

QGIS Application - Bug report #3652
labels buffer: wrong results if data defined setting is used for size

2011-03-21 03:06 AM - Giovanni Manghi

Status:	Closed	
Priority:	Low	
Assignee:	Larry Shaffer	
Category:	Labelling	
Affected QGIS version:	master	Regression?: No
Operating System:		Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: wontfix
Crashes QGIS or corrupts data:	No	Copied to github as #: 13711
Description		
see attached images		

History

#1 - 2011-04-17 02:50 PM - Anne Ghisla

Does it happen with every dataset? Is that maybe related to the font chosen?

#2 - 2011-12-16 01:48 PM - Giovanni Manghi

- Target version changed from Version 1.7.0 to Version 1.7.4

#3 - 2012-02-11 01:49 PM - Giovanni Manghi

- Assignee deleted (nobody -)
- Operating System deleted (All)
- Status info deleted (0)
- Pull Request or Patch supplied set to No
- Crashes QGIS or corrupts data set to No
- Affected QGIS version set to master
- Category changed from Map Legend to Labelling

still true on master, it does not depend on the font or the dataset.

#4 - 2012-03-11 11:21 AM - Martin Dobias

What property is data-defined - font size or buffer size?

Please note that Qt has problems drawing bigger buffers - ~5mm and more. Maybe that is what you wanted to report?

#5 - 2012-03-14 10:54 AM - Giovanni Manghi

- File labels.jpeg added

Martin Dobias wrote:

| What property is data-defined - font size or buffer size?

buffer size

Please note that Qt has problems drawing bigger buffers - ~5mm and more. Maybe that is what you wanted to report?

see the attached image:

with both label engines at 4mm buffer size (with a 14pt font size) the buffer looks already pretty bad, with the old engine (red) a little better than the new one (orange)

If this is how it works then as user I would probably expect another thing: up to 2/3 mm buffer size see the buffer follow the font, after that just see a rectangle (maybe with an option to round corners).

#6 - 2012-04-16 06:26 AM - Paolo Cavallini

- Target version changed from Version 1.7.4 to Version 1.8.0

#7 - 2012-08-22 09:08 PM - Larry Shaffer

As of commit `commit:7079f20` there is an additional option (layer-only for now) for defining the pen join style. Try the 'Round' join style to see if it helps. It will no overcome the Qt limitations of larger buffers, but can make some look better and reduce artifacts. The default join style was (is) 'Bevel'.

#8 - 2012-08-23 03:11 AM - Giovanni Manghi

Larry Shaffer wrote:

As of commit `commit:7079f20` there is an additional option (layer-only for now) for defining the pen join style. Try the 'Round' join style to see if it helps. It will no overcome the Qt limitations of larger buffers, but can make some look better and reduce artifacts. The default join style was (is) 'Bevel'.

Hi Larry, thanks a lot for your effort. I will try the option asap. What about my above suggestion (above a certain size render the buffer as a rectangle with eventually an option to round corners)?

#9 - 2012-08-23 12:45 PM - Larry Shaffer

Giovanni Manghi wrote:

... What about my above suggestion (above a certain size render the buffer as a rectangle with eventually an option to round corners)?

Curious as to why you need such large buffers?

Tim Sutton started work on rounded rectangles with his [Shield Labels](#) separate branch. I think another approach might be to place SVG symbols behind the text, with options to auto-resize to label size (plus a buffer distance), orient to label rotation (or not) and symbol placement relative to the label itself (center, rotation point, etc.). Having a data-defined field for scaling the symbol would be good. Colorizing the symbol would also be great.

#10 - 2012-08-24 02:35 AM - Giovanni Manghi

Larry Shaffer wrote:

Giovanni Manghi wrote:

... What about my above suggestion (above a certain size render the buffer as a rectangle with eventually an option to round corners)?

Curious as to why you need such large buffers?

I don't :)

I was just saying that if the buffers above a certain size are unusable then instead it would be better to render them as rectangles. But obviously if there are more elegant solutions in the work then just ignore my idea. Cheers!

#11 - 2012-09-04 12:00 PM - Paolo Cavallini

- Target version changed from Version 1.8.0 to Version 2.0.0

#12 - 2012-09-16 01:47 AM - Giovanni Manghi

- Assignee set to Larry Shaffer

- Status changed from Open to Feedback

Larry Shaffer wrote:

Giovanni Manghi wrote:

... What about my above suggestion (above a certain size render the buffer as a rectangle with eventually an option to round corners)?

Curious as to why you need such large buffers?

I made a few tests and to me it does not make a lot of sense to allow the user choose a buffer bigger than a certain size, because it produces anyway unexpected results. With a 14 points font the buffer can be ok up to 8mm with the "round" pen join style, but with other pen join styles it is ok just with much smaller buffers (2/3mm).

When the label includes letter as "O" or "B" there is also the issue of the buffer have empty parts, even with the "color inside of pen stroke" option selected (if this option was added to avoid this issue).

In lack of a better solution (regular shaped buffers) I guess we can just add a warning or even do not allow buffers of any size, by limiting the upper limit of the spin box.

#13 - 2012-09-16 03:25 PM - Larry Shaffer

Giovanni Manghi wrote:

I made a few tests and to me it does not make a lot of sense to allow the user choose a buffer bigger than a certain size, because it produces anyway unexpected results. With a 14 points font the buffer can be ok up to 8mm with the "round" pen join style, but with other pen join styles it is ok just with much smaller buffers (2/3mm).

Some of this also depends upon the font style used. There are many settings in QGIS that can give completely crazy results (not as a flaw) because the parameters just don't work. Experimentation, and finding the right 'look' will take some trial and error (hopefully just using the label's sample/preview).

When the label includes letter as "O" or "B" there is also the issue of the buffer have empty parts, even with the "color inside of pen stroke" option selected (if this option was added to avoid this issue).

Current label creation method doesn't provide an easy option for filling those character holes (which I'm assuming you want to do). Similarly, that's usually a manual, multi-step process in high-end illustration programs.

I included that buffer option because when buffers are drawn, they are filled with their color at about 50% transparency. If you set the text to also be partially transparent, the buffer's fill will interact and give mixed color transparency results. Turning off the buffer fill fixes that issue (except where the interior aspect of the buffer's stroke intersects with the text's fill) and also allows the user to make outlined text. Outlined text is when the text transparency is 100%, the buffer fill is not shown, and all that remains is the buffer's stroked outline of the perimeter of the text. Add a little buffer transparency and letter spacing and you have a nice subdued-but-readable look for very large background labels.

In lack of a better solution (regular shaped buffers) I guess we can just add a warning or even do not allow buffers of any size, by limiting the upper limit of the spin box.

Currently, the sample/preview label is fairly accurate. I think sample should be enough of a 'warning' for the user that the operation they are about to do may turn out looking crazy. Limiting the spinboxes might be an answer, but not sure how to programmatically 'see' when the buffer is wrong. Also recently added was buffers in map units; so there really shouldn't be any fixed limiting to the spinboxes. One solution might be to dynamically set the spinbox limit to a percentage of buffer size compared to text size, augmented by other factors (join style, etc). All sounds like a bunch of work, given the sample already provides user-feedback.

In other words, I think maybe it should stay as is, and a different approach to creating alternative label backgrounds that dynamically adapt to the text (e.g. SVG) should be implemented.

#14 - 2012-09-16 11:57 PM - Giovanni Manghi

In other words, I think maybe it should stay as is, and a different approach to creating alternative label backgrounds that dynamically adapt to the text (e.g. SVG) should be implemented.

ok Larry I agree, then I think we should reject this ticket.

#15 - 2012-10-06 05:55 AM - Giovanni Manghi

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

Files

Screenshot.png	213 KB	2011-03-21	Giovanni Manghi
buffer_label.png	163 KB	2011-03-21	Giovanni Manghi
labels.jpeg	101 KB	2012-03-14	Giovanni Manghi