QGIS Application - Bug report #20614
Problem with encoding when using GRASS (v.generalize) through QGIS

2018-11-23 06:52 PM - Bastien Ferland

Status:	Closed		
Priority:	High		
Assignee:			
Category:	Processing/GRASS		
Affected QGIS version:	3.2.2	**Regression?:**	Yes
Operating System:	Windows	**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	wontfix
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	28434

Description

When using GRASS plugin `v.generalize`, the encoding get messed up. This isn't a big deal and can be manage, but it's still annoying and I think it deserve to be a known issue.

I guess any shape would do, but in my case the shape is from:
http://www12.statcan.gc.ca/census-recensement/2011/geo/bound-limit/files-fichiers/2016/lda_000b16a_e.zip

The process is quite simple:

1) load shape in QGIS: encoding ok
2) use GRASS `v.generalize` with the output saved as Temporary file

When I do so, the shape loaded in 1) as proper encoding (see figure step1 attached), but the one created automatically in 2) doesn't (see figure after_generalize).

After investigation, it seems the problem is that the loading of the new generalize shape set default encoding to UTF8 instead of system. Changing it back to `system` in Properties - Source - Data source encoding fixes the problem. For unaware users, this can be problematic. Would it be possible to have this done by default?

Related issues:

Related to QGIS Application - Bug report # 20556: processing doesn't output a...	**Feedback**	**2018-11-19**

History

#1 - 2018-11-25 08:44 PM - Giovanni Manghi

- Priority changed from Low to High

- Category changed from GRASS to Processing/GRASS

- Status changed from Open to Feedback

Possibly duplicate of #20556 ? This is not an issue on 2.18, right?

#2 - 2018-11-26 02:11 PM - Bastien Ferland

Tried on QGIS 2.18.23 and the encoding is properly manage. It seems it is indeed a QGIS 3.X.X problem.

#3 - 2018-11-26 04:05 PM - Giovanni Manghi

- Regression? changed from No to Yes

- Status changed from Feedback to Open

- Operating System deleted (Windows Server 2016)

#4 - 2019-01-22 01:41 PM - Alexander Bruy

- Related to Bug report #20556: processing doesn't output a shapefile's encoding added

#5 - 2019-02-11 03:38 PM - Luigi Pirelli

- Assignee set to Luigi Pirelli

investigating. First trying to reproduce

#6 - 2019-02-11 03:55 PM - Luigi Pirelli

first rough test on a shape with changed encoding to korean, generate the result in the correct encoding... now checking the attached shape that is originally generated with a no UTF8 encoding. BTW just for not, please use data subsets instead of a big example data.

#7 - 2019-02-11 08:43 PM - Luigi Pirelli

- Operating System set to Windows

confirmed only on windows. System is changed to UTF8

#8 - 2019-02-11 08:47 PM - Luigi Pirelli

- File issue_20614.7z added

Luigi Pirelli wrote:

> *confirmed only on windows. System is changed to UTF8*

attached a cut of portuguese shapefile with system encoding

#9 - 2019-02-11 09:24 PM - Luigi Pirelli

This is not a bug but a feature ;). the standard temporary format in qgis processing is always gpkg, that by default has only allowed UTF-8, UTF-16BE or UTF16LE encoding.
In fact if you select a no-temporary file as shapefile, the output respect the encoding because the format allow you.

IMHO this is not a clear bug. Can be solved adding a validation before to run any grass command that involve temporary file where the original source is not UTF8 or UTF16.

I'll investigate if it is simple to add this check, but probably have to be discussed if have to be implemented or not.

#10 - 2019-02-12 01:05 PM - Luigi Pirelli

I'm not sure what would be a decent solution. A generic use case would collect provider encoding for all inputs, then check if there is an out in gpkg (or some other format with limited encoding) and check if input can be mapped in output encoding.

Because the nature of processing, any check should be related to the nature of the grass command to map correctly if input encoding is propagated to an output layer.
IMHO there is no a generic solution and any generic solution does not consider the nature of the grass command with the risk to create false alerts when not necessary.
The possible workaround is to better document the correct use of the backends. e.g.
If you use temporary file they will be gpkg (if configured) and then UTF8/16 encoded. If your input data use different encoding take care of:
1) do no t use system but prefer explicit and correct encoding
2) take care that this encoding will be changed to UTF8/16 if you use temporary gpkg out files

#11 - 2019-02-12 01:06 PM - Luigi Pirelli

Bastien Ferland what do you mind about my comments?

#12 - 2019-02-12 01:07 PM - Luigi Pirelli

- Assignee deleted (Luigi Pirelli)

I leave the issue in case someone find a clever solution. IMHO I would tag as "Won't fix"

#13 - 2019-02-12 01:23 PM - Luigi Pirelli

- Status changed from Open to Feedback

#14 - 2019-02-12 01:41 PM - Bastien Ferland

I red your comments and now I think I understand the problem. Saving as temporary files will always create a UTF file (because it's a gpkg) meaning that I have two options: 1) convert my source file to UTF before the grass command, or 2) don't use temporary file and save it directly in a specified file. Do I understand properly?

Adding a warning would have been nice, but if it's not possible, I agree that adding a note in the documentation for temporary file is acceptable. That, plus this thread, should allow people to fix the problem quickly when encounter.

Thanks for your time.

#15 - 2019-02-12 07:45 PM - Giovanni Manghi

- Status changed from Feedback to Closed
- Resolution set to wontfix

> *I agree that adding a note in the documentation for temporary file is acceptable.*

ok let's file a ticket against the docs then

Files

after_generalize.PNG	49.3 KB	2018-11-23	Bastien Ferland
step1.PNG	53.2 KB	2018-11-23	Bastien Ferland
issue_20614.7z	521 KB	2019-02-11	Luigi Pirelli