Virtual layers not working in Processing

for instance:

GDAL command:

gdal_rasterize -a attr -ot Float32 -tr 100 100 -a_nodata -9999 -co
COMPRESS=DEFLATE -co PREDICTOR=1 -co ZLEVEL=6 -l
point?crs=EPSG:3003&memoryid={8e06ae07-51d0-415f-81e2-2565db34cace}
point?crs=EPSG:3003&memoryid={8e06ae07-51d0-415f-81e2-2565db34cace}
/tmp/processing543c8fb958c34d0a8b161fc7140b98d0/33d7bfb8c05c4d98a60d365fdebcdcf4/OUTPUT.tif

GDAL command output:

/bin/sh: 1: 
/tmp/processing543c8fb958c34d0a8b161fc7140b98d0/33d7bfb8c05c4d98a60d365fdebcdcf4/OUTPUT.tif: not found
/bin/sh: 1: point?crs=EPSG:3003: not found

Missing source or destination.

They should preferably be enabled for analyses, as they are very convenient for on the fly analyses. If this is not yet possible, they should not appear on the dropdown.

Associated revisions

Revision f9d69051 - 2016-02-25 10:37 PM - Jürgen Fischer

new spatialite layer: allow tables with only primary key and geometry (fixes #14313)

History

#1 - 2016-02-17 12:29 AM - Hugo Mercier

Hi.

Apparently you are using a memory layer as input, right ? But the result would be similar with a virtual layer.

Actually with anything that is not loaded by ogr.

The rasterize algorithm expects a vector layer opened with ogr, but the dropdown displays any vector layer, whatever the provider.

Algorithms should somehow declare which types of vector layers they can read. Just an idea: an optional parameter to ParameterVector to list the acceptable provider keys ? Victor what do you think ?
Algorithms should somehow declare which types of vector layers they can read. Just an idea: an optional parameter to ParameterVector to list the acceptable provider keys? Victor what do you think?

This is already in TODO, see #11616

#3 - 2016-02-25 01:38 PM - Jürgen Fischer
- Status changed from Open to Closed

Fixed in changeset commit:"f9d6905166ec7a43bedce4babcb80f7115f0478337".