

QGIS Application - Bug report #8481

"Errors: SUCCESS: 6 attribute(s) added. ERROR: field with index 4 is not the same!" when adding columns to vectors

2013-08-18 02:31 AM - Giovanni Manghi

Status:	Closed	
Priority:	Severe/Regression	
Assignee:		
Category:	Vectors	
Affected QGIS version:	master	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:		Resolution:
Crashes QGIS or corrupts data:		Copied to github as #: 17248
Description		
<p>Add a column in a PostGIS or Spatialite vector (or Shapefile, but does not seems to happen always in this case), then toggle editing to save edits.</p> <p>The new status bar will pop up, and if you go read the message you will see something like</p> <p>Errors: SUCCESS: 6 attribute(s) added. ERROR: field with index 4 is not the same!</p> <p>The edit is usually successful anyway, sometimes it is needed to toggle editing again.</p>		

Associated revisions

Revision 24b3ed17 - 2014-02-17 02:07 AM - Jürgen Fischer

postgres provider: fix detection of character fields (related to #8481)

Revision dc9148c6 - 2014-06-11 08:25 PM - Jürgen Fischer

enhance error message when change/added attributes mismatch (refs #8481)

History

#1 - 2013-08-27 08:45 AM - Matthias Kuhn

This is related to definitions of field types in the QGIS provider implementation which do not match the field type which is created by the driver in the background.

Therefore it is essentially to have a list of affected providers / field type pairs.

OGR Provider : Date is fixed in commit:d758340c

OGR Provider : DateTime is not supported by all OGR drivers. E.g. Shapefile does not support it. Therefore, this fieldtype needs special attention and we need to find eigher a way to query OGR drivers if they support this type or create and maintain a blacklist/whitelist.

#2 - 2013-08-28 01:02 AM - Matthias Kuhn

OGR Provider : DateTime Blacklisted this datatype for ShapeFile in commit:a8a1090. Please help to extend the list for other formats which do not support this field type. Querying OGR for support is unfortunately not possible according to #gdal.

#3 - 2013-08-29 02:22 AM - Giovanni Manghi

For what it can matter I noticed that on a postgis layer there is no error just when adding "numeric" and "decimal" fields (cannot test real and double because of #8480) and it gives the error when choosing any other type.

In my opinion among the list of blockers this issue is the really important one as it is a regression of a pretty basic (q)qgis function. Any chance to have this fixed (in time)?

thanks in advance

#4 - 2013-08-29 02:25 AM - Giovanni Manghi

Giovanni Manghi wrote:

For what it can matter I noticed that on a postgis layer there is no error just when adding "numeric" and "decimal" fields (cannot test real and double because of #8480) and it gives the error when choosing any other type.

on SL the error shows when adding integer/double column types, but not "text".

#5 - 2013-08-29 05:50 AM - Matthias Kuhn

commit:2f2e088 solves this partially

The other half of the problem is, that the [qgis] spatialite provider indicates, that you can define length and precision of these fields between 0 and 20, while sqlite does in fact not support this. Therefore afterwards any column will have a default length (0) and default precision (0) assigned.

I would propose to remove the option to define these (useless) options from the spatialite drivers.

For now: As long as you don't change them (i.e. leave them set to 0) you should not get any warnings.

#6 - 2013-08-29 06:23 AM - Giovanni Manghi

Hi Matthias

For now: As long as you don't change them (i.e. leave them set to 0) you should not get any warnings.

In SL I get the error also when creating integer fields even when leaving 0 as width, does the above patch fix this case? Thanks!

#7 - 2013-08-29 06:32 AM - Matthias Kuhn

Giovanni Manghi wrote:

In SL I get the error also when creating integer fields even when leaving 0 as width, does the above patch fix this case? Thanks!

Exactly

#8 - 2013-08-29 07:01 AM - Giovanni Manghi

Matthias Kuhn wrote:

Giovanni Manghi wrote:

In SL I get the error also when creating integer fields even when leaving 0 as width, does the above patch fix this case? Thanks!

Exactly

sweet. I guess that fixing postgresql is a different beast(?). thanks again.

#9 - 2013-08-29 07:27 AM - Matthias Kuhn

Fixed the trivial ones in commit:a9c05d7

Remaining problems are:

- Postgres: Field length and precision are not correctly read from the database. (For char type with configurable length)
- Spatialite: Field length and precision are configurable although they should not be. (Easy fix, but I need confirmation that the length is really not used by anything)

Giovanni, could you also test the other providers, that the changes did not introduce any errors which have been masqueraded by the add attribute dialog behavior before. Thank you very much.

#10 - 2013-09-01 01:46 AM - Giovanni Manghi

Giovanni, could you also test the other providers, that the changes did not introduce any errors which have been masqueraded by the add attribute dialog behavior before. Thank you very much.

Hi Matthias, for the moment I re-tested postgis, SL and shapes.

Postgis:

now I see the error only when adding a char (text, fixed length) column

SL:

I see no errors when adding columns and leave width/precision to 0, but when adding an integer column I see that the values stored are still decimal

SHP:

no problems

#11 - 2013-09-06 12:35 PM - Giovanni Manghi

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

#12 - 2013-10-20 04:04 AM - Matthias Kuhn

Fixed the spatialite problems in commit:249526b

No other value than 0 is allowed for integer and real columns.

I'm pretty confident, that this is ok. Length / precision settings will be lost anyway when saving the fields to the provider according to <https://www.sqlite.org/datatype3.html#storageclasses>

#13 - 2013-10-20 04:06 AM - Matthias Kuhn

I was hesitating to backport this to the 2.0 fixes but if there is demand, I can do so. The task itself is easy.

#14 - 2014-01-28 04:08 AM - Matthias Kuhn

- Status changed from Open to Feedback

Is this still an issue?

Looking at my earlier comments, I didn't take care of this one: "Postgres: Field length and precision are not correctly read from the database. (For char type with configurable length)"

Can somebody confirm this (or are there others affected)

#15 - 2014-01-29 11:42 AM - Giovanni Manghi

Matthias Kuhn wrote:

Is this still an issue?

Looking at my earlier comments, I didn't take care of this one: "Postgres: Field length and precision are not correctly read from the database. (For char type with configurable length)"

Can somebody confirm this (or are there others affected)

when I try to add a "char" column on a PostGIS layer, on save I get

Could not commit changes to layer sele

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 53 is not the same!

#16 - 2014-02-08 01:44 AM - Giovanni Manghi

- Target version changed from Future Release - High Priority to Version 2.2

- Status changed from Feedback to Open

#17 - 2014-02-16 03:38 PM - Giovanni Manghi

Giovanni Manghi wrote:

Matthias Kuhn wrote:

Is this still an issue?

Looking at my earlier comments, I didn't take care of this one: "Postgres: Field length and precision are not correctly read from the database. (For char type with configurable length)"

Can somebody confirm this (or are there others affected)

when I try to add a "char" column on a PostGIS layer, on save I get

Could not commit changes to layer sele

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 53 is not the same!

PostGIS:

still getting error when saving a newly created char column, example

Could not commit changes to layer evora

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 8 is not the same!

after discarding the edits the column seems that can be updated without errors

#18 - 2014-02-18 11:02 AM - Giovanni Manghi

- Status changed from Open to Closed

- Resolution set to fixed/implemented

Giovanni Manghi wrote:

Giovanni Manghi wrote:

Matthias Kuhn wrote:

Is this still an issue?

Looking at my earlier comments, I didn't take care of this one: "Postgres: Field length and precision are not correctly read from the database. (For char type with configurable length)"

Can somebody confirm this (or are there others affected)

when I try to add a "char" column on a PostGIS layer, on save I get

Could not commit changes to layer sele

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 53 is not the same!

PostGIS:

still getting error when saving a newly created char column, example

Could not commit changes to layer evora

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 8 is not the same!

after discarding the edits the column seems that can be updated without errors

a recent commit by Jürgen has fixed this.

#19 - 2014-03-05 11:26 AM - Giovanni Manghi

- Status changed from Closed to Reopened

- Target version changed from Version 2.2 to Version 2.4

- Resolution deleted (fixed/implemented)

I have seen this again, on master, working on a shapefile

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 8 is not the same!

#20 - 2014-03-05 11:35 AM - Jürgen Fischer

Giovanni Manghi wrote:

I have seen this again, on master, working on a shapefile

Errors: SUCCESS: 1 attribute(s) added.

ERROR: field with index 8 is not the same!

And which types did you use?

#21 - 2014-03-05 05:41 PM - Giovanni Manghi

And which types did you use?

will try to replicate again and let you know.

#22 - 2014-03-06 04:52 PM - Giovanni Manghi

And which types did you use?

just got it when after adding an integer column to a point file, and filling it with field calculator and the rand function.

#23 - 2014-04-17 09:58 AM - Giovanni Manghi

and just seen this on a PostGIS layer, with QGIS master

Errors: SUCCESS: 1 attribute(s) added.

ERROR: the count of fields is incorrect after addition/removal of fields!

#24 - 2014-06-03 03:34 AM - Matthias Kuhn

When updating this issue, **please provide details** not only about the provider, but also the **data type** you used, this considerably increases the chances of having the issue fixed ;-)

#25 - 2014-06-04 01:16 AM - Giovanni Manghi

Matthias Kuhn wrote:

*When updating this issue, **please provide details** not only about the provider, but also the **data type** you used, this considerably increases the chances of having the issue fixed ;-)*

Hi Matthias,

will do it. Anyway I have already done it (sometimes), see for example in #8481-22

cheers!

PS

the problem with this issue is that is not easily replicable.

#26 - 2014-06-04 02:36 AM - Jürgen Fischer

Giovanni Manghi wrote:

will do it. Anyway I have already done it (sometimes), see for example #8481-22

Well, the problem is the mapping between QgsField (name, type, precision) to a provider specific type on creation of the field and the retrieval of field definitions from a provider and their mapping back to QgsField. After the field is created, the provider retrieves the new field definitions and checks if the QgsField created from that, matches what you originally entered. A mismatch produces the error message.

For instance the postgres provider didn't care to retrieve the length of character fields and therefore each character field with a given length constraint would produce that error - because the provider wouldn't reflect the length constraint in the QgsField it reported (and in turn not limit edit widget in forms to that given length later and lead to a commit error if the user entered a value exceeding that length). That's what the related commit above fixes.

The spatialite fix however removes the ability to specify a length constraint for float and integer field in spatialite databases as spatialite would ignore and drop constraint anyway and lead to the same error message - although it's actually a different problem.

With other types and/or providers there might be other different mismatches (the name might change case or be truncated to a certain length, types might change (eg. when an integer width exceeds some limit, it might be turned into a real type). So to reproduce we need to know what combination was actually

used (point file and integer doesn't really cut it ;)).

But reporting and fixing another combination that produces this error wouldn't 'fix' this ticket. Maybe we should close it in favor of individual tickets for each failing combination we find.

#27 - 2014-06-04 01:54 PM - Matthias Kuhn

maybe we should put some more details into the log containing dumps of the specified and provider reported field definitions, so there is a bit more information about what failed.

#28 - 2014-06-11 08:13 AM - Giovanni Manghi

- Status changed from Reopened to Feedback

It is a few days now that I don't see anymore this kind of messages, before it was quite common.

Maybe we can lower the priority of this ticket?

#29 - 2014-06-11 09:31 AM - Jürgen Fischer

Giovanni Manghi wrote:

It is a few days now that I don't see anymore this kind of messages, before it was quite common.

Maybe we can lower the priority of this ticket?

As said to me this are individual issues - although there is always the same error message popping up. So I close this and open new ones when a new combinations where creation and retrieval of types disagree is identified.

#30 - 2014-06-11 09:34 AM - Giovanni Manghi

- Status changed from Feedback to Closed