Carl Reed, February 2017.

**Comments on GeoPackage version 1.2.**

Thank you for the opportunity to review and comment on the draft GeoPackage 1.2 version. This comment document is divided into three sections: General Comments, Comments on the revisions proposed for 1.2, and Comments on version 1.1 of the standard. For the latter, the comments are mostly editorial in nature. There are some comments for some sentences or clauses that I think may need some additional clarification.

GeoPackage represents considerable effort and creative thought. My comments are targeted at improving GeoPackage to 1.) help enhance uptake and 2.) enhance interoperability of GeoPackages. Please accept my comments in the spirit that they are intended.

I should also add that some may wonder why I am providing many of these comments now as opposed to when I was OGC staff. Simply, bandwidth. As OGC staff, I occasionally did deep reviews and edits of candidate standards. This usually occurred for standards being developed by a SWG with little or no previous experience in developing an OGC standard and/or where few if any of the editors were native English speakers. For all OGC standards – either new or revisions – I also did reviews and minor edits. Edits were usually confined to all sections up to and including Scope and Introduction. I also checked for structural errors in the documents. Basically, I trusted the SWG to develop, agree to, and write the normative content.

So now I have time to focus on a few OGC standards activities and do those deep edit dives I did not have time for as OGC staff ☺

Thanks

Carl

Contents

[General Comments. 1](#_Toc476037897)

[Comments on revisions as documented in the Release Notes 2](#_Toc476037898)

[6.1. 221 Adding Attributes Section 2](#_Toc476037905)

[6.2. 234 Deprecate Requirement #69 2](#_Toc476037906)

[6.3. 235 Deprecate Extensions F.2, F.4, and F.5 3](#_Toc476037907)

[6.4. 242 Add Elevation Extension to Standard 3](#_Toc476037908)

[6.5. 255 Update versioning mechanism, allow for version increments in SQLite header 4](#_Toc476037909)

[6.6. 258 Column Name for WKT for Coordinate Reference Systems 4](#_Toc476037910)

[Comments on GeoPackage – beyond those documented in the release notes 4](#_Toc476037911)

[From the Cover Page 4](#_Toc476037912)

[Introduction 4](#_Toc476037913)

[1. Base 5](#_Toc476037914)

[1.1. Core 5](#_Toc476037915)

[2 Options 7](#_Toc476037916)

# General Comments.

* The release notes do not reference specific clauses. This makes finding the relevant content in the standard very difficult!! Having the issue number is fine but should be labeled as such. For example, the release notes have a heading: “6.1. 221 Adding Attributes Section”. There is no indication as to where this is in the document.
* Inconsistent use of “standard” and “specification”: All ISO and OGC standards should be referred to as standards and not specifications. There is an OGC policy to this effect.
* There are numerous “SHALLs” scattered through the text but are not specifically called out as an official “Requirement”. Why not? They should be.
* The HTML version has many links (good) from one clause to another. However, the links (urls) are no good in PDF and Word files. Therefore, the text should also contain Section (clause) reference numbers. Examples are provided below.
* Draft (candidate) standards should not state that they are approved standards, especially if they are publicly available. (See below)
* PDF files are a pain to work with. For example, the reader cannot navigate from the ToC to a clause/section of interest. See above also.
* I still believe more work is required to restrict and clarify the use and role of extensions. The current standard is very much open to abuse in this area. The ability to specify non-interoperable extensions is way too easy. Further, I would consider adding wording such as:
  + - This document specifies the GeoPackage core. Every implementation of a GeoPackage shall adhere to the requirements as specified in this standard. Extensions to the core define requirements to add additional functionality, such as the use of spatial indexing. Indeed, several GeoPackage extensions are required in order to completely specify a compliant GeoPackage whose implementation is fully operational. This document indicates which extensions, at a minimum, need to be considered in addition to this core to allow for a complete GeoPackage implementation.
    - The above is paraphrased from the WCS core standard. I would go further and follow the documentation model of other OGC standards and move all “core” extensions to a separate document and label that document as such. I would go even further and state that only extensions submitted to the OGC and approved using the OGC standards process SHALL be considered extensions.

# Comments on revisions as documented in the Release Notes



### 221 Adding Attributes Section

I am assuming this is clause 2.4 in the draft standard. I would suggest providing a definition of what is meant by “attribute”. I would also suggest providing some informative examples. I suggest this to make sure that there is no confusion. For example, in GML the definition of attribute is: attribute <XML> name-value pair contained in an element. I suspect that this is different from how the term is being used in GeoPackage. ISO TC 211 uses the definition, “named property of an entity”. Attribute data can be spatial or aspatial or a combination of both. I think to be clear what we are talking about is attribute data with no associated geometry. As such, it might be helpful to provide an example of aspatial attribute data.

### 234 Deprecate Requirement #69

No comments or issues.

### 235 Deprecate Extensions F.2, F.4, and F.5

No issues. However, a more general observation (comment 18 below): If there are geometries specified, including extensions that define additional geometries, why not simply state “All geometries specified in this standard (core) or in user defined extensions SHALL be consistent with ISO 19107 and thereby consistent with OGC/ISO Simple Features”?

### 242 Add Elevation Extension to Standard

I am assuming that this is Annex F.11. First, I believe that there should be clear definition of what is meant by the term “elevation”. I would check the CDB stand for an example (Clause 5.6.1). Further, CDB also uses TIFF as the storage format for elevation data. I have more comments on the use of TIFF below. I mention CDB because that standard has been implemented and used in numerous modeling and simulation systems for many, many years. There is considerable implementation knowledge instantiated in the CDB standard.

I would edit the sentence, “The GeoPackage Standards Working Group (SWG) has developed the ability to store 16-bit and 32-bit tiled gridded elevation data in a GeoPackage” to clearly state what the extension does define. No need to say that the SWG approved this extension. A non OGC reader had no idea what an OGC SWG is nor do they care.

Table 30 mentions EPSG 4979. Perhaps the extensions should include a note providing the URL to the definition? The URL is http://www.epsg-registry.org/report.htm?type=selection&entity=urn:ogc:def:crs:EPSG::4979&reportDetail=long&style=urn:uuid:report-style:default-with-code&style\_name=OGP%20Default%20With%20Code&title=

What about bathymetry? There are no words about whether bathymetry is considered elevation data. Should have some mention.

While the draft Elevation Extension does provide some requirements related to the use of TIFF for encoding a 32 bit elevation matrix, I would suggest that more guidance is required to better ensure interoperability and the ability of content providers to produce elevation content encoded as TIFF that all clients will be able to ingest and use. I would check out CDB Volume 10 Implementation Guidance, Chapter 8 TIFF Implementation Guidance. For example, In GeoPackage 1.2 there is no guidance in the elevation extension regarding color palettes, color maps, date/time, and many other fields contained in a TIFF encoding.

A more general comment has to do with the inclusion of the elevation extension in the main GeoPackage document. I would really encourage extensions such as the Elevation Extension be incorporated in a separate OGC document that follows the OGC document template. I would also strongly encourage the SWG to provide more informative and normative material in the Elevation Extension to make sure that implementations are truly interoperable. I would also encourage the SWG to consider the EE to be an OGC Best Practice rather than a full standard.

### 255 Update versioning mechanism, allow for version increments in SQLite header

No issues or comments.

### 258 Column Name for WKT for Coordinate Reference Systems

No issues or comments related to this revision.

# Comments on GeoPackage – beyond those documented in the release notes

### From the Cover Page

**Warning**

This document is an OGC Member approved international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation. This is a technical amendment to OGC 12-128r1.

Document type: OGC® Publicly Available Standard  
Document subtype: Encoding Standard   
Document stage: Approved  
Document language: English

<snip>

## Introduction

Mobile device users who require map/geospatial application services and operate in disconnected or limited network connectivity environments are challenged by limited storage capacity and the lack of open format geospatial data to support these applications. The current situation is that each map/geospatial application requires its own potentially proprietary geospatial data store. These separate application-specific data stores may contain the same geospatial data, wasting the limited storage available, and requiring custom applications for data translation, replication, and synchronization to enable different map/geospatial applications to share the same world view. In addition, many existing geospatial data stores are platform-specific, which means that users with different platforms must translate data to share it.

An open, standards-based, application-independent, platform-independent, portable, interoperable, self-describing, GeoPackage (GPKG) data container, API and manifest are needed to overcome these challenges and to effectively support multiple map/geospatial applications such as fixed product distribution, local data collection, and geospatially enabled analytics. This standard is intended to facilitate widespread adoption and use of GeoPackages by both COTS and open-source software applications on enterprise production platforms as well as mobile hand-held devices [[B1]](http://www.geopackage.org/spec/#B1) [[B2]](http://www.geopackage.org/spec/#B2), given that mobile hand held devices do not yet have the processing power or battery life to effectively tackle difficult geospatial product production and analysis tasks. An application that accesses a GPKG will make use of the GPKG capabilities it requires; few if any such applications will make use of all GPKG capabilities.

This OGC® Encoding Standard defines GeoPackages for exchange and GeoPackage SQLite Extensions for direct use of vector geospatial features and / or tile matrix sets of earth images and raster maps at various scales. Direct use means the ability to access and update data in a “native” format without intermediate format translations in an environment (e.g. through an API) that guarantees data model and data set integrity and identical access and update results in response to identical requests from different client applications.

A **GeoPackage** is a platform-independent SQLite [[5]](http://www.geopackage.org/spec/#5) database file that contains geospatial; data and metadata tables shown in [GeoPackage Tables Overview](http://www.geopackage.org/spec/#geopackage_tables_figure) below. The file, also known as a GeoPackage continaer, provides definitions, integrity assertions, format limitations and content constraints. The allowable content of a GeoPackage is entirely defined in this standard.

An **Extended GeoPackage** is a **GeoPackage** that contains any additional data elements (tables or columns) or SQL constructs (data types, functions, indexes, constraints or triggers) that are not automatically maintained within the SQLite data file or that result in a change in behavior not specified in GeoPackage standard.

A **GeoPackage** MAY be “empty” (contain user data table(s) for vector features, non-spatial attributes, and/or tile matrix pyramids with no row record content). Conversely, a GeoPackage may contain one or many vector feature type records and /or one or many tile matrix pyramid tile images. GeoPackage metadata CAN describe GeoPackage data contents and identify external data synchronization sources and targets. A GeoPackage MAY contain spatial indexes on feature geometries and SQL triggers to maintain indexes and enforce content constraints.

<SNIP>

## 1. Base

The requirements specified in this section (Clause 1) serve as the base for the core, for options specified in clause [Options](http://www.geopackage.org/spec/#_options) and extensions specified in clause [Registered Extensions (Normative)](http://www.geopackage.org/spec/#registered_extensions). All gpkg\_\* tables and views and all tiles user data tables specified in this standard SHALL have only the specified columns and table constraints. Any features user data tables MAY have columns in addition to those specified. All specified table, view, column, trigger, and constraint name values SHALL be lowercase.

### 1.1. Core

The equirement in this Section 1.1 (this clause) SHALL be implemented by every **GeoPackage** and **GeoPackage SQLite Configuration**. They define a core GeoPackage.

#### 1.1.1. SQLite Container

The SQLite software library provides a self-contained, single-file, cross-platform, serverless, transactional, open source RDBMS container. The GeoPackage standard defines a SQL database schema designed for use with the SQLite software library. Using SQLite as the basis for GeoPackage simplifies production, distribution and use of GeoPackages and assists in guaranteeing the integrity of the data they contain.

<SNIP>

Requirement 2

A GeoPackage SHALL contain a value of 0x47504B47 ("GPKG" in ASCII) in the "application\_id" field of the SQLite database header to indicate that it is a GeoPackage. [[3](http://www.geopackage.org/spec/#_footnote_3)] A GeoPackage SHALL contain an appropriate value in "user\_version" field of the SQLite database header to indicate its version. The value SHALL be in integer with a major version, two-digit minor version, and two-digit bug-fix. For GeoPackage Version 1.2 this value is 0x000027D8 (the hexadecimal value for 10200). [[4](http://www.geopackage.org/spec/#_footnote_4)]

<SNIP>

From requirement 4

In order to guarantee maximum interoperability between applications, GeoPackages SHALL NOT contain data elements (tables or columns), SQL constructs (data types, indexes, constraints or triggers) or extensions that are not specified in this encoding standard. SQLite databases that use constructs from the GeoPackage standard but extend those constructs to contain elements not specified in the core GeoPackage standard are referred to as Extended GeoPackages throughout this standard.

<SNIP>

###### 1.1.3.1.2. Table Data Values

Requirement 14

The table\_name column value in a gpkg\_contents table row SHALL contain the name of a SQLite table or view.

The data\_type specifies the type of content contained in the table, for example “features” per clause [Features](http://www.geopackage.org/spec/#features) (Section 2.1), “attributes” per clause [Attributes](http://www.geopackage.org/spec/#attributes), “tiles” per clause [Tiles](http://www.geopackage.org/spec/#tiles), or an implementer-defined value for other data tables per clause in an Extended GeoPackage.

<SNIP>

After requirement 15

The bounding box (min\_x, min\_y, max\_x, max\_y) provides an informative bounding box (not necessarily minimum bounding box) of the content. If the srs\_id column value references a geographic coordinate reference system (CRS), then the min/max x/y values are in decimal degrees; otherwise, the srs\_id references a projected CRS and the min/max x/y values are in the units specified by that CRS.

<SNIP>

## 2 Options

The optional requirements specified in Section 2 (Options) depend on the required capabilities specified in clause [Base](http://www.geopackage.org/spec/#_base) above (Section xx). Each subclause of this clause defines an indivisible module of functionality that can be used in GeoPackages. These modules are referred to as options. GeoPackages MAY use one or more options defined in this section. GeoPackages MAY omit the tables for options that are not used. As a minimum, a GeoPackage SHALL contain one user data table as defined by the Features or Tiles options in clauses [Features](http://www.geopackage.org/spec/#features) (Section xx) and [Tiles](http://www.geopackage.org/spec/#tiles) (Section xx) respectively.

Requirement 17

A GeoPackage SHALL contain features per clause [Features](http://www.geopackage.org/spec/#features) (Section xx) and/or tiles per clause [Tiles](http://www.geopackage.org/spec/#tiles) (Section xx) and row(s) in the gpkg\_contents table with lowercase data\_type column values of “features” and/or “tiles” describing the user data tables.

<SNIP>

In 2.1.1. The interpretation of the coordinates is subject to the coordinate reference systems associated to the point. All coordinates within a geometry object should be in the same coordinate reference systems.

<SNIP>

In 2.1.1. International standards [[9]](http://www.geopackage.org/spec/#9)[[10]](http://www.geopackage.org/spec/#10)[[11]](http://www.geopackage.org/spec/#11)[[12]](http://www.geopackage.org/spec/#12) have standardized practices for the storage,

<SNIP>

All geometry types described in this standard are defined so that instances of Geometry are topologically closed, i.e. all represented geometries include their boundary as point sets. This does not affect their representation, and open version of the same classes MAY be used in other circumstances, such as topological representations.

<SNIP>

Just below table 7.

Views of this table or view MAY be used to provide compatibility with the SQL/MM [[12]](http://www.geopackage.org/spec/#12) [SQL/MM View of gpkg\_geometry\_columns Definition SQL (Informative)](http://www.geopackage.org/spec/#sqlmm_gpkg_geometry_columns_sql) and OGC Simple Features SQL [[9]](http://www.geopackage.org/spec/#9)[[10]](http://www.geopackage.org/spec/#10)[[11]](http://www.geopackage.org/spec/#11) [SF/SQL VIEW of gpkg\_geometry\_columns Definition SQL (Informative)](http://www.geopackage.org/spec/#sfsql_gpkg_geometry_columns_sql) specifications.

<SNIP?

Clause 2.3/2.3.1 Extension Mechanism

A GeoPackage extension is a set of one or more requirements clauses that are documented by filling out the GeoPackage Extension Template in [GeoPackage Extension Template (Informative)](http://www.geopackage.org/spec/#extension_template).