

QGIS Application - Bug report #5749

transparent polygons not rendered

2012-06-06 05:53 AM - Bernhard Ströbl

Status: Closed	
Priority: Severe/Regression	
Assignee: Marco Hugentobler	
Category: QGIS Server	
Affected QGIS version: master	Regression?: No
Operating System: OpenSUSE 64 bit	Easy fix?: No
Pull Request or Patch supplied: No	Resolution:
Crashes QGIS or corrupts data: No	Copied to github as #: 15255
Description	
Polygons with transparency > 0% and < 100% are not rendered example project contains data from OSM project (tiny subset of roads and landuse). Layer landuse is included in the project twice: transparency = 0% and transparency = 32%. regression from 1.7.4	

Associated revisions

Revision ee2f1bdb - 2012-06-08 06:50 PM - Jürgen Fischer

Use QString::number with QDomElement::setAttribute with double/float (fixes #5749)

History

#1 - 2012-06-06 08:28 AM - Marco Hugentobler

- Assignee set to Marco Hugentobler

#2 - 2012-06-06 02:33 PM - Werner Macho

not confirmed here - latest master self compiled - testfiles are displayed transparently with every value from 0 - 100

#3 - 2012-06-06 02:40 PM - Larry Shaffer

- File 5749_8b.png added

Not confirmed with Mac OS X 10.6.8 (1.8_c695198). Transparency works as expected with test data.

#4 - 2012-06-06 02:49 PM - Tim Sutton

Works for me on desktop. He has set the category to mapserver so I wonder if it fails only when publishing the layer via wms.

#5 - 2012-06-06 11:07 PM - Marco Hugentobler

The problem is that the double alpha value in the project file has a comma instead of a dot:

<symbols>

<symbol outputUnit="MM" alpha="0,6784313725490196" type="fill" name="0">

And it should be:

```
<symbols>
  <symbol outputUnit="MM" alpha="0.6784313725490196" type="fill" name="0">
```

With that modification, it works here.

#6 - 2012-06-07 12:03 AM - Bernhard Ströbl

Marco Hugentobler wrote:

The problem is that the double alpha value in the project file has a comma instead of a dot:

```
<symbols>
<symbol outputUnit="MM" alpha="0,6784313725490196" type="fill" name="0">
```

And it should be:

```
<symbols>
<symbol outputUnit="MM" alpha="0.6784313725490196" type="fill" name="0">
```

With that modification, it works here.

ok, I checked here: QGIS 1.7.4 writes a dot (alpha="0.6784313725490196") whereas 1.8 writes a comma (alpha="0,6784313725490196") this maybe because it is a German system and we use a comma instead of dot for decimals and yes, only QGIS Server is affected

#7 - 2012-06-07 12:22 AM - Giovanni Manghi

this #2104 seems also related.

#8 - 2012-06-07 12:27 AM - Marco Hugentobler

and yes, only QGIS Server is affected

I have the same behaviour (alpha with comma does not work) in the desktop application. Probably it depends on your settings or locale (though mine is de_CH.UTF-8).

#9 - 2012-06-07 12:43 AM - Jürgen Fischer

Marco Hugentobler wrote:

and yes, only QGIS Server is affected

I have the same behaviour (alpha with comma does not work) in the desktop application. Probably it depends on your settings or locale (though mine is de_CH.UTF-8).

are desktop and server running in different environments?

Why are we using `QDomElement::setAttribute (const QString & name, const QString & value)` with `QString::number` all over the place, when we just could use the version for double and/or int.

Like in <https://gist.github.com/2887218> (compiles, but not tested).

#10 - 2012-06-07 01:27 AM - Bernhard Ströbl

Marco Hugentobler wrote:

and yes, only QGIS Server is affected

I have the same behaviour (alpha with comma does not work) in the desktop application. Probably it depends on your settings or locale (though mine is de_CH.UTF-8).

mine is de_DE (also UTF-8)

#11 - 2012-06-07 11:51 PM - Marco Hugentobler

Why are we using `QDomElement::setAttribute (const QString & name, const QString & value)` with `QString::number` all over the place, when we just could use the version for >double and/or int.

Because the developer was not aware there is a formatting difference between these two.

But looking at the Qt documentation, I think it is better if we format numbers in the project file not according to the locale. Imagine in an international company, a person with locale de_xx prepares a project for his boss (with locale en_xx). Or someone publishes a project from the desktop to the server. So `QString::number()` without locale aware formatting seems better to me.

#12 - 2012-06-08 01:31 AM - Jürgen Fischer

Marco Hugentobler wrote:

Because the developer was not aware there is a formatting difference between these two.

There isn't any, is it? The documentation clearly says that those are localized too.

So which dumb developer suggested to use `QDomElement::setAttribute()` for doubles? ;)

#13 - 2012-06-08 01:41 AM - Marco Hugentobler

There isn't any, is it? The documentation clearly says that those are localized too.

I'm a bit confused. I interpreted it the way they are not localized (but I did not really test it):

"Unlike `QLocale::toString()`, this function does not honor the user's locale settings" (<http://qt-project.org/doc/qt-4.8/qstring.html#number-2>)

#14 - 2012-06-08 04:16 AM - Jürgen Fischer

Marco Hugentobler wrote:

```
| There isn't any, is it? The documentation clearly says that those are localized too.  
|  
| I'm a bit confused. I interpreted it the way they are not localized (but I did not really test it):
```

Sorry.

My assumption was, that localization of XML attributes doesn't make any sense and therefore I (dumb me) suggested the double version of `QDomElement::setAttribute`. Obviously that's wrong, because its documentation (which I didn't read before) clearly says that the output *is* localized.

```
| "Unlike QLocale::toString(), this function does not honor the user's locale settings" (http://qt-project.org/doc/qt-4.8/qstring.html#number-2)
```

Yes - of course. Ignore everything I said.

Looks like that's the actual fix:

```
diff --git a/src/core/symbology-ng/qgssymbollayerv2utils.cpp b/src/core/symbology-ng/qgssymbollayerv2utils.cpp  
index 89074e4..38d1ad3 100644  
--- a/src/core/symbology-ng/qgssymbollayerv2utils.cpp  
+++ b/src/core/symbology-ng/qgssymbollayerv2utils.cpp  
@@ -713,7 +713,7 @@ QDomElement QgsSymbolLayerV2Utils::saveSymbol( QString name, QgsSymbolV2* symbol  
    symEl.setAttribute( "type", _nameForSymbolType( symbol->type() ) );  
    symEl.setAttribute( "name", name );  
    symEl.setAttribute( "outputUnit", encodeOutputUnit( symbol->outputUnit() ) );  
- symEl.setAttribute( "alpha", symbol->alpha() );  
+ symEl.setAttribute( "alpha", QString::number( symbol->alpha() ) );  
    QgsDebugMsg( "num layers " + QString::number( symbol->symbolLayerCount() ) );  
    for ( int i = 0; i < symbol->symbolLayerCount(); i++ )  
    {
```

#15 - 2012-06-08 05:52 AM - Jürgen Fischer

Jürgen Fischer wrote:

```
| Looks like that's the actual fix:  
|  
| [...]
```

Um, but that's just one place. I declared the two Qt methods deprecated in `qdom.h`:

```
Q_DECL_DEPRECATED void setAttribute(const QString& name, float value);  
Q_DECL_DEPRECATED void setAttribute(const QString& name, double value);
```

And <https://gist.github.com/2895431> is the rather large patch that is needed to fix all the deprecation warnings:

```
src/core/composer/qgscomposerarrow.cpp      | 12 +++++-----
src/core/composer/qgscomposeritem.cpp      | 12 +++++-----
src/core/composer/qgscomposerlegend.cpp    | 4 +-
src/core/composer/qgscomposermap.cpp       | 22 ++++++++-----
src/core/composer/qgscomposerpicture.cpp   | 4 +-
src/core/composer/qgscomposerscalebar.cpp  | 14 +++++-----
src/core/composer/qgscomposershape.cpp     | 6 +---
src/core/composer/qgscomposertable.cpp     | 4 +-
src/core/composer/qgscomposition.cpp       | 10 +++++-----
src/core/qgsdiagramrendererv2.cpp          | 24 ++++++++-----
src/core/qgslabel.cpp                       | 10 +++++-----
src/core/qgsmaplayer.cpp                    | 8 +++++
src/core/qgsvectorlayer.cpp                 | 4 +-
.../symbolology-ng/qgsgraduatedsymbolrendererv2.cpp | 4 +-
.../symbolology-ng/qgspointdisplacementrendererv2.cpp | 6 +---
src/core/symbolology-ng/qgssymbolayerv2utils.cpp | 2 +-
src/gui/qgsannotationitem.cpp               | 18 ++++++++-----
src/mapserver/qgswmsserver.cpp             | 16 +++++-----
.../qgsrasterterrainanalysisdialog.cpp     | 4 +-
19 files changed, 92 insertions(+), 92 deletions(-)
```

Still untested. I wonder why nobody noticed so far - I suppose anyone using a localization with a comma instead of a dot should have suffered from truncation of decimal places, which using `QDomElement::setAttribute(...,somedouble)` to write and `QDomElement::attribute(...).toDouble()` to read should produce.

#16 - 2012-06-08 07:45 AM - Marco Hugentobler

Thanks Jürgen, please apply the patch.

#17 - 2012-06-08 09:36 AM - Jürgen Fischer

Marco Hugentobler wrote:

| *Thanks Jürgen, please apply the patch.*

The large one or the small one? ;)

#18 - 2012-06-08 09:48 AM - Marco Hugentobler

The large one

#19 - 2012-06-08 09:51 AM - Jürgen Fischer

- Status changed from Open to Closed

Fixed in changeset commit:"ee2f1bdbaf20c37bb8ba5026b5de73f58a0b6ad0".

Files

wms_transparency.zip	45.6 KB	2012-06-06	Bernhard Ströbl
5749_8b.png	120 KB	2012-06-06	Larry Shaffer