

## QGIS Application - Feature request #1035 automated generation of srs.db (in line with postgis, proj and gdal)

2008-04-11 07:42 AM - Richard Duivenvoorde

<b>Status:</b> Closed	
<b>Priority:</b> Low	
<b>Assignee:</b> nobody -	
<b>Category:</b> Projection Support	
<b>Pull Request or Patch supplied:</b>	<b>Resolution:</b> fixed
<b>Easy fix?:</b> No	<b>Copied to github as #:</b> 11095
<b>Description</b>	
<p>after some discussion on the list about the qgis-epsg definitions in srs.db not being up to date with gdal and proj it seems better to automate the generation of srs.db instead of adding changes line by line.</p> <p>In a short discussion on #gdal FrankW proposed to use/adjust the script that is generating the lists for gdal, proj and postgis: <a href="http://trac.osgeo.org/gdal/browser/trunk/gdal/swig/python/scripts/epsg_tr.py">http://trac.osgeo.org/gdal/browser/trunk/gdal/swig/python/scripts/epsg_tr.py</a> so it can generate sqlite-update/inserts for qgis also.</p> <p>But only if it does not interfere with the current generation of srs.db (though looking in trac it seems it's being updated by hand recently...)</p>	

### History

#### #1 - 2008-04-13 11:30 PM - Richard Duivenvoorde

- there is an issue with trailing zero's for the k-factor (#1037) and gdal. Don't know if we should combine 'solving' it with this one by removing trailing zero's, or that it should be fixed in qgis code elsewhere ...

#### #2 - 2008-04-19 05:04 AM - Richard Duivenvoorde

Some remarks that people made to me when talking about this (hopefully I can make stuff clear here):

- if I understand it ok, there is also the option that certain EPSG offers more than one transformation to WGS84 for a datum see

<http://www.nabble.com/epsg-parameters-td16447986.html>

Instead of skipping the "+towgs84=" part, is it an idea then to give some notice (in comments?) or link to a osgeo/qgis-wiki page in which we show the different possible transformations? So an ignorant user at least can see there is an issue with the chosen projection/transformation and go some place to choose the right parameters, or finds more info?

- I heard a rumour that things have changed in a new(er?) version of proj, in which the towgs84-part in the parameter-list is no longer optional? Is this true, and if so do we have to do something with this?

#### #3 - 2008-05-25 06:50 PM - hamish -

see also bug #1079

#### #4 - 2008-05-25 07:24 PM - hamish -

IMHO it is absolutely critical that srs.db be generated automatically. It is just too big a task to edit the 3500 standard EPSG codes by hand for every new release.

The srs.db as now is broken. +towgs84= params are missing when they are part of the EPSG definition. The ambiguity arises when +datum=foo is given but there are multiple ways to get to foo, such as any number of local 3 term, 7 term, or NTV2 distortion grid file transform methods.

if the EPSG code includes +towgs84= you already have your explicit answer of which transform params to use. but those seem to be stripped out in QGIS's srs.db. e.g. EPSG code # 31370. also +datum= is in the EPSG file for NZMG (epsg 27200) but removed from QGIS's srs.sb entry. That's an example of one with multiple choices (see comment in bug #1079).

So that missing info is nothing to do with the ambiguity cases AFAIK. (?)

fyi, grass throws up a GUI in these cases asking the user which one they'd like to use:

[http://grass.osgeo.org/grass63/screenshots/images/grass63\\_startup2.png](http://grass.osgeo.org/grass63/screenshots/images/grass63_startup2.png)

I think it is a great idea that work should go into adding a --qgis extension to GDAL's python epsg processing script. outsource it to the experts! For GRASS we (ie Markus) will typically sync and regenerate our \$GISBASE/etc/ogr\_csv/\* files after each new GDAL release/before each one of our releases. It seems to work well. We also have datum.table, datumtransform.table, ellipse.table, and proj\* files in our (share) etc/ directory to draw from. You'd have to ask Paul Kelly for details of how it all works together.

The k-factor bug is just a symptom I think, with the real cause being that QGIS is both out-of-sync with itself (srs.db and other internal code??) and expecting exact string matches. Both of which are rather brittle.

see GRASS's libgis for code for removing trailing 0s from a numeric string.

the change in GDAL which seemed to have raised the issue happened due to the replacement of atof() in the code with a i18n comma as decimal place capable version, which lead to FP differences in the lesser significant bits (well it being treated as something other than a string too). The new fn has been modified AFAIK in GDAL svn, but a new epsg file for proj has not yet been generated (I checked the CVS yesterday). Use the epsg file that came with PROJ 4.5.0 if you are worried.

the "towgs84-part in the parameter-list is no longer optional" is more to do with the change in gdal 1.5.0? regarding if datum transforms will happen if datum (or ellps?) params are explicitly given on the from and +to coord systems.

hope that helps, and that I'm not too confused by how this all fits together.

summary: srs.db is broken as compared to proj's epsg text file. automate the generation of it ASAP or continue to feel the pain. eventually QGIS will need a datum picking GUI to resolve the ambiguous cases. If doing exact string matching, make sure your sources are in sync.

Hamish

**#5 - 2008-05-26 07:10 AM - hamish -**

Paul Kelly wrote on the grass-users list:

*There are actually only a few +datum= codes recognised by PROJ.4 - most of them are specific to GRASS. So it isn't meaningful to include a +datum=xxxx if the PROJ string is destined to be used by something other than GRASS. The OGR functions that GRASS uses to interpret EPSG codes somehow manage to return the full EPSG name for the datum, which is then compared to GRASS's datum.table (the second field in this table holds the full EPSG name). Combined with datumtransform.table that gives GRASS enough information to go on to present the user with a choice of datum transform parameters. However the full EPSG name isn't included in the PROJ.4 string, and if that is all QGIS is using then maybe*

*that's where its deficiency stems from. It is all very messy and complicated.*

*Paul*

#### #6 - 2008-05-26 11:08 AM - Jürgen Fischer

Replying to [comment:5 hamish]:

*fyi, grass throws up a GUI in these cases asking the user which one they'd like to use:*  
[http://grass.osgeo.org/grass63/screenshots/images/grass63\\_startup2.png](http://grass.osgeo.org/grass63/screenshots/images/grass63_startup2.png)

Good idea - but I just noticed that QGIS also does this if you configure it to "Prompt for Projection" in Settings/Options in the projection tab. The default is to fallback to WGS84.

#### #7 - 2008-05-26 11:31 PM - hamish -

Replying to [comment:7 jef]:

*Good idea - but I just noticed that QGIS also does this if you configure it to "Prompt for Projection" in Settings/Options in the projection tab. The default is to fallback to WGS84.*

No it is something different. That prompts you for the projection to use, letting you pick from the entries in QGIS's srs.db SQLite database. But the srs.db entries are currently missing data :)

What I was speaking of with GRASS is a companion GUI window that, once you have selected an EPSG "projection" code, lets you pick the associated datum transform parameters to use with that. But "projection" is the wrong thing to call an EPSG code.....

terminology I'm using, as I understand it:

[PROJ.4 parameter in brackets]

- projection [+proj=] - mathematical bending method to impose 2D from a 3D world (e.g. tmerc [transverse mercator], lcc [lambert conformal conic]). They are often centered at a specific lat/lon [+lat\_0= +lon\_0=]

- datum [+datum=] - where the center of mass of the world is (where you place 0,0,0), typically defined with a standard ellipsoid (deformation of geoid from a perfect sphere) Datums created after the satellite age are typically very close. WGS84 defines both a datum and an ellipsoid with the same name, so that's slightly confusing. ?{If the software knows how it can expand +datum= into +ellps=,+a=,+k=. +a,+b are radius of Earth (major/minor?), +k is scaling factor.}?

- false eastings & northings [+x\_0= +y\_0=] - number to add to the projected coordinates. often to keep them always positive or distinct from other systems, often in the millions

- SRS [+init=epsg:1234] - spatial reference system (eg the EPSG codes) a commonly used combination of a projection, datum and/or ellipsoid, false eastings & northings, etc. Strictly speaking the "projection" is just the +proj= part, not the whole SRS even though it is often called that. Not all SRSs define

a datum, in those cases you just get the ellipsoid(?).

- datum transform params [+towgs84=] - any given SRS *may* use slightly different coefficients to convert to wgs84 (as the middleman datum). often these will be given as 3 or 7 numbers which are tuned to an average for the entire country or to give very good results for a single province. So for a given SRS you get to choose which state or province you want it tuned for, or a generalized set of terms. There are also distortion grids (aka NTV2 or nadgrids) you can get which supply local corrections for the entire country. These will typically give better results than a 3 or 7 term +towgs84= param, but they add a file dependency to the mix which adds installation difficulties.

further reading:

[http://grass.osgeo.org/wiki/Gis\\_Concepts](http://grass.osgeo.org/wiki/Gis_Concepts)

if you look in /usr/share/proj/epsg you will see these things, along with some like:

```
# Qornoq 1927 / Greenland zone 5 west
# Unable to translate coordinate system EPSG:2304 into PROJ.4 format.
```

Perhaps in that one PROJ doesn't do polar stereographic?

and some old ones like:

```
# Dabola 1981 / UTM zone 28N (deprecated)
<2063> +proj=utm +zone=28 +a=6378249.2 +b=6356515 +towgs84=-23,259,-9,0,0,0,0 +units=m +no_defs <>
```

hope that helps, and that I've not butchered the concepts too badly.

Hamish

**#8 - 2008-05-30 06:19 AM - Maciej Sieczka -**

Replying to [comment:8 hamish]:

```
- SRS [+init=epsg:1234] - spatial reference system (eg the EPSG codes) a commonly used
  combination of a projection, datum and/or ellipsoid, false eastings & northings, etc. Strictly > speaking the "projection" is just the +proj= part,
  not the whole SRS even though it is often
  called that.
```

And it should not be called that. "Projection" is merely a component of a "spatial reference system" aka "coordinate system".

```
and some old ones like:
```

```
1. Dabola 1981 / UTM zone 28N (deprecated)
```

"Deprecated" are "broken" rather than "old ones". They are kept in the EPSG database though for keeping the track of changes. Eg. 2171 is "deprecated" because it has a huge error. It's valid counterpart was added as 3120, about a year ago.

**#9 - 2008-07-10 11:17 AM - Maciej Sieczka -**

- Status changed from Open to Closed

- Resolution set to fixed

The attached script can be used to generate the srs.db in accordance with the installed GDAL. Closing ticket.

**#10 - 2009-08-22 12:57 AM - Anonymous**

Milestone Version 1.0.0 deleted

**Files**

---

qgis_srs.sh	5.84 KB	2008-07-10	Maciej Sieczka -
-------------	---------	------------	------------------