

QGIS Application - Feature request #6325

Add PAL labelling features for Python access

2012-09-10 03:23 AM - Ian Packham

Status: Closed	
Priority: Normal	
Assignee:	
Category: Labelling	
Pull Request or Patch Supplied:	Resolution: worksforme
Easy fix?: No	Copied to github as #: 15612
Description	
<p>The new PAL Labelling system looks to be superseding the old one (now shown as deprecated under Layer Properties). Could the public (accessed via GUI) features please be made available for Python programming.</p> <p>I want to programatically set labels and other attributes from the Python code so the user gets a nice looking layer without needing to change the layer properties themselves. Being able to set features available under the new labelling engine (also available on the Toolbar would be useful), such as (not) "Label every part of a multi-part feature" and avoiding label collisions. Also the look of the labels generated using the latter method are usually better. If this is to become the default labelling system, then it would be good to be able to recreate the look/feel via Python.</p>	

History

#1 - 2012-09-10 03:33 AM - Ian Packham

Sorry, I have created this issue under the wrong category - should be under "Labelling". I cannot change the category. Please amend or, if you prefer, delete this issue and I will recreate it in the right place.

#2 - 2012-09-10 03:38 AM - Paolo Cavallini

- Project changed from 27 to QGIS Application

#3 - 2012-09-10 03:39 AM - Paolo Cavallini

- Category set to Labelling

#4 - 2012-09-10 08:11 PM - Larry Shaffer

See also, for background: <http://gis.stackexchange.com/questions/32847>

#5 - 2012-09-16 04:00 PM - Larry Shaffer

- Resolution set to worksforme

- Status changed from Open to Feedback

- Target version set to Version 2.0.0

Ian,

Jürgen added the Python bindings with commit commit:d8675ba3 (and follow up commit)

Please test. I have tested with setting PAL layer options (followed by canvas refresh) and seems to work well.

#6 - 2012-09-18 05:30 AM - Ian Packham

That was fast, thank you.

Would this commit be included in the qgis-dev nightly builds of the master (available in the OsGeo4W-setup for Windows Installer)? I have installed the latest (v1.9.0-66), but when I go to Python Console, I get this error:

```
Failed to open Python console:  
exceptions.ImportError: No module named Qsci
```

I also added the qscintilla package under Libs, but this didn't help.

I am a newbie at testing code from dev. From the commit link I can't work out which methods etc are now available in Python. Is there somewhere else I should look as well for documentation?

#7 - 2012-09-18 05:40 AM - Jürgen Fischer

Ian Packham wrote:

I also added the qscintilla package under Libs, but this didn't help.

Try python-qscintilla.

From the commit link I can't work out which methods etc are now available in Python.

I missed the new file in the first commit - that's why Larry mentioned the follow up commit (commit:8898e6b847) which has qgspallabeling.sip.

#8 - 2012-09-18 09:06 AM - Ian Packham

Sorry, I was looking for it, but I missed your missed commit file. Got the build working OK with python-qscintilla.

Managed to set some attributes of PAL from python, although not tested everything. Setting `palyr.labelPerPart = True` did not change the labels in a polygon layer, but I also cannot change it via the PalLabelling toolbar and clicking the checkbox. so maybe something else is missing in the underlying code?

I like that this setting is labelPerPart feature is False by default, by the way; in fact I probably don't want to change much for these settings, just being able to switch this PalLabelling on programmatically is very useful.

#9 - 2012-09-18 11:07 AM - Larry Shaffer

Ian Packham wrote:

Managed to set some attributes of PAL from python, although not tested everything. Setting `palyr.labelPerPart = True` did not change the labels in a polygon layer, but I also cannot change it via the PalLabelling toolbar and clicking the checkbox. so maybe something else is missing in the underlying code?

The **labelPerPart** property refers to features that have multiple parts; so, you wouldn't see any changes to your labeling if the features of the layer (that are valid, visible and being labeled) do not have multiple parts. It's fairly common to have line layers, e.g. roads, with multiple parts per feature.

#10 - 2012-09-19 02:20 AM - Ian Packham

Larry

The layer I was testing is a MultiPolygon, some of the features have multiple parts, it is a postgres layer and ST_IsValid returns true for the features in question. Loading the same layer in v1.8.0 and checking the "Label every part of multi-part features" checkbox, I get the same label on more than 1 part of the feature. Using the same layer in the dev build and checking the box in the same way, does not change the labelling.

#11 - 2012-09-19 09:19 AM - Larry Shaffer

Hi Ian,

Sounds like a bug that may have been caused by some recent code I've added. Please create a new issue (bug) ticket and assign it to me. If you can, provide a sample of your data set that includes single and multi-part features. Also describe which placement options you were using. I suspect the multi-part option does work, but not for the Offset and Around placements. I will look into this ASAP.

Thank you for reporting the issue.

If you have tested the new Python bindings, and they seem to work well, consider closing this issue, if you have the permission to do so; or, I can.

#12 - 2012-09-20 02:45 AM - Ian Packham

New issue created: #6378. I don't have permissions to close an issue - yet?

#13 - 2012-09-20 10:50 AM - Larry Shaffer

- *Status changed from Feedback to Closed*