QGIS Application - Bug report #19936
Split Features tool changing the attributes of the new feature on QGIS 3.3

2018-09-25 02:54 AM - Philipe Borba

Status:	Closed		
Priority:	High		
Assignee:	Alessandro Pasotti		
Category:	Digitising		
Affected QGIS version:	3.3(master)	**Regression?:**	Yes
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	Yes	**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	27758

Description

Instead of the original behaviour of the split features tool from previous QGIS (split and both parts getting the same attribute set), split tool creates a new one and does not get the original feature attribute.

The attached gif shows the described behaviour.

I'm running QGIS on MacOS Mojave with the following specs:

QGIS version: 3.3.0-Master
QGIS code revision: 9dd1406539
Compiled against Qt: 5.11.2
Running against Qt: 5.11.2
Compiled against GDAL/OGR: 2.3.1
Running against GDAL/OGR: 2.3.1
Compiled against GEOS: 3.6.3-CAPI-1.10.3
Running against GEOS: 3.6.3-CAPI-1.10.3 80c13047
PostgreSQL Client Version: 10.5
SpatiaLite Version: 4.3.0a
QWT Version: 6.1.3
QScintilla2 Version: 2.10.4
Compiled against PROJ: 520
Running against PROJ: 5.2.0

Associated revisions

Revision e375d1dd - 2018-10-03 10:19 PM - Philipe Borba

[BUG FIX][Split Tool] Bug fix for split features tool

Fixes #19936 by prioritising existing attributes over provider side defaults in some circumstances

History

#1 - 2018-09-25 08:41 AM - Alessandro Pasotti

- Assignee set to Alessandro Pasotti

I cannot reproduce this issue, can you please share a small project and data where the issue can be reproduced?

#2 - 2018-09-25 08:41 AM - Alessandro Pasotti

- Status changed from Open to Feedback

#3 - 2018-09-25 09:48 AM - Giovanni Manghi

- Category changed from Map Tools to Digitising

- Priority changed from Normal to High

Also can't confirm on master/linux.

#4 - 2018-09-25 02:12 PM - Philipe Borba

Giovanni Manghi wrote:

> *Also can't confirm on master/linux.*

I have tested here with data from SpatiaLite and PostGIS and I could only reproduce that bug with postgis data. My database has default attributes.

The SpatiaLite with the data I'm using is available at https://github.com/phborba/dados_artigo_borba_et_al_simgeo2018?files=1 but you have to use the data in a PostgreSQL database. I'll export a dump of my database and will send it to you.

#5 - 2018-09-25 05:34 PM - Philipe Borba

I tested here on linux, the bug happens as well and only on PostGIS database with default values on provider.

#6 - 2018-09-25 05:40 PM - Alessandro Pasotti

Can you check Options -> Data Sources-> Data source handling -> Evaluate default values ?

#7 - 2018-09-25 05:50 PM - Philipe Borba

Alessandro Pasotti wrote:

> *Can you check Options -> Data Sources-> Data source handling -> Evaluate default values ?*

Both linux and macos have the evaluate default values unchecked.

#8 - 2018-09-25 06:40 PM - Giovanni Manghi

Philipe Borba wrote:

> *Giovanni Manghi wrote:*
>
> > *Also can't confirm on master/linux.*
>
> *I have tested here with data from SpatiaLite and PostGIS and I could only reproduce that bug with postgis data. My database has default attributes.*

> *The SpatiaLite with the data I'm using is available at https://github.com/phborba/dados_artigo_borba_et_al_simgeo2018?files=1 but you have to use the data in a PostgreSQL database. I'll export a dump of my database and will send it to you.*

I just tested with PostGIS data and can't confirm. Also your dataset in SL format contains several layers, can you provide us with a minimal project and dataset?
Also the animated gif is not clear to me (is also very small): why do you say that after the split the new feature has different attributes, because the change of symbology? have you checked the table of attributes? Provide also the symbology you are using.

#9 - 2018-09-26 03:10 PM - Philipe Borba

- File debug_test.backup added

- File bug_report.mp4 added

I've done a small database to ilustrate the bug.

In this database there is one table called split_test with fields id (serial primary key), field1 (smallint with default 1) and geom (Multilinestring with EPSG 4326).

The bug is: select one feature with attribute field1 != 1 and use split tool. The desired result would be two features with attribute field1, instead QGIS changes one of the feature's field1 attribute value to 1 (database default).

Inside the database there is a QGIS project with a categorized symbology (red is for field1=1 blue is for else case).

I have also provided a small video ilustrating my text above.

Giovanni Manghi wrote:

> *Philipe Borba wrote:*
>
> > *Giovanni Manghi wrote:*
> >
> > > *Also can't confirm on master/linux.*
> >
> > *I have tested here with data from SpatiaLite and PostGIS and I could only reproduce that bug with postgis data. My database has default attributes.*
> >
> > *The SpatiaLite with the data I'm using is available at https://github.com/phborba/dados_artigo_borba_et_al_simgeo2018?files=1 but you have to use the data in a PostgreSQL database. I'll export a dump of my database and will send it to you.*
>
> *I just tested with PostGIS data and can't confirm. Also your dataset in SL format contains several layers, can you provide us with a minimal project and dataset?*
> *Also the animated gif is not clear to me (is also very small): why do you say that after the split the new feature has different attributes, because the change of symbology? have you checked the table of attributes? Provide also the symbology you are using.*

#10 - 2018-09-26 03:21 PM - Alessandro Pasotti

- Status changed from Feedback to In Progress

Thanks, I was able to reproduce the issue, working on it.

#11 - 2018-09-26 03:26 PM - Philipe Borba

I guess the problem is here:

https://github.com/qgis/QGIS/blob/32ee71634fe96699964fa86dea98273454f977db/src/core/qgsvectorlayereditutils.cpp#L356

When splitFeatures calls QgsVectorLayerUtils::createFeature and passes an attribute map, create feature creates a new feature using provider default but does not overwrite it. On line 398 of QgsVectorLayerUtils::createFeature the v value is overwritten by providerDefault without checking the attribute map. Then, at line 418, the if clause is false and the clause at 420 is not executed (this line would change the value of v)

Maybe a solution would be changing !v.isValid() && attributes.contains(idx) for v.isValid() && attributes.contains(idx) in the if clause of line 418

Philipe Borba wrote:

> *I've done a small database to ilustrate the bug.*
>
> *In this database there is one table called split_test with fields id (serial primary key), field1 (smallint with default 1) and geom (Multilinestring with EPSG 4326).*
>
> *The bug is: select one feature with attribute field1 != 1 and use split tool. The desired result would be two features with attribute field1, instead QGIS changes one of the feature's field1 attribute value to 1 (database default).*
>
> *Inside the database there is a QGIS project with a categorized symbology (red is for field1=1 blue is for else case).*
>
> *I have also provided a small video ilustrating my text above.*
>
> *Giovanni Manghi wrote:*
>
> > *Philipe Borba wrote:*
> >
> > > *Giovanni Manghi wrote:*
> > >
> > > > *Also can't confirm on master/linux.*
> > >
> > > *I have tested here with data from SpatiaLite and PostGIS and I could only reproduce that bug with postgis data. My database has default attributes.*
> > >
> > > *The SpatiaLite with the data I'm using is available at https://github.com/phborba/dados_artigo_borba_et_al_simgeo2018?files=1 but you have to use the data in a PostgreSQL database. I'll export a dump of my database and will send it to you.*
> >
> > *I just tested with PostGIS data and can't confirm. Also your dataset in SL format contains several layers, can you provide us with a minimal project and dataset?*
> > *Also the animated gif is not clear to me (is also very small): why do you say that after the split the new feature has different attributes, because the change of symbology? have you checked the table of attributes? Provide also the symbology you are using.*

#12 - 2018-09-26 03:43 PM - Alessandro Pasotti

I'm afraid this is a won't fix: the current implementation of the new feature creation is

!!! in order of priority !!!

1. client side default expression
   client side default expression set - takes precedence over all. Why? Well, this is the only default
   which QGIS users have control over, so we assume that they're deliberately overriding any
   provider defaults for some good reason and we should respect that
2. provider side default value clause
   note - not an else if deliberately. Users may return null from a default value expression to fallback to provider defaults
3. provider side default literal
   note - deliberately not using else if!
4. passed attribute value

This kind of make sense: if the provider sets a default value, that value is respected any time a new feature is created for example think of a sequence) and split feature does actually create a new feature.

I can't even think about a special case for split: if we have a default we cannot easily know if that's the result of a sequence, a function or anything else, we only know there is default value, and we must honor it.

Btw, feel free to start a discussion on the developer mailing list, perhaps there will be some good ideas for a possible solution.

#13 - 2018-09-26 03:43 PM - Philipe Borba

Tried my suggestion here and it messed up primary constraint evaluation =/

Philipe Borba wrote:

> *I guess the problem is here:*
>
> *https://github.com/qgis/QGIS/blob/32ee71634fe96699964fa86dea98273454f977db/src/core/qgsvectorlayereditutils.cpp#L356*
>
> *When splitFeatures calls QgsVectorLayerUtils::createFeature and passes an attribute map, create feature creates a new feature using provider default but does not overwrite it. On line 398 of QgsVectorLayerUtils::createFeature the v value is overwritten by providerDefault without checking the attribute map. Then, at line 418, the if clause is false and the clause at 420 is not executed (this line would change the value of v)*
>
> *Maybe a solution would be changing !v.isValid() && attributes.contains(idx) for v.isValid() && attributes.contains(idx) in the if clause of line 418*
>
> *Philipe Borba wrote:*
>
> > *I've done a small database to ilustrate the bug.*
> >
> > *In this database there is one table called split_test with fields id (serial primary key), field1 (smallint with default 1) and geom (Multilinestring with EPSG 4326).*
> >
> > *The bug is: select one feature with attribute field1 != 1 and use split tool. The desired result would be two features with attribute field1, instead QGIS changes one of the feature's field1 attribute value to 1 (database default).*
> >
> > *Inside the database there is a QGIS project with a categorized symbology (red is for field1=1 blue is for else case).*
> >
> > *I have also provided a small video ilustrating my text above.*
> >
> > *Giovanni Manghi wrote:*
> >
> > > *Philipe Borba wrote:*

> > > *Giovanni Manghi wrote:*
>
> > *Also can't confirm on master/linux.*
>
> *I have tested here with data from SpatiaLite and PostGIS and I could only reproduce that bug with postgis data. My database has default attributes.*
>
> *The SpatiaLite with the data I'm using is available at https://github.com/phborba/dados_artigo_borba_et_al_simgeo2018?files=1 but you have to use the data in a PostgreSQL database. I'll export a dump of my database and will send it to you.*

I just tested with PostGIS data and can't confirm. Also your dataset in SL format contains several layers, can you provide us with a minimal project and dataset?
Also the animated gif is not clear to me (is also very small): why do you say that after the split the new feature has different attributes, because the change of symbology? have you checked the table of attributes? Provide also the symbology you are using.

#14 - 2018-09-26 03:45 PM - Alessandro Pasotti

Yeah, we crossed comments!

Btw, the issue here is that we cannot blindly override provider defaults with the attribute values coming from the original feature.

#15 - 2018-09-26 03:56 PM - Philipe Borba

What if we tried this approach:

First check if the attribute index is in the primary key index list (QgsVectorLayer has a method to get primary key's fields) and than if the attribute is not in this list, allow to override provider default, because we provided a map.

I guess provider defaults should prevail when there is no map provided, even if the user messed up when digitising, his choice should be considered. The evaluation of the validity of the attribute would be done uppon commit.

Alessandro Pasotti wrote:

> *Yeah, we crossed comments!*
>
> *Btw, the issue here is that we cannot blindly override provider defaults with the attribute values coming from the original feature.*

#16 - 2018-09-26 04:00 PM - Alessandro Pasotti

Yeah, I thought about that too.

But this would break not-PK sequences ... perhaps it's a corner case but still.

The best we can do is make a PR and request for comments.

#17 - 2018-09-26 04:09 PM - Philipe Borba

Changing line 418 with:

if (v.isValid() && attributes.contains(idx) && !layer->primaryKeyAttributes().contains(idx))

did the trick here, but I'm not sure if I broke other stuff.

Would you like me to do a pull request of this change?

Alessandro Pasotti wrote:

> *Yeah, I thought about that too.*
>
> *But this would break not-PK sequences ... perhaps it's a corner case but still.*
>
> *The best we can do is make a PR and request for comments.*

#18 - 2018-09-26 04:13 PM - Alessandro Pasotti

Philipe Borba wrote:

> *Changing line 418 with:*
>
> *if (v.isValid() && attributes.contains(idx) && !layer->primaryKeyAttributes().contains(idx))*
>
> *did the trick here, but I'm not sure if I broke other stuff.*

nothing I can think of (perhaps something in the tests) but Travis will tell you.

> *Would you like me to do a pull request of this change?*

Sure, go ahead!

btw, if accepted, if would be nice to add some test cases for this kind of scenarios.

#19 - 2018-09-26 04:15 PM - Alessandro Pasotti

Just a note: would it be necessary/desireable to check for NULL/empty before overwriting the provider default?

#20 - 2018-09-26 04:46 PM - Philipe Borba

Alessandro Pasotti wrote:

> *Just a note: would it be necessary/desireable to check for NULL/empty before overwriting the provider default?*

On one hand I feel like the tool itself should not change what the user has done (even if he is doing a wrong thing). On the other hand, for other purposes, a check would be nice.

What I've come up with is that if the user has provided a map, this map should be used, even if if results in commit problems. Do you agree?

#21 - 2018-09-26 05:10 PM - Alessandro Pasotti

Philipe Borba wrote:

> *Alessandro Pasotti wrote:*
>
> > *Just a note: would it be necessary/desireable to check for NULL/empty before overwriting the provider default?*
>
> *On one hand I feel like the tool itself should not change what the user has done (even if he is doing a wrong thing). On the other hand, for other purposes, a check would be nice.*
>
> *What I've come up with is that if the user has provided a map, this map should be used, even if if results in commit problems. Do you agree?*

Generally yes but keep in mind that there is not always a map in createFeature.

The split operation needs to create one or more features and in that case we do have a map, I think that we should copy the attributes from the splitted (original) feature only if they are not primary keys.

I'm in doubt about the NULL case when there is a default, I would probably take the NULL over the default, because if in the original feature it was NULL it was probably set to NULL by the user.

Also, please note that QgsVectorLayerUtils::createFeature is widely used across QGIS code, even in processing, and we cannot alter its internal logic completely unless there is a real bug.

#22 - 2018-09-26 05:37 PM - Philipe Borba

Alessandro Pasotti wrote:

> *Philipe Borba wrote:*
>
> > *Alessandro Pasotti wrote:*
> >
> > > *Just a note: would it be necessary/desireable to check for NULL/empty before overwriting the provider default?*
> >
> > *On one hand I feel like the tool itself should not change what the user has done (even if he is doing a wrong thing). On the other hand, for other purposes, a check would be nice.*
> >
> > *What I've come up with is that if the user has provided a map, this map should be used, even if if results in commit problems. Do you agree?*
>
> *Generally yes but keep in mind that there is not always a map in createFeature.*
>
> *The split operation needs to create one or more features and in that case we do have a map, I think that we should copy the attributes from the splitted (original) feature only if they are not primary keys.*
>
> *I'm in doubt about the NULL case when there is a default, I would probably take the NULL over the default, because if in the original feature it was NULL it was probably set to NULL by the user.*
>
> *Also, please note that QgsVectorLayerUtils::createFeature is widely used across QGIS code, even in processing, and we cannot alter its internal*

> *logic completely unless there is a real bug.*

I agree with you, but I guess that this change would not change the behavior of the previous "ifs".

I'm still self-conscious, because this is my first PR in QGIS =]

Do you think there is a better approach? We can change if you advise against this change I'm proposing.

#23 - 2018-09-26 07:06 PM - Alessandro Pasotti

- Operating System deleted (MacOS Mojave)

- Pull Request or Patch supplied changed from No to Yes

PR: https://github.com/qgis/QGIS/pull/8035

#24 - 2018-10-03 10:19 PM - Philipe Borba

- Status changed from In Progress to Closed

- % Done changed from 0 to 100

Applied in changeset commit:qgis|e375d1ddd70f7ee66a415f3beb21a00844f7b044.

Files

Sep-24-2018 21-46-24.gif	1.96 MB	2018-09-25	Philipe Borba
debug_test.backup	14.5 KB	2018-09-26	Philipe Borba
bug_report.mp4	586 KB	2018-09-26	Philipe Borba