

QGIS Application - Bug report #3170

many GDAL Tools don't work in OS X standalone

2010-10-31 09:32 AM - William Kyngesburye

Status:	Closed	
Priority:	Low	
Assignee:	Giuseppe Sucameli	
Category:	GDAL Tools	
Affected QGIS version:		Regression?: No
Operating System:	OS X	Easy fix?: No
Pull Request or Patch supplied:	No	Resolution: fixed
Crashes QGIS or corrupts data:		Copied to github as #: 13230
Description		
<p>On OS X standalone Qgis, the GDAL python extension is in the Qgis python folder in the application, but this is not passed on to any of the GDAL python tools when run (they don't use the qgis builtin python interpreter).</p> <p>Either [[GdalTools]] needs to automatically do this (or Qgis needs to do this), or the disable "GDAL pymod path" option in the [[GdalTools]] settings dialog needs to be enabled and programmed so the user can set the path.</p> <p>Or Qgis needs to provide a general environment settings for common variables - see #3097.</p>		

Associated revisions

Revision cd7984e1 - 2011-12-09 05:40 PM - Giuseppe Sucameli

[BACKPORT] Enable path to pymods in GdalTools settings dialog and automatically set default OSX environment (fix #3170)

Revision 6dc7deed - 2011-12-10 12:00 PM - Giuseppe Sucameli

[BACKPORT] Enable path to pymods in GdalTools settings dialog and automatically set default OSX environment (fix #3170)

History

#1 - 2011-03-05 04:43 AM - Borys Jurgiel

Isn't there any way to set a fixed path on the build level?

#2 - 2011-04-16 12:15 PM - William Kyngesburye

Not a blocker for general release, just for standalone.

As mentioned, there is a [[GdalTools]] setting to set the pymod path, but it is disabled (it doesn't look like the feature is programmed yet). This is really all that is needed for those cases where GDAL is not where expected, like bundled in a standalone package. No need to hardwire the path at compile time.

#3 - 2011-04-17 04:06 AM - Giuseppe Sucameli

python path setting is disabled because not implemented yet. I think would be better to hide it.

However, you can use GDAL binaries setting instead of python path setting. Maybe are there some issues using it?

#4 - 2011-04-23 09:41 AM - William Kyngesburye

I didn't know the GDAL *executable* path is also used for the Python *module* path. Doesn't make sense, they're different things.

Another possibility - add the QGIS python path to the environment PYTHONPATH before running external python scripts. Even better would be to add the QGIS bin folder to the environment PATH before running external programs. This takes care of the standalone case, AND the user doesn't need to fuss with configuring the GDAL path (the standalone should be zero-config so the user can start using QGIS immediately).

#5 - 2011-12-06 04:29 PM - Giuseppe Sucameli

- Pull Request or Patch supplied set to No

Now this problem seems clear to me.

William Kyngesburye wrote:

| I didn't know the GDAL *executable* path is also used for the Python *module* path. Doesn't make sense, they're different things.

It make sense instead: GDAL python modules are called as executables, not just imported by code. For this reason everything works fine setting the path to GDAL executable.

This value will be added to the PATH env variable, so calling both binary and python executables using QProcess works because it's able to find the file.

| Another possibility - add the QGIS python path to the environment PYTHONPATH before running external python scripts.

This may only be useful if the gdal.py module (or ogr.py or whatever module is imported by code) is not found, but without gdal.py the GdalTools plugin won't be loaded at all.

Anyway, I didn't remember any case of error due to a missing gdal.py module, so I think we could remove the "GDAL pymod path" setting to avoid confusion.

| Even better would be to add the QGIS bin folder to the environment PATH before running external programs.

The problem is that in MacOS the GDAL binaries are not in the QGIS bin folder, they are in the GDAL framework directory.

In Windows instead GDAL executables, both gdal*.exe and gdal*.bat (with related gdal*.py) are in the QGIS bin folder. In addition, the user runs qgis.bat (not qgis.exe) which sets the needed environment variables up (e.g. PATH).

| This takes care of the standalone case, AND the user doesn't need to fuss with configuring the GDAL path (the standalone should be zero-config so the user can start using QGIS immediately).

The standalone in windows works fine, as I wrote it adds QGIS bin folder to PATH before running QGIS. I don't know if there's a way to do the same on MacOS.

In this moment MacOS users have to set the path to GDAL executables setting to something like /Library/Frameworks/GDAL.framework/Versions/1.8/Programs. I added a tooltip to help them.

IMHO setting this value by default on MacOS systems could be a good workaround, but I don't know if the MacOS installer puts GDAL always in that path (except the GDAL version number, of course, I can retrieve it by code).

William, what's your opinion?

#6 - 2011-12-06 06:35 PM - William Kyngesburye

I didn't know the GDAL executable path is also used for the Python module path. Doesn't make sense, they're different things.

It make sense instead: GDAL python modules are called as executables, not just imported by code. For this reason everything works fine setting the path to GDAL executable.

This value will be added to the PATH env variable, so calling both binary and python executables using QProcess works because it's able to find the file.

Well, GDAL (any, really) python modules (gdal.py, ogr.py, ...) are not the same as the GDAL python executables (the tools, ie gdal2tiles.py, ...). Python modules are only imported, executing them is meaningless. (I suppose modules can be written to be executable also, or the other way around, but not the GDAL modules.) They are in different locations (in the basic *nix case) - bin/ for the executable tools and the system, user or other (ie Qgis app python folder) site-packages for the modules.

The problem is that in MacOS the GDAL binaries are not in the Qgis bin folder, they are in the GDAL framework directory.

In Windows instead GDAL executables, both gdal.exe and gdal*.bat (with related gdal*.py) are in the QGis bin folder. In addition, the user runs qgis.bat (not qgis.exe) which sets the needed environment variables up (e.g. PATH).*

Actually, the GDAL executables (both compiled and python) **are** in the Qgis bin/ folder in the OS X standalone. But the GDAL modules are in the Qgis python folder.

The standalone in windows works fine, as I wrote it adds QGis bin folder to PATH before running QGis. I don't know if there's a way to do the same on MacOS.

I don't think this is a problem, as long as it sets/adds to the environment PATH.

In this moment MacOS users have to set the path to GDAL executables setting to something like /Library/Frameworks/GDAL.framework/Versions/1.8/Programs. I added a tooltip to help them.

IMHO setting this value by default on MacOS systems could be a good workaround, but I don't know if the MacOS installer puts GDAL always in that path (except the GDAL version number, of course, I can retrieve it by code).

That path is standard, for the "official" GDAL build framework (though there's another unofficial GDAL framework out there that does things a little different), and as you say you should try to get the GDAL version at runtime.

But this still does not fix the standalone problem, the subject of this bug. If you are already adding the Qgis bin/ to the PATH before executing the GDAL tools, that's one part of the solution. The other part is that those executed tools still don't know about the Qgis python module folder to be able to find any bundled modules (putting them in the bin/ folder is just a hack).

#7 - 2011-12-06 07:26 PM - Giuseppe Sucameli

William Kyngesburye wrote:

*the GDAL executables (both compiled and python) **are** in the Qgis bin/ folder in the OS X standalone.*

Ok, I misunderstood the subject, so the QGis application I tried is not the standalone one even if I thought it was.

I've found it now in the archive section of your site. Is 1.5.0-3 the latest QGIS standalone version for MacOS, isn't it?

#8 - 2011-12-06 08:27 PM - William Kyngesburye

Correct. That's the version I realized that it was pointless to package a standalone if part of QGIS didn't function, and reported the GdalTools bug.

#9 - 2011-12-09 07:46 AM - Giuseppe Sucameli

Just a summary,

`QgsApplication.prefixPath()` (named `qgis_prefix`) points to the folder containing the QGIS executable (`Qgis.app/MacOS`):

if QGIS standalone I need to set `PATH` to `qgis_prefix/bin` and `PYTHONPATH` to `qgis_prefix/../Resources/python`

otherwise (non-standalone version) I set `PATH` to the default GDAL.framework path (`/Library/Frameworks/...`).

Making a such change and then running `gdal_merge.py` (Raster->Merge) from QGIS 1.5.0-3 standalone it loads the `gdal.py` module (I get no `ImportError`).

Did I miss anything else?

To understand if it's a standalone version I check if the `GDAL.framework` folder exists within the `qgis_prefix/../Frameworks`.

Is this the right way or maybe is there another way to do it?

If you agree with those changes, I will commit them to master and backport them to 1.7 branch as well (so you can start again to create working standalone packages for next releases), but I will also make available a new version of GdalTools to the Faunalia repo to make it available for older QGIS standalone versions.

#10 - 2011-12-09 08:11 AM - William Kyngesburye

Sounds OK.

thanks

#11 - 2011-12-09 08:43 AM - Giuseppe Sucameli

- *Resolution set to fixed*

- *Status changed from Open to Closed*

Fixed in commit `cd7984e128645851606b862309414930ea74da5c`