# OGR VFK Driver Implementation Issues

Martin Landa

Department of Mapping and Cartography, FCE, CTU in Prague, Thákurova 7
166 29, Prague 6, Czech Republic
martin.landa@fsv.cvut.cz

**Abstract.** OGR is a C++ open source library providing read (and sometimes write) access to a variety of vector GIS formats including ESRI Shapefile, S-57, SDTS, PostGIS, Oracle Spatial, and Mapinfo formats. The library is commonly used by Free Software projects like GRASS GIS, QGIS or MapServer, but also by proprietary products (e.g., ESRI ArcGIS 9.2+). The library provides access to a many vector formats, however support for specific Czech cadastral exchange data format (VFK) is missing.

There are already at least two projects which are focused on VFK data processing using Free Software tools. The first one is the v.in.vfk GRASS program which converts VFK data into GRASS native vector format (Martin Landa, CTU in Prague, 2005). This program can be used only within GRASS GIS package, which is a major drawback of this approach. Moreover, further development of the v.in.vfk module is not supported by the author. Other similar project is called "Open Cadastre" (Karel Jedlička, et al., University of West Bohemia, 2007) and provides a group of tools for importing VFK data into PostGIS geospatial database. Both projects are focused on a selected tool (GRASS GIS or PostGIS). From this point of view the OGR VFK driver would represent a more general solution - VFK data format would be supported by all software packages which use this library (GRASS GIS, QGIS, PostGIS, MapServer, ESRI ArcGIS 9.2+, etc.).

This paper covers ideas on the implementation of support for the VFK data format in OGR library. Capability to read and process cadastral data in VFK format using Free Software tools would be a very interesting option for the Czech users.

**Keywords:** Free Software, OGR, cadastre, exchange data format, data conversion

## Background

Cadastre of real estates is one of the basic geospatial data sources in most countries around the world. Spatial information collected from graphical and attributive cadastral data is frequently used in Geographical Information Systems (GIS) [1].

Ability to read cadastral data is becoming very important when solving various tasks in GIS. Cadastral data can be distributed in various data formats – standard(ish) GIS vector formats like ESRI Shapefile or specific internal formats, e.g. VFK data format for Czech cadastre of real estate.

## ISKN

*The real estate cadastre information system* (ISKN), launched in 2001, is an information system inteded for improving performance of real estate registry and public administration, and for assuring real estate registry services for users. ISKN has been developed and implemented by Ness Technologies company and CUZK (Czech Office for Surveying, Mapping and Cadastre). ISKN enables users to access cadastre data electronically, e.g. via remote web access. The ISKN project establishes an integral information system and speeds up and improves the quality of services provided to the public and to the institutions. Key benefits of the system include immediate availability of cadastre data from anywhere, including electronically signed documents, and facilitation of public administration operations by connecting the real estate cadastre with other public registries. It is the essential system supporting the functioning of the Czech real estate market [2,4].

ISKN contains geographical data and official maps as well as survey data, property attributes, and fund documentation.

## OGR Library

*The OGR Simple Features Library* is a C++ open source library (and command-line tools) providing read (and sometimes write) access to a variety of vector data formats including ESRI Shapefile, S-57, SDTS, PostGIS, Oracle Spatial, and Mapinfo mid/mif and TAB formats. OGR is a part of the GDAL library (http://gdal.osgeo.org). This library is released under an X/MIT style Open Source license by the Open Source Geospatial Foundation (OSGeo; http://www.osgeo.org). As a library, it presents a single abstract data model to the calling application for all supported formats. It may also be built with a variety of useful command-line utilities for data translation and processing [7].

## Exchange Data Format of Cadastre of Real Estate

Cadastral data can be distributed in the exchange data format (VFK format). VFK is a plain text format which includes alphanumeric and graphical data. For more information about the VFK format see author's master thesis [6] or the official VFK format documentation provided by CUZK [3]. The VFK file has four main parts:

- *Header* - &H
- *Data Blocks* - &B
- *Data Records* - &D
- *End of file* - &K

Data record ends by '<CR><LF>' characters, '¤' indicates that the record continues on the next line.

### Header

The header contains information about data content, extent, recency or constraint conditions. Header records start with &H followed by a property name and a list of values separated by semicolon, e.g. version of VFK format.

```
&HVERZE;”3.0”
```

### Data Blocks

Data blocks are represented by two types of the elements.

- Definition of data block (&B) - ordered list of attributes and their data types (see tab. 1). Data block can be represented as a table in the relational database system.
- Data records (&D) - ordered list of values separated by semicolon.

Example of the data block definition and data record:

```
&BPAR;ID N30;STAV_DAT N2;DATUM_VZNIKU D;DATUM_ZANIKU D;PRIZNAK_KONTEXTU N1;
RIZENI_ID_VZNIKU N30;RIZENI_ID_ZANIKU N30;PKN_ID N30;PAR_TYPE T10;KATUZE_KOD
N6;KATUZE_KOD_PUV N6;DRUH_CISLOVANI_PAR N1;KMENOVE_CISLO_PAR N5;  ZDPAZE_KOD
N1;PODDELENI_CISLA_PAR N3;DIL_PARCELY N1;MAPLIS_KOD N30;  ZPURVY_KOD N1;DRUPOZ_KOD
N2;ZPVYPA_KOD N4;TYP_PARCELY N1;VYMERA_PARCELY N9;CENA_NEMOVITOSTI
N14.2;DEFINICNI_BOD_PAR T100;TEL_ID N30;PAR_ID N30;BUD_ID N30;IDENT_BUD T1

&DPAR;28256708;0;"12.12.199612:00:00";"";2;96835708;;;"PKN";675008;;1;100;;;;6780;2
;13;;;113;;"";13913708;;4282708;"a"
```

Note: Attribute identifiers are defined in the Czech language based on [3]. Only identifiers which are important for understanding this paper have been translated and their meaning explained.

**Table 1.** VFK data types.

| Code | Data type | Number |
|---|---|---|
| N | Numeric | Length (precision) |
| T | Text | Text length |
| D | Date Timestamp | |

## OGR VFK Driver

The VFK data file is recognized as an OGR *datasource*, and the VFK data blocks as OGR *layers.*

### Header processing

Data records (&H) of the header are stored as metadata of the OGR datasource. Here is a sample of the code to print metadata information:

```
OGRVFKDataSource *poDS;
// open OGR datasource
//...
// print metadata
std::cout << "version: " << poDS->GetInfo("VERZE") << std::endl;
```

### Data blocks processing

Data blocks are represented in OGR data model by a layer. Data blocks definition (&B) provides OGR layer name, list of the attributes and their data types. Data records (&D) define OGR *features*.

```
### print list of available OGR layers and their feature type
$ ogrinfo data.vfk
1: PAR (Polygon)
2: BUD (Polygon)
3: ZPOCHN (None)
...
59: ZPMZ (None)
60: NZZP (None)
61: SPOL (Point)
```

For example, according to the following data block definition

```
&BSBP;ID N30;STAV_DAT N2;DATUM_VZNIKU D;DATUM_ZANIKU D;PRIZNAK_KONTEXTU N1;
RIZENI_ID_VZNIKU N30;RIZENI_ID_ZANIKU N30;BP_ID N30;PORADOVE_CISLO_BODU N38;OB_ID
N30;HP_ID N30;DPM_ID N30;PARAMETRY_SPOJENI T100;ZVB_ID N30
```

is created OGR layer 'SBP'.

```
# print OGR layer 'SBP' info (list of attributes)
$ ogrinfo data.vfk SBP
Layer name: SBP
ID: Integer (30.0)
STAV_DAT: Integer (2.0)
DATUM_VZNIKU: DateTime (0.0)
DATUM_ZANIKU: DateTime (0.0)
PRIZNAK_KONTEXTU: Integer (1.0)
RIZENI_ID_VZNIKU: Integer (30.0)
RIZENI_ID_ZANIKU: Integer (30.0)
BP_ID: Integer (30.0)
PORADOVE_CISLO_BODU: Integer (38.0)
OB_ID: Integer (30.0)
HP_ID: Integer (30.0)
DPM_ID: Integer (30.0)
PARAMETRY_SPOJENI: String (100.0)
ZVB_ID: Integer (30.0)
```

### Geometry of map features

Selected data blocks are used for building geometry of the features of the digital cadastral map (see fig. 1).

**Points.** Data block 'SOBR' (points of the planimetric survey), 'OBBP' (points of the minor control), 'SPOL' (points of the survey) contains information about points. Coordinates (x, y) of the points are stored as 'SOURADNICE_Y' and 'SOURADNICE_X' attributes.

```
# print point (selected by ID) info - 'SOBR' layer
$ ogrinfo data.vfk SOBR -where 'ID = 311040708'

Layer name: SOBR
Geometry: Point
...
OGRFeature(SOBR):1
  ID (Integer) = 311040708
  STAV_DAT (Integer) = 0
  ...
  UPLNE_CISLO (Integer) = 1014270001
  SOURADNICE_Y (Real) =  650451.45
  SOURADNICE_X (Real) = 1069791.42
  KODCHB_KOD (Integer) = 4
  POINT (-650451.449999999953434 -1069791.419999999925494)
```

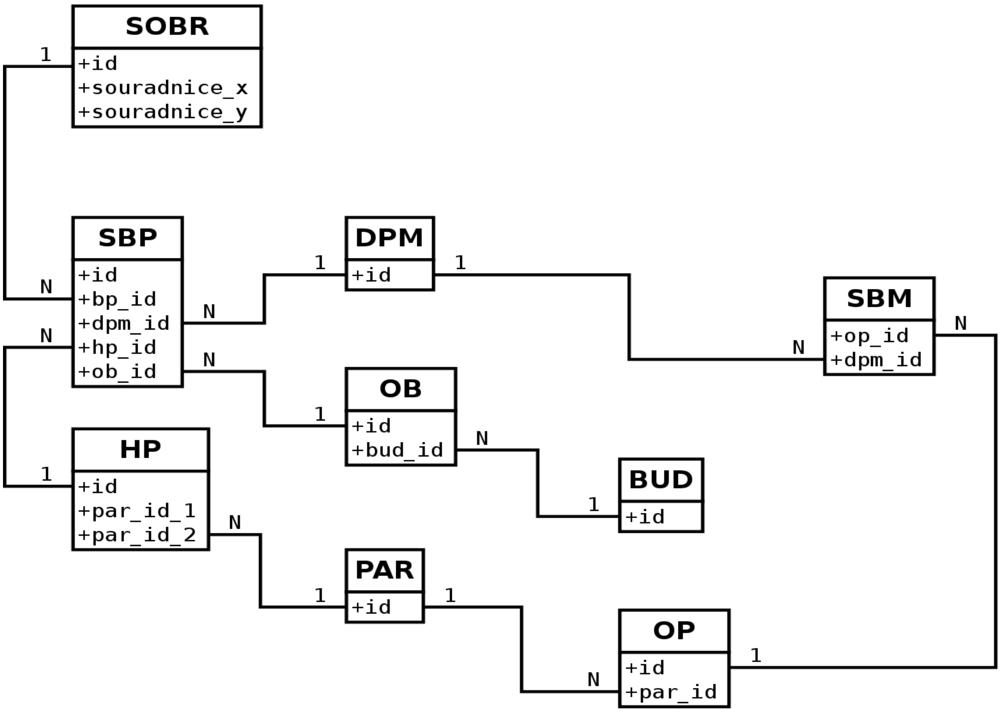


**Fig. 1.** VFK data block diagram. List of the data blocks to construct features geometry.

**Lines.** Data blocks defining linear features are 'SBP' and 'SBM'. Geometry of the linear features is constructed at least from two data records (i.e. data record defines a vertex). Order of the vertices is determined by the attribute 'PORADOVE_CISLO_BODU'. Attribute 'BP_ID' is used as a foreign key in the data block 'SOBR' (attribute 'ID'). E.g. based on the following data records is constructed a line with one segment (i.e. two vertices).

```
SOBR:

    ID     | ... | SOURADNICE_Y | SOURADNICE_X | KODCHB_KOD
-----------+-----+--------------+--------------+------------
 311127708 |     |    651377.77 |   1069006.53 |          4
 311130708 |     |    651381.75 |   1069005.69 |          4
```

```
SBP:

    ID     | ... | PORADOVE_CISLO_BODU | ... |    BP_ID   | ...
-----------+-----+---------------------+-----+------------+-----
 286100708 |     |                   1 |     |  311127708 |
 286101708 |     |                   2 |     |  311130708 |

OGRFeature(SBP):1
  ID (Integer) = 286100708
  STAV_DAT (Integer) = 0
  ...
  OB_ID (Integer) = (null)
  HP_ID (Integer) = 151540708
  DPM_ID (Integer) = (null)
  PARAMETRY_SPOJENI (String) = "4"
  ZVB_ID (Integer) = (null)
  LINESTRING (-651377.77 -1069006.53, -651381.75 -1069005.69)
```

**Polygons.** Some data blocks (e.g. 'PAR') defines polygon features. Exterior and interior boundaries (rings) are constructed from the lines defined in 'SPB' layer (in the case of 'PAR' those lines which have defined 'HP_ID' attribute).

```
OGRFeature(PAR):1
  ID (Integer) = 92340708
  STAV_DAT (Integer) = 0
  ...
  TEL_ID (Integer) = 64156708
  PAR_ID (Integer) = (null)
  BUD_ID (Integer) = 25735708
  IDENT_BUD (String) = "a"
  POLYGON ((-651240.46999999997206 -1069564.110000000102445
            -651217.300000000046566 -1069525.409999999916181,
            -651212.400000000023283 -1069517.21999999997206,
            -651215.810000000055879 -1069515.679999999934807,
            -651219.959999999962747 -1069524.169999999925494,
            -651225.680000000051223 -1069521.5,
            -651235.0 -1069517.149999999906868,
            -651239.520000000018626 -1069527.510000000009313,
            -651241.260000000009313 -1069526.75,
            -651242.770000000018626 -1069530.179999999934807,
            -651241.020000000018626 -1069530.939999999944121,
            -651253.819999999948777 -1069560.239999999990687,
            -651244.46999999997206 -1069564.21999999997206,
            -651240.46999999997206 -1069564.110000000102445))
```

### Usage example

OGR VFK driver allows easily import VFK data to PostGIS geodatabase. The following example shows how to import data to PostGIS database, create a view and export the view as a layer in ESRI Shapefile data format.

```
# default encoding is WIN1250
$ export PGCLIENTENCODING=WIN1250

# import VFK data to PostGIS geodatabase
$ ogr2ogr -f PostgreSQL PG:dbname=kn data.vfk

# create a view
CREATE VIEW par_view as
 SELECT par.ogc_fid,par.wkb_geometry,kat.nazev,par.vymera_parcely,
        par.par_type,map.oznaceni_mapoveho_listu
  FROM par JOIN katuze AS kat ON par.katuze_kod = kat.kod
  JOIN maplis AS map ON par.maplis_kod = map.id;

# export view as a Shapefile
$ ogr2ogr -f 'ESRI Shapefile' . PG:dbname=kn par_view
```

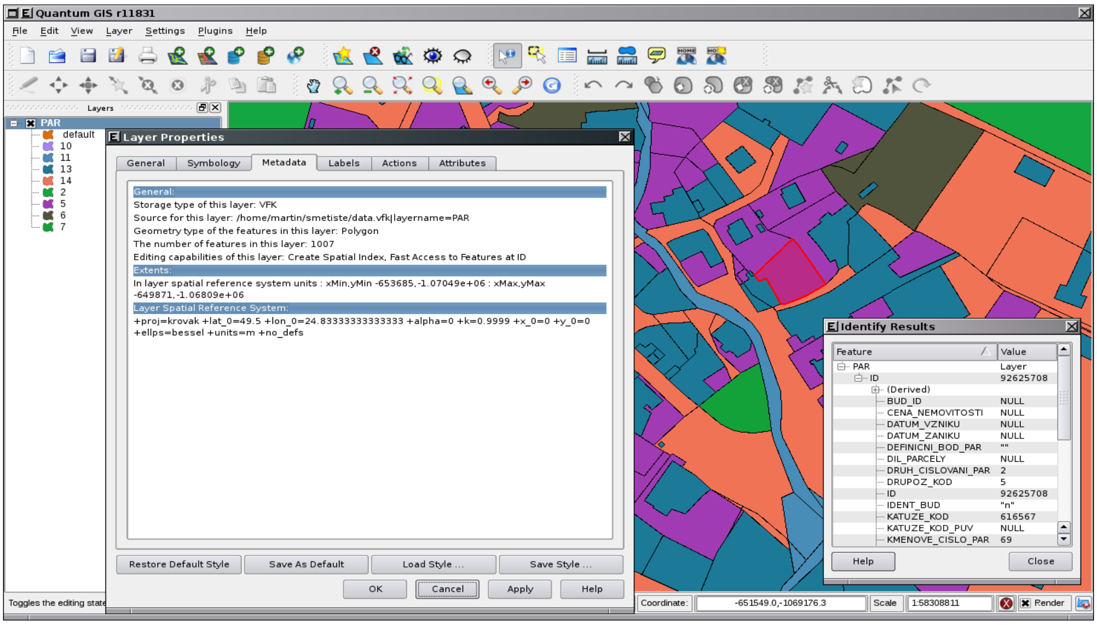
```
# check the result
$ ogrinfo par_view.shp -al -fid 1

Layer name: par_view
Geometry: Polygon
Feature Count: 1007
Extent: (-653684.850000, -1070485.950000) - (-649871.280000, -1068089.260000)
Layer SRS WKT:
(unknown)
nazev: String (80.0)
oznaceni_m: String (80.0)
par_type: String (80.0)
vymera_par: Real (11.0)
OGRFeature(par_view):1
  nazev (String) = Bylany
  oznaceni_m (String) = DKM
  par_type (String) = PKN
  vymera_par (Real) = 1108
  POLYGON ((-651241.459999999962747 -1069577.010000000009313,
            -651244.46999999997206  -1069564.21999999997206,
            -651240.46999999997206 -1069564.110000000102445,
            -651205.959999999962747 -1069563.090000000083819,
            -651191.959999999962747 -1069562.679999999934807,
            -651190.339999999967404 -1069581.110000000102445,
            -651240.050000000046566 -1069588.939999999944121,
            -651241.459999999962747 -1069577.010000000009313))
```

## Conclusion

The paper is focused on interoperability of data and systems. The described solutions provide users with new possibilities of processing cadastral data distributed in VFK data format using Free Software and Open Source tools. The working horse of this solution is a well supported OGR library. OGR abstract data model enables reading data provided in VFK data format without any restrictions.

Beside implementation of the OGR VFK driver, the development of specialized tools for QGIS (http://www.qgis.org) and GRASS GIS (http://grass.osgeo.org) projects is planned. Currently there is no complex open source tool for cadastral data processing on the market. In the short-term period, the development of the plugin for QGIS to display, process and query digital catastral maps is planned.

Up-to-date information about the project can be found at http://grass.fsv.cvut.cz/gwiki/OGR-VFK.

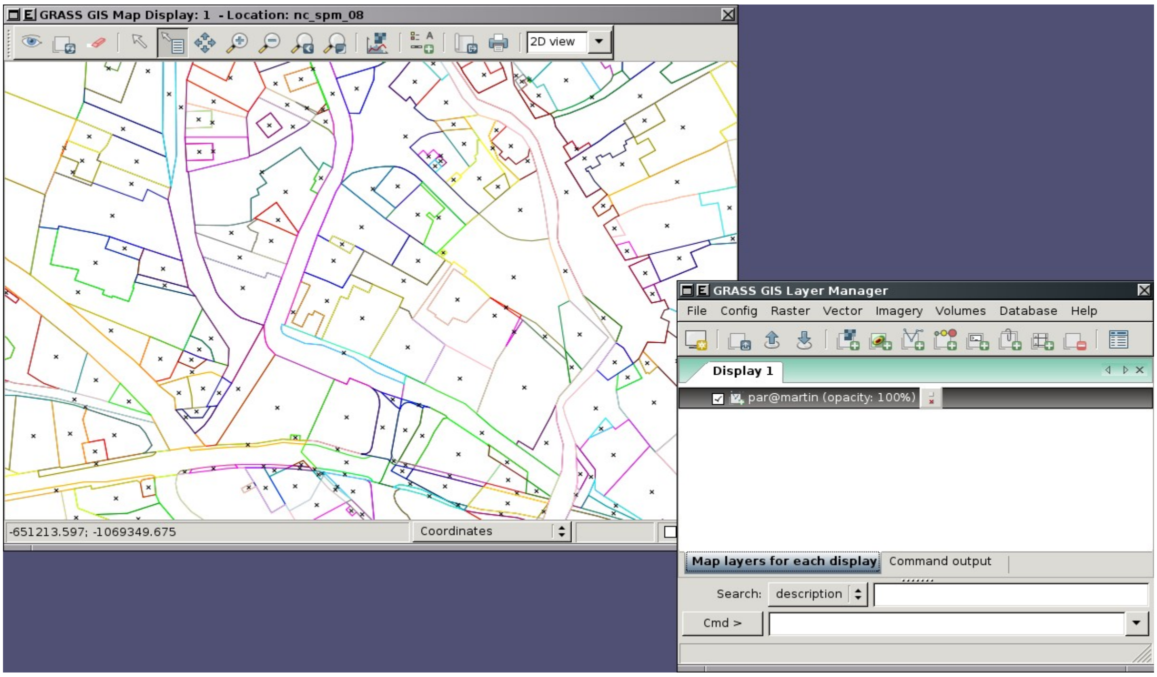


General:
Storage type of this layer: VFK
Source for this layer: /home/martin/smetiste/data.vfk|layername=PAR
Geometry type of the features in this layer: Polygon
The number of features in this layer: 1007
Editing capabilities of this layer: Create Spatial Index, Fast Access to Features at ID
Extents:
In layer spatial reference system units : xMin,yMin -653685,-1.07049e+06 : xMax,yMax -649871,-1.06809e+06
Layer Spatial Reference System:
+proj=krovak +lat_0=49.5 +lon_0=24.83333333333333 +alpha=0 +k=0.9999 +x_0=0 +y_0=0 +ellps=bessel +units=m +no_defs

| Feature | Value |
|---|---|
| PAR | Layer |
| ID | 92625708 |
| (Derived) | |
| BUD_ID | NULL |
| CENA_NEMOVITOSTI | NULL |
| DATUM_VZNIKU | NULL |
| DATUM_ZANIKU | NULL |
| DEFINICNI_BOD_PAR | "" |
| DIL_PARCELY | NULL |
| DRUH_CISLOVANI_PAR | 2 |
| DRUPOZ_KOD | 5 |
| ID | 92625708 |
| IDENT_BUD | "n" |
| KATUZE_KOD | 616567 |
| KATUZE_KOD_PUV | NULL |
| KMENOVE_CISLO_PAR | 69 |

Coordinate: -651549.0,-1069176.3 Scale 1:58308811

**Fig. 2.** Visualization of the cadastral data in QGIS.



-651213.597; -1069349.675

☑ par@martin (opacity: 100%)

**Fig. 3.** Visualization of the cadastral data in GRASS GIS.

## References

[1] Slaboch V. *Combining Cadastre of Real Estate with GIS Data – A Contribution to Sustainability.* 2nd FIG Regional Conference, Morocco, December 2003. http://www.fig.net/pub/morocco/proceedings/TS15/TS15_1_slaboch.pdf

[2] CUZK. ISKN project. http://www.cuzk.cz/Dokument.aspx?PRARESKOD=10&MENUID=10283&AKCE=DOC:10-INFORMACNI_SYSTEM

[3] CUZK. VFK documentation. http://www.cuzk.cz/Dokument.aspx?PRARESKOD=998&MENUID=0&AKCE=DOC:10-VF_ISKNTEXT

[4] Jedlička K. el al. *Otevřený katastr - svobodné internetové řešení pro prohlížení dat výměnného formátu katastru nemovitostí.* Geoinformatics FCE CTU 2007, volume 2. ISSN 1802-2669. http://gama.fsv.cvut.cz/data/geoinformatics/2007/09/19/geoinformatics-fce-ctu-2007-02.pdf

[5] Kocáb M. *Geographical Information Systems and Cadaster of Real Estates.* 50 years of the Research Institute of Geodesy, Topography and Cartography. Jubille Proceedings 1954-2004. Zdiby 2005. ISBN 80-85881-23-3.

[6] Landa M. *Implementation of a GRASS module for importing data in Czech cadastral exchange data format.* Master thesis. CTU in Prague, 2005. http://gama.fsv.cvut.cz/~landa/publications/2005/diploma_thesis/martin.landa-thesis.pdf

[7] Wikipedia, the free encyclopedia. GDAL/OGR Library. http://en.wikipedia.org/wiki/GDAL

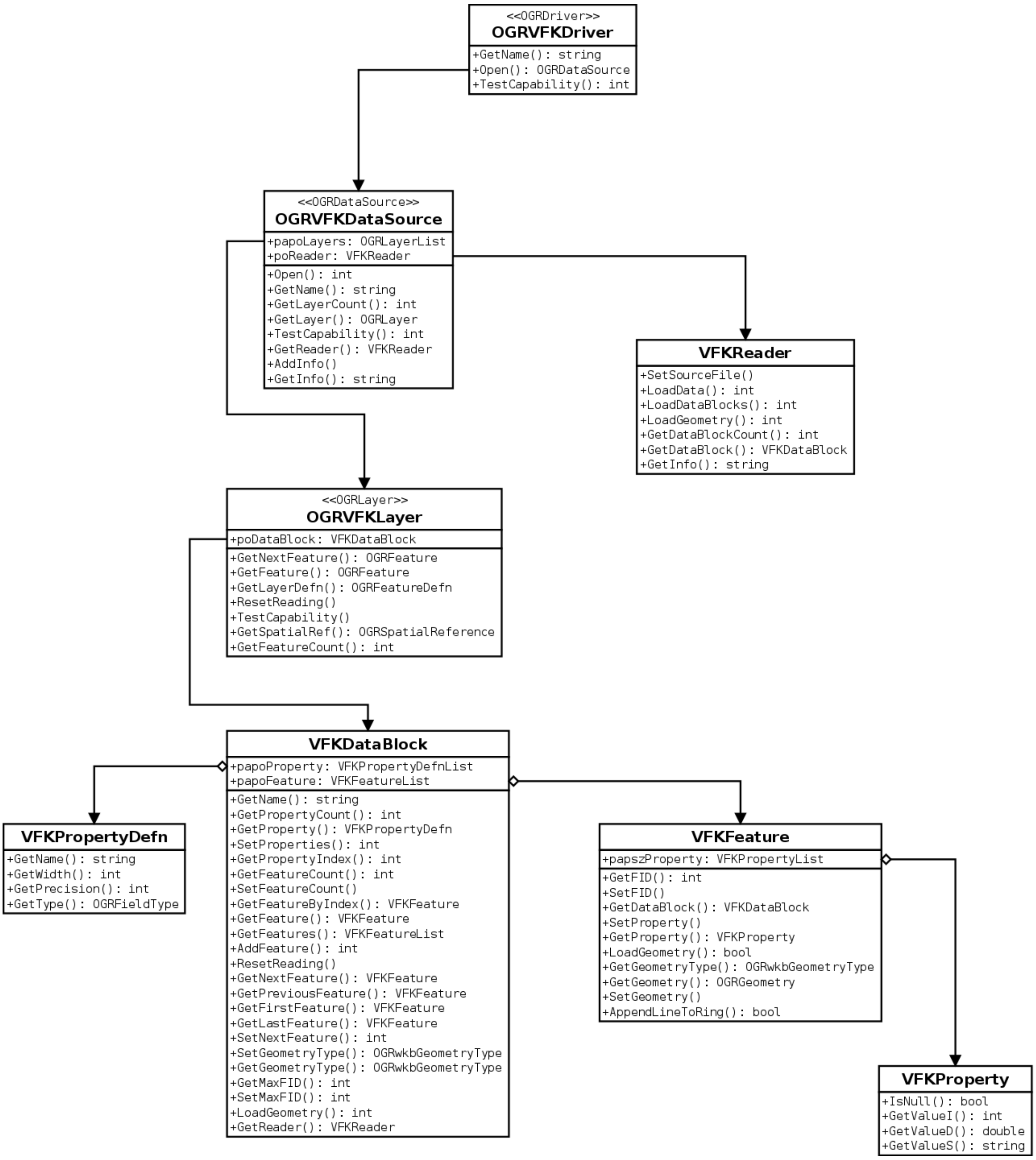


**Fig. 4.** OGR VFK driver - C++ classes diagram.