

## QGIS Application - Bug report #843

### Import PostGIS data into GRASS via QGIS toolbox

2007-11-30 06:45 AM - Otto Dassau

|  |                |                                     |
|--|----------------|-------------------------------------|
| <b>Status:</b>   | Closed         |                                     |
| <b>Priority:</b>   | Normal         |                                     |
| <b>Assignee:</b>   | Jürgen Fischer |                                     |
| <b>Category:</b>   | GRASS          |                                     |
| <b>Affected QGIS version:</b>  | master         | <b>Regression?:</b> No              |
| <b>Operating System:</b>   | Debian         | <b>Easy fix?:</b> No                |
| <b>Pull Request or Patch applied:</b>  | No             | <b>Resolution:</b> fixed            |
| <b>Crashes QGIS or corrupts data:</b>  | No             | <b>Copied to github as #:</b> 10902 |
| <b>Description</b>   |                |                                     |
| <p>I tried to import [[PostGIS]] data into GRASS with the toolbox command v.in.ogr. I get an error "Layer not available" in the toolbox with the used command:</p> <pre>v.in.ogr 'dsn=PG:dbname=test host=localhost port=5432 user=dassau' layer=public.roads output=roads -o</pre> <p>the problem seems to be that if you use v.in.ogr from QGIS it takes the layer name together with the name of the schema and so the "wrong" layer definition in the command line looks like this:</p> <pre>layer=public.roads</pre> <p>which is not the way to use it with v.in.ogr. Here I only need the layer name without schema:</p> <pre>layer=roads</pre> <p>thanks a lot<br/>Otto</p> |                |                                     |
| <b>Related issues:</b>   |                |                                     |
| Related to QGIS Application - Bug report # 6599: Tables in current_schema() g...   |                | <b>Closed</b> <b>2012-11-01</b>     |

#### Associated revisions

##### Revision eefe6791 - 2007-12-01 03:41 PM - Jürgen Fischer

fix #843

git-svn-id: <http://svn.osgeo.org/qgis/trunk/qgis@7699> c8812cc2-4d05-0410-92ff-de0c093fc19c

##### Revision b537a16e - 2007-12-01 03:41 PM - Jürgen Fischer

fix #843

git-svn-id: <http://svn.osgeo.org/qgis/trunk@7699> c8812cc2-4d05-0410-92ff-de0c093fc19c

#### History

##### #1 - 2007-12-01 06:46 AM - Jürgen Fischer

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

fixed in commit:b537a16e (SVN r7700)

**#2 - 2007-12-03 03:05 AM - horst-duester -**

- Resolution deleted (fixed)
- Status changed from Closed to Feedback

The commit:b537a16e (SVN r7700) solution doesn't seem to be a good direction. In principle v.in.ogr supports postgres schemas and QGIS should always use qualified table names.

In commit:b537a16e (SVN r7700) QGIS uses mOgrLayers[current]. This is the layer name in the QGIS TOC loaded via Postgres Connection and not the name of the connected schema qualified table. As the result you get the error message "Layer not found"

I'm a little bit confused due to the change of content of mOgrLayers[current]. commit:97e9a169 (SVN r7668) has the full qualified table name. In commit:b537a16e (SVN r7700) mOgrLayers[current] contains only the table name.

**#3 - 2007-12-03 04:14 AM - Jürgen Fischer**

- Status changed from Feedback to Open

Replying to [comment:2 horst.duester]:

*The commit:b537a16e (SVN r7700) solution doesn't seem to be a good direction. In principle v.in.ogr supports postgres schemas and QGIS should always use qualified table names.*

Which breaks things for Otto. The compile time checks to figure out if grass/ogr actually supports schemas obviously aren't enough. For now the unqualified tablename is used, which should work everywhere unless we actually need the schema name - which I for one consider an edge case. We need a better test.

*In commit:b537a16e (SVN r7700) QGIS uses mOgrLayers[current]. This is the layer name in the QGIS TOC loaded via Postgres Connection and not the name of the connected schema qualified table. As the result you get the error message "Layer not found"*

No, it's the name of the table just without the schema.

**#4 - 2007-12-03 07:28 AM - horst-duester -**

I don't get it work! All loaded layer are named without schema. I'm not able to load a layer from different schema than public.

My system:  
GDAL-1.4.2  
GRASS-6.2.3

**#5 - 2007-12-03 07:41 AM - Otto Dassau**

Replying to [comment:4 horst.duester]:

*I don't get it work! All loaded layer are named without schema. I'm not able to load a layer from different schema than public.*

*My system:*

*GDAL-1.4.2*

*GRASS-6.2.3*

I worked and tested with public schema and it works, now that layers are named without schema. I haven't tested with other schemas - but that's the problem I guess.

Layers named without schema, can only be load data for schema public and layers named with schema can be loaded for schemas different from schema public.

Maybe, it could fixed with an if-statement. If schema is public use layername without schema else with schema?

#### **#6 - 2007-12-03 08:11 AM - Jürgen Fischer**

Replying to [comment:5 dassau]:

*I worked and tested with public schema and it works, now that layers are named without schema. I haven't tested with other schemas - but that's the problem I guess.*

The code makes a distinction between GDAL >=1.4 and earlier versions. With some <1.4 the schema is removed and a warning is put out.

Which version of GDAL and GRASS are you running? Looks like your running a GDAL >=1.4 and your v.in.ogr still doesn't support schemas.

IIRC my GDAL is 1.4.3 and v.in.ogr has the same problem (I'll verify this tonight).

I guess the GRASS version should also put into consideration.

#### **#7 - 2007-12-03 08:36 AM - Otto Dassau**

Replying to [comment:6 jef]:

*Replying to [comment:5 dassau]:*

*I worked and tested with public schema and it works, now that layers are named without schema. I haven't tested with other schemas - but that's the problem I guess.*

*The code makes a distinction between GDAL >=1.4 and earlier versions. With some <1.4 the schema is removed and a warning is put out.*

*Which version of GDAL and GRASS are you running? Looks like your running a GDAL >=1.4 and your v.in.ogr still doesn't support schemas.*

*IIRC my GDAL is 1.4.3 and v.in.ogr has the same problem (I'll verify this tonight).*

*I guess the GRASS version should also put into consideration.*

I work on debian testing and use GRASS 6.3~cvs20071023 from: deb <http://debian.gfoss.it/> lenny main. GDAL comes from an official debian repository: gdal-bin 1.4.2-1 and libgdal1-grass -> 1.4.1-1

**#8 - 2007-12-03 03:52 PM - Jürgen Fischer**

workaround in commit:7af34268 (SVN r7711).

GDAL prepends the schema only to tables that are not in the current schema.

The current schema defaults to the username, if such a schema exists or public otherwise. We would need to introduce a postgres dependency in the grass plugin to figure out the current schema. I'm not sure if that's the right thing to do.

The workaround simply removes the schema from the user's and public tables. This makes public tables accessible, unless a user schema exists. If it does public tables will still be inaccessible.

**#9 - 2007-12-04 04:17 AM - Jürgen Fischer**

- *Status changed from Open to Closed*

- *Resolution set to fixed*

commit:5b4e6997 (SVN r7712) solves the schema problem in the postgres provider and removes the workaround.

**#10 - 2009-08-22 12:52 AM - Anonymous**

Milestone Version 0.9.1 deleted

**#11 - 2012-11-01 09:47 AM - Sandro Santilli**

- *Pull Request or Patch supplied set to No*

- *Affected QGIS version set to master*

- *Crashes QGIS or corrupts data set to No*

I would like to drop this schema omission workaround for a bug which is in the v.in.ogr user for reasons described in #6599

Can anyone give step-by-step instructions on how to reproduce that old problem ?

Reading the original submission I believe the bug is in the v.in.ogr code, not in qgis...

That is I agree with Horst in comment 2

**#12 - 2012-11-01 09:50 AM - Sandro Santilli**

- *Priority changed from Low to Normal*

- *Status changed from Closed to Reopened*

Also, I didn't find any conditional in the code dropping the schema qualification

**#13 - 2012-11-01 09:54 AM - Sandro Santilli**

- *Resolution deleted (fixed)*

So commit:0dc01f5 removes the workaround.

Otto, Horst, could you try to do the exports today, of both tables in current schema and in foreign schema ?

If the problem persists must be fixed at the appropriate place IMHO

**#14 - 2012-11-01 09:57 AM - Giovanni Manghi**

- *Status changed from Reopened to Feedback*

**#15 - 2013-01-05 05:55 PM - Jürgen Fischer**

- *Status changed from Feedback to Closed*

- *Resolution set to fixed*

closing for the lack of feedback.