

QGIS Application - Feature request #964

PostGIS internationalization problems on database object names

2008-02-28 08:04 PM - Steven Mizuno

Status:	Closed	
Priority:	Low	
Assignee:	Jürgen Fischer	
Category:	Data Provider	
Pull Request or Patch supplied:		Resolution: fixed
Easy fix?:	No	Copied to github as #: 11023
Description		
<p>While investigating uppercase support issues on [[PostGIS]] layers, I noticed that international characters likely would not work for [[PostgreSQL]] object names, but would work for data values.</p> <p>This may be a reason for some complaints about [[PostGIS]] layers I have seen in the various QGIS forums, something like: "I loaded a [[PostGIS]] layer and don't see any data."</p> <p>I decided to attempt to fix this problem. I believe that I have done so and expect to submit a patch in a few days after some further testing. There is one more aspect of [[PostgreSQL]] object names I haven't tackled yet - the use of quotes (double and single) and backslash. This requires escaping the strings, plus the rules have changed over various versions.</p> <p>I also fixed several other problems with [[PostGIS]] layer handling, some of which hindered testing internationalization. See below for the details.</p> <p>Also, I created a list of my expectations for [[PostGIS]] layers, which is attached.</p> <p>I am using a [[PostgreSQL]] server with UTF8 as its database encoding, so all of these notes are based on that. I have considered what happens if a database has some other encoding, but I haven't tested this very much. Once connected there shouldn't be a problem as long as UTF8 is used for client encoding.</p> <p>Some observations:</p> <p>On looking at various source files concerning [[PostGIS]] layers I found that strings passed to libpq and returned were not of the proper character set. Qt uses Unicode in QStrings, which are 16-bit entities, while libpq takes C strings, which have 8-bit characters. Thus, there must be a character set conversion for communication with [[PostgreSQL]] databases.</p> <p>From the Qt documentation I found that the default codec for C strings is Latin1.</p> <p>The local 8 bit encoder in en_US locale appears to be UTF-8. This depends on the locale determined or selected in QGIS.</p> <p>Various QGIS [[PostgreSQL]] modules set the client encoding on the database to UNICODE when they connect. Most string handling for calls to libpq/results from libpq were using default encoding. A few had specific encoding set, but these were inconsistent - some Local8Bit, some Utf8. This likely causes the wrong character set translation to be used, unless all strings passed to/from the database are UTF-8 encoded.</p> <p>The solution:</p> <p>I have tried to find all occurrences of the libpq calls and apply toUtf8() or fromUtf8() QString functions as appropriate to the strings (both QString and C strings), except for when connecting to a database where I used toLocal8Bit()/fromLocal8Bit. This results in uniform behavior for international characters when used in database objects and data. However, there is a potential for problems on database names if the locale uses a different encoding from the database default, if the database name contains non-ASCII characters.</p>		

I have tested fairly extensively with databases/schemas/tables/column names having spaces, uppercase, and international characters - load layer, save & load project, attribute table, identify.

I have not worked on the SPIT plug-in, two threaded modules (they appear to be no longer used), and ...postgisbox... files (no code is affected).

In the process of fixing and testing the string handling on libpq calls I discovered a number of little problems in the [[PostGIS]] layer modules:

- + that a test for a blank stored password used QString::null instead of QString::isEmpty() so a Password dialog was never raised. Fixed
- + a side effect of the improper password test appears that the New Connection dialog was made to always store the password. Fixed so the Save Password checkbox actually does something.
- + type detection did not return the correct name for int2 (it was null) because the query used to retrieve type information needed a SELECT DISTINCT. The int2 type returns 2 rows, others don't (I don't know why), so the query fails. Fixed
- + QgsDbSourceSelect::getTableInfo() has a note about not being able to get a query to work on uppercase names. I provided a query that does work and removed some code farther down that was a work-around.
- + found that int8 data values were not handled correctly due to Int type being used to store integers, including int8. I added code to use [[LongLong]] on int8.
- + in [[QgsPgQueryBuilder]] the list of available data values would intermittently quote string data when inserted in the query. The mActualFieldsChar member variable was being declared as a local variable in two functions, so it was not set properly when it was tested for character or number field type. Fixed
- + I have replaced in several places SQL query text of the form:
"... =(select oid from pg_class where relname='sometablename' and relnamespace=(select oid from pg_namespace where nspname='someschema'))"
with
...=regclass("someschema"."sometablename")::oid
The ::oid cast is necessary in some situations to force a return of oid rather than text. Also, the names need the double quotes for uppercase use.
(of course the quotes need to be properly escaped in the C string)
This isn't so much a problem, but it makes for cleaner code and is future-proof for when catalog support is available in [[PostgreSQL]].

History

#1 - 2008-03-08 06:57 AM - Steven Mizuno

Attached is my patch that is primarily for fixing internationalization issues on [[PostgreSQL]] databases, but also incorporates a number of other fixes as I have noted.

I have also added a quoteIdentifier() function in [[QgsPostgresProvider]] to provide a more convenient and less error-prone way to properly double quote identifiers in most situations as I suggested in ticket #954.

I believe the only characters that may cause problems with database object names now are the quotes (single and double) and the backslash.

I was doing all of the work on a 64-bit Mandriva Linux system. I have done a cursory check of the fixes on a Windows XP system and did not find any problems related to a different system.

I am also including a script file to create a schema, table and view that I used for testing. The script should have the schema and table placeholders replaced with names containing any characters that someone wishes to test.

Further information on the int2 type detection - I changed the query to get the type name directly instead of in a round about way. It may need more work to support array types.

Additional problems found and fixed:

- addFeature() in [[QgsPostgresProvider]], a the code to escape single quotes was doing all values instead of character types.

- SRCFromViewColumn() and findColumns() in [[QgsPostgresProvider]], changed to use a query similar to what the information_schema views use, but minus the qualification for table owner. This allows VIEWS to be used by any user with select privilege on them. It appears that this had been done previously. fix for bug #955.
- [[QgsDataSourceUri]] - added quoting for database and user values so spaces can be in these items, as well as password, but I didn't provide for escaping certain characters. I feel a better solution for this is needed.
- addAttribute() and deleteAttribute() in [[QgsPostgresProvider]], error handling wasn't detecting failed commands to the database because a test for null bypassed the test for command ok.
- [[QgsPgQueryBuilder]] wouldn't quote text type values as it does for character type. fixed the test to include text type.
- getGeometryColumnInfo() in [[QgsDbSourceSelect]] is not called from anywhere. It seems to be a near duplicate of getTableInfo(). I just enclosed it with #if 0/#endif to keep it from compiling.
- [[QgsDbSourceSelect]]::getTableInfo() - also restricts the return list to tables the user has SELECT privilege on, rather than all tables.

#2 - 2008-03-12 02:05 PM - Jürgen Fischer

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

#3 - 2008-03-20 04:54 AM - Jürgen Fischer

fixed in commit:2ea7615b (SVN r8214)

Files			
QGIS-expected_behavior_-_PostGIS_layers.txt	2.48 KB	2008-02-28	Steven Mizuno
patch_for_bug_964.txt	68.1 KB	2008-03-08	Steven Mizuno
QGIS-PostGIS_test_tables-20080302.sql	7.07 KB	2008-03-08	Steven Mizuno