticket with EPSG to add these so for the interim it is suggested that they are added to the QGIS srsr.db with auth\_name 'ZANGI' (South Africa National Geospatial Information [department]).

In these cases all of the above definitions would be duplicated but with +axis=enu (which is the default but we should set it explicitly). We suggest the following schema for the auth\_id we would use for 'Lo projections this: SACRS\_NO\_<XX>' and for Hartebeeshoek projections this: TM\_CAPE\_NO\_<XX> where NO = North Orientated and <xx> is the zone.

## Test dataset

---

We should have a test dataset which can be used to verify that the CRS is properly detected. We propose two test datasets:

- A bounding box for South Africa in EPSG:4326
- A simple feature in SACRS\_NO\_23 which when reprojected on the fly to EPSG:4326 intersects with the country bbox above.

## EPSG formalised definitions

---

In the longer term, we should have EPSG definitions defined for the above SACRS and TM\_CAPE crs's and then remove the locally added definitions so that they are automatically added from EPSG.

**#15 - 2013-09-02 02:31 AM - craigleat -**

Tim,

For your proposed schema did you mean the following, as you appear to have them switched around?

Lo Hartebeeshoek94 -> SACRS\_NO\_<XX>

Lo Cape -> TM\_CAPE\_NO\_<XX>

**#16 - 2013-09-02 08:06 AM - Gavin Fleming**

- File test\_4326.csv added

- File test\_2050.csv added

- File test\_SACRS\_NO\_23.csv added

@craigleat: no, Tim is correct - we are proposing a new set of north-facing (or NO=north-oriented) definitions in addition to the existing set of south-facing (SO) definitions, such that each existing 'LO' code has a north-facing equivalent.

I attach here some test files to use. Each has the same triangle polygon in longlat (test\_4326), SACRS\_23 or "lo23/Hartebeesthoek"(test\_2050) and SACRS\_NO\_23 or "lo23/Hartebeesthoek north-facing" (test\_SACRS\_NO\_23).

These can be tested manually by loading with the delimited text loader and specifying the appropriate CRS and enabling on the fly CRS transformation. When all is working, all three polygons should coincide in the middle of South Africa.

When the bug that this ticket refers to is fixed and a new definition has been created, then test\_SACRS\_NO\_23 should overlay 4326.

When the +axis flag in the existing definition has been changed from +axis=wsu to +axis=swu then test\_SACRS\_NO\_23 should overlay 4326.

**#17 - 2013-09-02 08:16 AM - Gavin Fleming**

- File test\_ZA\_BB.csv added

**#18 - 2013-09-03 05:26 AM - Gavin Fleming**

- Priority changed from Severe/Regression to High

Let's hang five on Tim's "Incorrect project definitions" above. Technically they are correct according to the EPSG but the jury is out on which one of swu or wsu is the predominant axis order in practice in South Africa. Perhaps we should leave them as is and tell users they must conform with EPSG convention, until we can confirm otherwise. If we confirm in due course that swu is the local convention, then we should make the change.

Regarding the new north-facing SACRS ('ZANGI') CRS definitions, those are fine as Tim has described.

**#19 - 2013-09-03 06:07 AM - Gavin Fleming**

- Priority changed from High to Severe/Regression

I mistakenly downgraded this earlier - the original bug this Issue describes is still a blocker until resolved.

**#20 - 2013-09-03 04:19 PM - Tim Sutton**

Further to this, I have will insert the following CRS definitions into the srs.db. Due to the last minute nature of this commit, any corrections to these will need to be made in future release of QGIS 2.x

<quote>  
INSERT INTO "tbl\_ellipsoid" VALUES;

INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;

INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;  
INSERT INTO "tbl\_srs" VALUES;

</quote>

Notes on acronyms used:

- ZANGI - South Africa National Geospatial Information [Dept]
- CPNO<xx> (e.g. CPNO33) - Cape Datum North Oriented on centered longitude 33
- HBKNO<xx> (e.g. HBKNO33) - Hartebeeshoek North Oriented on centered longitude 33
- NO\_<xx> (e.g. NO\_15) - North Oriented on longitude 15

#21 - 2013-09-03 04:50 PM - Tim Sutton

commit:14f639d Adds the above CRS's

#22 - 2013-09-06 12:35 PM - Giovanni Manghi

- Target version changed from Version 2.0.0 to Future Release - High Priority

#23 - 2013-09-29 06:19 AM - Gavin Fleming

bug appears to be fixed in 2.1.0 Master (though the new north-oriented CRS definitions Tim added don't appear yet)

#24 - 2014-01-24 07:04 AM - Giovanni Manghi

- Status changed from Open to Feedback

Gavin Fleming wrote:

| bug appears to be fixed in 2.1.0 Master (though the new north-oriented CRS definitions Tim added don't appear yet)

can we close this?

#25 - 2014-01-26 01:24 AM - Gavin Fleming

Yes you can close it, thanks all. The original issue as well as the related issues brought up in the discussion are all fixed in master.

#26 - 2014-01-26 01:28 AM - Giovanni Manghi

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

Files

test_4326.csv	48 Bytes	2013-09-02	Gavin Fleming
test_2050.csv	209 Bytes	2013-09-02	Gavin Fleming
test_SACRS_NO_23.csv	215 Bytes	2013-09-02	Gavin Fleming
test_ZA_BB.csv	190 Bytes	2013-09-02	Gavin Fleming