QGIS Application - Bug report #12479
gdal 2.0.0

2015-03-31 02:56 AM - Mark Johnson

Status:	Closed		
Priority:	Normal		
Assignee:	Nyall Dawson		
Category:	Data Provider/OGR		
Affected QGIS version:	2.8.1	**Regression?:**	No
Operating System:		**Easy fix?:**	No
Pull Request or Patch supplied:	No	**Resolution:**	
Crashes QGIS or corrupts data:	No	**Copied to github as #:**	20627

Description

I have been working with the gdal 2.0.0 development code for a while and have noticed that due to the implementation of gdal RFC 41, code changes will be needed so that qgis will react in the same way with bot gdal 1.* and 2.*.

All of the changes are in
src/providers/ogr/qgsogrprovider.cpp

in the functions
- QgsOgrProvider::subLayers()
- QgsOgrProvider::getOgrGeomType

Also there is a problem in
- QgsOgrProvider::setSubsetString
where the needed dialect parameter is set to NULL instead of "OGRSQL"

All 3 cases effect only table that have MORE than 1 geometry.

Up to gdal 1.11.2, each geometry field is treated as 1 layer.
Starting with gdal 2.0 each table is treated a 1 layer and the geometry fields of that table must be retrieved.

For QgsOgrProvider::getOgrGeomType only one line must be changed:
from:

```
    geomType = OGR_FD_GetGeomType( fdef );
```

to:

```
    geomType = OGR_GFld_GetType(OGR_FD_GetGeomFieldDefn(fdef, 0));
```

For subLayers() it is a bit more comlecated, since now 2 loops are needed
1) Layer
2) fields of layers

```
      if ( !mSubLayerList.isEmpty() )
        return mSubLayerList;
      int layer_number=0; // depending on the gdal-version being used, the final result may be different that the result of layerCount()
      int layer_count=layerCount();
      for ( int i = 0; i < layer_count ; i++ )
      {
        OGRLayerH layer = OGR_DS_GetLayer( ogrDataSource, i );
```

```
    OGRFeatureDefnH fdef = OGR_L_GetLayerDefn( layer );
    QString theLayerName = FROM8( OGR_FD_GetName( fdef ) );
    int fieldCount=OGR_FD_GetGeomFieldCount(fdef);
    for(int j=0; j<fieldCount; j++ )
    {
     QString theLayerFieldName = FROM8(OGR_GFld_GetNameRef(OGR_FD_GetGeomFieldDefn(fdef,j)));
     OGRwkbGeometryType layerGeomType=OGR_GFld_GetType(OGR_FD_GetGeomFieldDefn(fdef,j))
     QgsDebugMsg( QString( "layerGeomType = %1" ).arg( layerGeomType ) );
     if ( layerGeomType != wkbUnknown )
    { // gdal up to version 1.11.2 will return a layer-name using the ogr-format 'table_name(field_name)'. gdal starting with 2.0.0 will
only return the table_name
      int theLayerFeatureCount = OGR_L_GetFeatureCount( layer, j );
      QString geom = ogrWkbGeometryTypeName( layerGeomType );
      // QgsMessageLog::logMessage( tr( "subLayers(%1) layer[%2 - %3/%4] OGR_FD_GetName[%5] field_count[%6/%7]
OGR_GFld_GetNameRef[%8] layerGeomType[%9] feature_count[%10]" ).arg(GDALVersionInfo( "RELEASE_NAME"
)).arg(layer_number).arg(i).arg(layer_count).arg( theLayerName).arg(j).arg(fieldCount).arg(
theLayerFieldName).arg(layerGeomType).arg(theLayerFeatureCount), tr( "OGR" ) );
      QString layer_name=QString("%1(%2)").arg( theLayerName ).arg(theLayerFieldName);
      if ((fieldCount == 1)  || (theLayerName.endsWith( QString("(%1)").arg(theLayerFieldName))))
      { // gdal previous 2.0: on tables with 1 geometry, may not use the ogr-format 'table_name(field_name)' ; or already formatted in
the  ogr-format 'table_name(field_name)'
       layer_name=QString("%1").arg(theLayerName);
      }
      mSubLayerList << QString( "%1:%2:%3:%4" ).arg( layer_number++ ).arg(layer_name).arg( theLayerFeatureCount == -1 ? tr(
"Unknown" ) : QString::number( theLayerFeatureCount ) ).arg( geom );
     }
     else
     { // This may not be needed
      QgsDebugMsg( "Unknown geometry type, count features for each geometry type" );
      ..
     }
    }
```

All of the OGR function used existed in both in gdal 1.* and 2.*.
The main difference between 1.* and 2.0 is that label-name syntax 'table_name(field_name)'
in 1.*: may only be used in a table with more than 1 geometry
in 2.* may be used for all geometries fields

Unfortunately I cannot test this on the present master code, since I cannot get it compiled.
This is the reason I have not submitted a diff or pull request.

These changes were tested on the qgis that I use and can compile (2.1.0) and return that same result when using gdal 1.11.2 and the present gdal 2.0.0dev code.
I have, however, looked a the ogr specific code in the areas that were change and it looksthe same to me.
The only difference is that now a `if ( wkbFlatten( layerGeomType ) != wkbUnknown )`
is done, but since there is no 'wkbUnknown25D' there is noting to flatten and is really not needed.

History

#1 - 2015-03-31 03:07 AM - Mark Johnson

For setSubsetString, the 'OGRSQL' parameter must be used on layers that use the 'table_name(field_name)' convention.

this matter was talked about in the gdal ticket:

http://trac.osgeo.org/gdal/ticket/5903

and the final conclusion of Even Rouault was that this code must be adapted to work correctly with gdal 2.0 in cases where there is more than 1 geometry in a table.

#2 - 2015-03-31 03:50 AM - Giovanni Manghi

- Status changed from Open to Feedback

Hi,

many thanks for this. I would appreciate a lot if you could raise this issues also on the qgis developers mailing list and/or on qgis github repo with a patch. Thanks!

#3 - 2015-03-31 04:16 AM - Mark Johnson

Since I cannot compile the present version, creating a patch is difficult since there are other changes in the file.

#4 - 2015-04-05 12:13 PM - Paolo Cavallini

- Priority changed from Normal to High

#5 - 2015-04-05 01:53 PM - Jürgen Fischer

- Priority changed from High to Normal
- Status changed from Feedback to Open

#6 - 2017-03-20 03:27 AM - Mark Johnson

- Target version changed from Future Release - High Priority to Version 3.0

This matter needs to be resolved for QGIS 3.

#7 - 2017-05-01 01:07 AM - Giovanni Manghi

- Regression? set to No
- Easy fix? set to No

#8 - 2018-05-06 03:23 AM - Nyall Dawson

- Status changed from Open to Closed
- Description updated

Fixed in QGIS 3

#9 - 2018-05-06 07:53 AM - Mark Johnson

- Assignee set to Nyall Dawson

Are you sure this has been corrected?

fdef.GetGeomFieldCount() will return the amount, thus a loop should exist from 0 to amount

```
QgsOgrProvider::addSubLayerDetailsToSubLayerList
  QgsOgrFeatureDefn &fdef = layer->GetLayerDefn();
  // Get first column name,
  // TODO: add support for multiple
  QString geometryColumnName;
  if ( fdef.GetGeomFieldCount() )
  {
    OGRGeomFieldDefnH geomH = fdef.GetGeomFieldDefn( 0 );
    geometryColumnName = QString::fromUtf8( OGR_GFld_GetNameRef( geomH ) );
  }
```

This code retrieves only the first, so if more than one exists the others will not be listed.
There is also the 'TODO' comment to add support for this.