**Open Geospatial Consortium**

Submission Date: <2020-01-21>

Approval Date: <2020-mm-dd>

Publication Date: <2020-mm-dd>

External identifier of this OGC® document: http://www.opengis.net/doc/IS/cdb-geopackage/1.2

Internal reference number of this OGC® document: 19-054

Version: 1.2

Category: OGC® Implementation Standard

Editor: Carl Reed

# Volume 13: OGC CDB Rules for Encoding CDB Vector Data using GeoPackage (Normative, Optional Extension).



**Warning**

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

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.

| | |
|---|---|
| Document type: | OGC® Standard |
| Document subtype: | if applicable |
| Document stage: | Draft |
| Document language: | English |

## License Agreement

Permission is hereby granted by the Open Geospatial Consortium, ("Licensor"), free of charge and subject to the terms set forth below, to any person obtaining a copy of this Intellectual Property and any associated documentation, to deal in the Intellectual Property without restriction (except as set forth below), including without limitation the rights to implement, use, copy, modify, merge, publish, distribute, and/or sublicense copies of the Intellectual Property, and to permit persons to whom the Intellectual Property is furnished to do so, provided that all copyright notices on the intellectual property are retained intact and that each person to whom the Intellectual Property is furnished agrees to the terms of this Agreement.

If you modify the Intellectual Property, all copies of the modified Intellectual Property must include, in addition to the above copyright notice, a notice that the Intellectual Property includes modifications that have not been approved or adopted by LICENSOR.

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

THE INTELLECTUAL PROPERTY IS PROVIDED "AS IS", WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, AND NONINFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

This license is effective until terminated. You may terminate it at any time by destroying the Intellectual Property together with all copies in any form. The license will also terminate if you fail to comply with any term or condition of this Agreement. Except as provided in the following sentence, no such termination of this license shall require the termination of any third party end-user sublicense to the Intellectual Property which is in force as of the date of notice of such termination. In addition, should the Intellectual Property, or the operation of the Intellectual Property, infringe, or in LICENSOR's sole opinion be likely to infringe, any patent, copyright, trademark or other right of a third party, you agree that LICENSOR, in its sole discretion, may terminate this license without any compensation or liability to you, your licensees or any other party. You agree upon termination of any kind to destroy or cause to be destroyed the Intellectual Property together with all copies in any form, whether held by you or by any third party.

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications. This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

# Table of Contents

## i. Abstract

This optional OGC CDB extension defines the requirements and provides CDB specific guidance on using GeoPackage containers in a CDB data store. There is a companion CDB Best Practice document that provide rules and guidance for transforming CDB structured Shapefiles into CDB structure GeoPackages that are compliant with the requirements and conformance classes as defined in this document.

## ii. Keywords

The following are keywords to be used by search engines and document catalogues.

ogcdoc, OGC document, CDB, GeoPackage

## iii. Preface

**Background for this optional CDB Extension**

The original requirement for this optional CDB extension was documented in OGC Change Request 545. This OGC change request was submitted based on work performed in OGC Testbed 13. The testbed activity and related change request captured a broad community requirement for being able to use GeoPackage containers in a CDB data store. At the same time, an additional requirement was identified to test and identify best practices for moving CDB vector files stored as Shapefiles into one or more GeoPackages.

In 2019, the CDB SWG executed the CDB Vector Data in GeoPackage Interoperability Experiment (IE). The participants in this IE tested transforming CDB Shapefile vector data into one or more GeoPackage(s) and storing the result in a CDB data store. GeoPackage Version 1.2 and CDB Version 1.1 and related Best Practices were the standards baseline used for this experiment. The IE built on the work described in the OGC CDB, Leveraging the GeoPackage Discussion Paper. A primary objective of the IE was to agree and document possible change requests and/or best practices for storing vector data in a CDB data using encodings and/or containers other than Shapefiles.

**OGC Patent Declaration**

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

## iv. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium (OGC):

Organization name(s)

## v. Submitters

- Carl Reed & Associates
- FlightSafety Visual Systems
- CAE Inc.

All questions regarding this submission should be directed to the editor or the submitters:

| **Name** | **Affiliation** |
|---|---|
| Carl Reed *(editor)* | Carl Reed & Associates |
| Ryan Franz | FlightSafety Visual Systems |
| David Graham | CAE Inc. |

# Chapter 1. Scope

This optional OGC CDB Extension defines the behavior and requirements for encoding vector data in a GeoPackage container for use in a CDB data store. The requirements and related guidance are grounded in the existing CDB Core requirements and the GeoPackage core requirements for vector data. As such, any GeoPackage that is to be referenced/used in a CDB data store must be 1.) compliant with the CDB core requirements for vector data and 2.) compliant with the GeoPackage core requirements for encoding vector data. Please note that some of the core GeoPackage requirements are profiled in order to be consistent with the CDB core requirements as specified in Volume 1: OGC CDB Core Standard: Model and Physical Data Store Structure. Further, there are associated normative additions to the CDB Core to accommodate not just the use of GeoPackages but also other vector storage encodings/containers.

Before reading this standard, please remember that the idea is to restrict the encoding of a dataset to a single vector format per CDB Version. Since a “CDB” is made of one or more “Versions” (as specified by Configuration.xml), and that each CDB Version can have a different encoding for a given dataset, the result is that a “CDB” may pretty well exist with multiple encodings for the same dataset. This means that if you wish to use GeoPackage containers, you need to create a version that will just contain GeoPackages of the vector data and not include any Shapefiles.

The following is a list of all of the CDB Standards and Best Practices documents;

- Volume 0: OGC CDB Companion Primer for the CDB standard (Best Practice).
- Volume 1: OGC CDB Core Standard: Model and Physical Data Store Structure. The main body (core) of the CDB standard (Normative).
- Volume 2: OGC CDB Core Model and Physical Structure Annexes (Best Practice).
- Volume 3: OGC CDB Terms and Definitions (Normative).
- Volume 4: OGC CDB Rules for Encoding CDB Vector Data using Shapefiles (Best Practice).
- Volume 5: OGC CDB Radar Cross Section (RCS) Models (Best Practice).
- Volume 6: OGC CDB Rules for Encoding CDB Models using OpenFlight (Best Practice).
- Volume 7: OGC CDB Data Model Guidance (Best Practice).
- Volume 8: OGC CDB Spatial Reference System Guidance (Best Practice).
- Volume 9: OGC CDB Schema Package: http://schemas.opengis.net/cdb/ provides the normative schemas for key features types required in the synthetic modelling environment. Essentially, these schemas are designed to enable semantic interoperability within the simulation context (Normative).
- Volume 10: OGC CDB Implementation Guidance (Best Practice).
- Volume 11: OGC CDB Core Standard Conceptual Model (Normative).
- Volume 12: OGC CDB Navaids Attribution and Navaids Attribution Enumeration Values (Best Practice).
- Volume 13: OGC CDB Rules for Encoding CDB Vector Data using GeoPackage (Normative, Optional Extension).

- Volume 14: OGC CDB Guidance on Conversion of CDB Shapefiles into CDB GeoPackages (Best Practice).
- Volume 15: OGC CDB Optional Multi-Spectral Imagery Extension (Normative).

# Chapter 2. Conformance

This standard defines [TBD] requirements / conformance classes.

The standardization targets of all conformance classes are "GeoPackages in CDB Data Store".

The main requirements class is:

```
Core. <<Add link to Annex A later>>
```

The Core specifies requirements that shall be implemented for all GeoPackages that are to be stored in a CDB store.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site.

In order to conform to this OGC® interface standard, a software implementation shall choose to implement the conformance levels specified in Annex A (normative)

All requirements-classes and conformance-classes described in this document are owned by the standard(s) identified.

# Chapter 3. References

The following normative documents contain provisions that, through reference in this text, constitute provisions of this document. For dated references, subsequent amendments to, or revisions of, any of these publications do not apply. For undated references, the latest edition of the normative document referred to applies.

ISO / TC 211: ISO 19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals (2014)

OGC: OGC 15-113r5, Volume 1: OGC CDB Core Standard: Model and Physical Data Store Structure (2017)

OGC: OGC 16-070r3, Volume 4: OGC CDB Best Practice use of Shapefiles for Vector Data Storage (2017)

OGC: OGC 12-128r15, OGC GeoPackage Encoding Standard - with Corrigendum (2018)

# Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply.

## 4.1. coordinate reference system

coordinate system that is related to the real world by a datum [ISO 19111]

## 4.2. coordinate system

set of mathematical rules for specifying how coordinates are to be assigned to points [ISO 19111]

## 4.3. dataset

collection of data, published or curated by a single agent, and available for access or download in one or more formats [DCAT]

| | |
|---|---|
| **NOTE** | The use of 'collection' in the definition from [DCAT] is broader than the use of the term collection in this standard. See the definition of feature collection. |

## 4.4. feature

abstraction of real world phenomena [ISO 19101-1:2014]

| | |
|---|---|
| **NOTE** | If you are unfamiliar with the term 'feature', the explanations on Spatial Things, Features and Geometry in the W3C/OGC Spatial Data on the Web Best Practice document provide more detail. |

## 4.5. feature collection; collection

a set of **features** from a **dataset**

| | |
|---|---|
| **NOTE** | In this standard, 'collection' is used as a synonym for 'feature collection'. This is done to make this document easier to understand for those that are not geo-experts. |

## 4.6. Height

Distance of a point from a chosen reference surface measured upward along a line perpendicular to that surface. [ISO 19111] Note 1 to entry: A height below the reference surface will have a negative value, which would embrace both gravity-related heights and ellipsoidal heights.

## 4.7. linestring

curve composed of straight-line segments (aka line) [ISO 19136:2007]

### 4.7.1. point

0-dimensional geometric primitive, representing a position [ISO 19136:2007]

Note to entry: The boundary of a point is the empty set.

## 4.8. polygon

planar surface defined by 1 exterior boundary and 0 or more interior boundaries [ISO 19136:2007]

## 4.9. GeoPackage file

a platform-independent SQLite database file that contains GeoPackage data and metadata tables with specified definitions, integrity assertions, format limitations and content constraints.

## 4.10. tile

a geometric shape with known properties that is the result of the tiling (tessellation) of a plane. A tile consists of a single connected "piece" without "holes" or "lines" (topological disc).

## 4.11. Valid GeoPackage

A GeoPackage that contains features per clause Features and/or tiles per clause Tiles and row(s) in the gpkg_contents table with data_type column values of "features" and/or "tiles" describing the user data tables.

## 4.12. vector and vector data

quantity having direction as well as magnitude

Note to entry: A directed line segment represents a vector if the length and direction of the line segment are equal to the magnitude and direction of the vector. The term vector data refers to data that represents the spatial configuration of features as a set of directed line segments. [ISO 19123:2005]

## 4.13. vector geometry

representation of geometry through the use of constructive geometric primitives [ISO 19107:2003]

## 4.14. Version

A collection of pure CDB Datasets and/or user-defined datasets. Please see section 3.2 of Volume 1:

OGC CDB Core Standard: Model and Physical Data Store Structure for details on the CDB versioning strategy and structure.

## 4.15. Abbreviations

**GPKG** GeoPackage

# Chapter 5. Conventions

This sections provides details and examples for any conventions used in the document. Examples of conventions are symbols, abbreviations, use of XML schema, or special notes regarding how to read the document.

## 5.1. Identifiers

Identifiers

The normative provisions in this standard are denoted by the URI http://www.opengis.net/spec/cdb-vector-geopackage/1.2

All requirements and conformance tests that appear in this document are denoted by partial URIs which are relative to this base.

# Chapter 6. Informative

This section provides information for understanding the role of using GeoPackage containers to encode vector data in a CDB data store. One key and overriding requirement to consider when implementing this CDB extension is that the structure if the content of the vector data in GeoPackage is also dictated by the core requirements in the CDB Standard. Examples would be for restrictions on the coordinate reference system (CRS) allowed and the size (extents) of the tiles.

## 6.1. What is a GeoPackage?

A GeoPackage is the SQLite container and the OGC GeoPackage Encoding Standard governs the rules and requirements of content stored in a GeoPackage container. The GeoPackage Standard defines the schema for a GeoPackage, including table definitions, integrity assertions, format limitations, and content constraints. The required and supported content of a GeoPackage is entirely defined in the standard.

## 6.2. What is CDB (or a CDB)?

CDB is an open standard defining physical, logical, and conceptual models for a single, “versionable,” virtual representation of the earth. CDB structured data stores provide for a geospatial content and model definition repository that is plug-and-play interoperable between database authoring workstations. Moreover, a CDB structured data store can be used as a common online (or runtime) repository from which various simulator client devices can simultaneously retrieve and modify, in real-time, relevant information to perform their respective runtime simulation tasks (OGC CDB Standard, 2018).

## 6.3. Why use GeoPackages in a CDB?

The geometries supported by the GeoPackage standard are consistent with the OGC Simple Feature Standard. A GeoPackage is capable of storing feature geometries as Points, LineStrings, Polygons, MultiPoints, MultiLineStrings, MultiPolygons, and GeomCollection. A strength of the GeoPackage is the portability and utility in non-traditional simulation environments, such as hand-held tactical devices. These strengths and the self describing nature of a GeoPackage created demand not just in the traditional modelling and simulation community but also other domains for having the ability to use GeoPackages in a CDB data store.

## 6.4. Geometry and Geometry Types

When working with vector feature data, one of the properties is the feature’s geometry. A geometry is an ordered sequence of vertices that are connected by straight line segments or circular arcs. The semantics of the geometry are determined by its type. The GeoPackage core supports six geometry types. Other types may be used but their definition and use requires development of a GeoPackage extension. The GeoPackage in CDB standard is based on the core geometry types and does not require any extensions to support additional geometries.

Further, the GeoPackage standard allows the additional specification of "Z" values and "M" values. Z

values may be thought of as "elevation". However, the use of "elevation" is inexact. Instead this standard uses the term *height* to define the semantics of the Z values. "M" stands for *measurement*. Measurements could be such properties as "temperature" or "reflectance". Whether Z and or M values are part of a given GeoPackage is determined by what values are set for the *z* and *m* columns in the GeoPackage Geometry Columns Table Definition. These are optional values.

Geometries are stored as binary blobs in a GeoPackage. Please read GeoPackage clause 2.1.3. Geometry Encoding carefully. This clause defines the binary encoding used and references OGC/ISO Well Known Binary Types (WKB) OGC WKB simple feature geometry types specified in OGC 06-103r4 are a subset of the ISO WKB geometry types. WKB geometry types are are restricted to 0, 1 and 2-dimensional geometric objects that exist in 2, 3 or 4-dimensional coordinate space. They are not geographic or geodesic geometry types. The axis order in WKB is always (x,y{,z}{,m}) where x is easting or longitude, y is northing or latitude, z is optional elevation and m is optional measure.

The following table provides a summary of the GeoPackage geometry types in relation to Shapefile geometries. As can be seen, the GeoPackage core set of geometry types is 100 percent compatible with the Shapefile geometry types.

| **GeoPackage Geometry Type** | **Shapefile Vector Type** |
|---|---|
| Point | Point |
| Linestring | Polyline |
| Polygon | Polygon |
| MultiPoint | MultiPoint |
| Point with "z" column set | PointZ |
| Linestring with "z" column set | PolylineZ |
| Polygon with "z" column set | PolygonZ |
| Linestring with "m" column set | PolylineM |
| Polygon with "M" column set | PolygonM |
| MultiPoint with "m" column set | MultiPointM |

## 6.5. Some Identifed Use Cases for Shapefile to GeoPackage

While previous work focused on the process and requirements for transforming CDB structured vector Shapefiles into GeoPackages, the identified use cases are apropos when defining how to store and access any other vector/geometry content regardless of encoding or format. This has to do with how the data are physically structured in a GeoPackage container.

However, vector data encodings and containers developed outside a CDB data transformation process would also require definition of additional requirements. For example, a GeoJSON encoding of feature data would require a transformation to "break" the GeoJSON file into the appropriate tile and LoD structure in order to be consistent with the CDB data store and structure requirements. The same would be true of vector data provided as none CDB structured Shapefiles, CityGML encodings, Oracle Spatial structured data, Esri GeoDataBase files and so on.

Both the OGC OGC CDB - Leveraging GeoPackage Discussion Paper and the OGC CDB Vector Data in GeoPackage Interoperability Experiment Engineering Report focused on a constrained set of well defined use cases related to the current CDB structure and how to best use GeoPackage for resolving each of the use cases. As such, dealing with non-CDB structured vector content was not addressed.

The four initial use cases identified in the OGC Discussion Paper are:

1. Replace each Shapefile with a GeoPackage: The easiest way to integrate a GeoPackage container into a CDB data store is to replace each Shapefile in a CDB data store with a GeoPackage.
2. Make each CDB tile a layer in a single GeoPackage: Constructing each vector tile within CDB as a table within a GeoPackage for a given CDB dataset is a straightforward approach to utilize GeoPackage capabilities and significantly reduce file counts in a CDB (note that in GeoPackage a table is known as a layer).
3. Store each CDB LOD as a layer in GeoPackage: Design approach #3 incorporates the lessons learn from experimentation with approach #2 to limit the number of tables within a GeoPackage and reduce the number of files in a CDB.
4. Store each CDB Geocell as a layer in GeoPackage: Design approach #4 extends design approach #3 to have a single GeoPackage per geotile of CDB. In this approach, the tables in the GeoPackage correspond to each data store of CDB (such as road networks, geospecific points, etc.).

In all cases, the experimentation focused on just vector data encoding as Shapefiles.

The research and tests performed and reported in the Discussion Paper were refined and provided the basis for the work done in the OGC GeoPackage in CDB Interoperability Experiment. In the interoperability experiment, Use Case 1 above was further refined into 4 options - each a possible approach for one to one conversions. All options were tested and reported on by the particpants. Many of the lessons learned, independent of the IE focus on Shapefiles already in a CDB, are germane to any process or workflow in which data are stored in a GeoPackage container that is to then be stored in a CDB data store.

If you wish to transform CDB structured Shapefiles into CDB structured GeoPackages, please refer to the Volume 14: OGC CDB Guidance on Conversion of CDB Shapefiles into CDB GeoPackages (Best Practice) *add URL and description when document is available*.

# Chapter 7. Requirements for Using GeoPackages in a CDB Data Store

This section documents the mandatory requirements for having GeoPackage containers in a structured CDB data store. Many of the requirements are *by reference* to specific requirements in both the GeoPackage and CDB standards. The referenced requirements are known as *dependent* requirements. For example, the GeoPackage standrds specifies that any valid GeoPackage shall have a file extension of .gpkg. The optional CDB GeoPackage extension specifies the same requirement.

## 7.1. General requirements

### 7.1.1. Requirement that one and only vector format can be in a CDB version

The following requirement clearly states that any given version has only one allowed vector format. Currently, the two supported formats are the traditional Shapefiles and now GeoPackages. However, a CDB data store may have multiple versions. Therefore, the Shapefile format could be used in one version and GeoPackages used in another version.

| | |
|---|---|
| **Requirement 1**<br>Vector Format by Version | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-core<br><br>Any version in a CDB data store *SHALL* contain one and only one vector data format. |

There is a related CDB recommendation that a feature should not have two representations. This means that a feature should not be a point at one LoD and an polygon in another LoD.

### 7.1.2. Requirement on geometry types and feature instances

The following requirement clarifies the geometry types allowed for instances of feature data referenced to a tile. Essentially, all instances of a specific feature type (code?) SHALL be of the same geometry type.

| | |
|---|---|
| **Requirement 2**<br>Vector GeoPackage Geometry Type | http://www.opengis.net/spec/cdb/1.2/geopackage/vector-geom-rule<br><br>All instances of a given feature code *SHALL* be of the same geometry type. While the GeoPackage model supports encoding of 8 different types that can be stored in the same GeoPackage, the CDB standard requires a maximum of one geometry type for point features, a maximum of one geometry type for lineal features and a maximum of one geometry type for polygon features for each tile (for a maximum of 3 feature geometry types per tile). |

### 7.1.3. Literal case including GeoPackage file names

To ensure interoperability, file names and literal strings need to follow the case rules identified in

both the CDB and GeoPackage Standards. The following requirement speciffies the literal case rules.

| **Requirement 12**<br>Tiled Vector Datasets Attribution | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-gpkg-literal-case<br><br>Implementations *SHALL* support the literal case rules as specified in the CDB standard. CDB file naming conventions are case sensitive. Further, regardless of case, any name such as “house” *SHALL* have the same semantic meaning. Additionally, any name *SHALL* have its first 10 characters as unique. Finally, the GeoPackage file extension .gpkg *SHALL* always be lower case. |
|---|---|

# 7.2. GeoPackage Requirements

This section defines the GeoPackage requirements that need to be considered when implementing a CDB compliant GeoPackage for use in a CDB data store. This section also provides any clarifications and/or profiles of those requirements.

## 7.2.1. GeoPackage Requirements - core

The following requirement captures all of the core GeoPackage requirements that need to be implemented in order to be a fully compliant GeoPackage for use in a CDB data store. Please note that GeoPackage Requirements 10 and 11 on Coordiante Reference Systems has been profiled to be consistent with the mandatory requirements in the CDB standard.

| **Requirement 3**<br>GeoPackage Core | http://www.opengis.net/spec/cdb/1.2/geopackage/geopackage-core<br><br>Any CDB structured GeoPackage *SHALL* be compliant with GeoPackage Requirements 1 through 16 inclusive. Please see Requirement 4 of this standard for a restriction (profile) on GeoPackage Requirements 10 and 11 - Spatial Reference Systems (aka coordinate reference systems in CDB). |
|---|---|

## 7.2.2. GeoPackage Requirements constrained by CDB - CRS Profile

The following requirement clarifies (profiles) GeoPackage requirements 10 and 11 for specifying a Coordinate Reference System (CRS). The GeoPackage standard uses the term Spatial Reference System instead of Coordinate Reference System. This is because GeoPackage standard was designed to be flexible and to be able to accommodate any reference system. As such, GeoPackage tables that deal with CRS us the "spatial_ref" literal as part of the table name.

| | |
|---|---|
| **Requirement 4** CDB - GeoPackage Core CRS | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-geopackage-core-crs<br><br>As per CDB Requirement 8, any CDB structured GeoPackage *SHALL* specify geographic locations using WGS-84 (World Geodetic System 1984), equivalent to EPSG (European Petroleum Survey Group) code 4326 (2 dimensions) or EPSG code 4979 (3 dimensions). If a geographic location also has an altitude, the altitude *SHALL* be expressed relative to the WGS-84 reference ellipsoid. |

Please also note the definition of Volume 1 CDB Core Requirement 7 Units of Measure and Requirement 111 Vextex CRS for guidance on proper and compliant use of CRS in a CDB compliant GeoPackage.

### 7.2.3. GeoPackage Requirements Options - Features

The following requirement clarifies the GeoPackage requirements that shall be followed when implementing vector fectures in a GeoPackage. Please note that GeoPackage geometry and feature definitions are consistent with the OGC Simple Features model and ISO 19107.

| | |
|---|---|
| **Requirement 5** GeoPackage Vector Features | http://www.opengis.net/spec/cdb/1.0/geopackage/geopackage-features<br><br>Any CDB structured GeoPackage that encodes features *SHALL* be compliant with GeoPackage Requirements 18 through 33 inclusive and GeoPackage Requirements 146 and 150. Please see Requirement 20 of the GeoPackage standard for additional clarification on vector feature geometry types. These requirements are included in `Clause 2.1 Features` of the GeoPackage Standard. |

### 7.2.4. Requirement CDB Core - Need to discuss this one. Could be changed to a backend process and not a client.

The following requirement defines how vector data in a GeoPackage is to be structured based on requirements specified in the CDB Core document.

| | |
|---|---|
| **Requirement 6** GeoPackage polygon readers | http://www.opengis.net/spec/cdb/1.2/geopackage/polygon-rules-reader<br><br>GeoPackage readers *SHALL* handle the following cases with proper error handling and reporting for Polygon geometries:<br><br>* Has no self-intersections or co-linear segments<br><br>* Has no identical consecutive points (no zero-length segments)<br><br>* Does not degenerate into zero-area parts<br><br>* Does not have clock-wise inner rings (“Dirty Polygon”)<br><br>NOTE: This requirement may change in a future version. These cases may be handled as rules for data preparation. |

## 7.3. CDB Requirements for a CDB compliant GeoPackage Vector encoding

This section documents the CDB requirements that need to be considered when implementing a CDB compliant GeoPackage for use in a CDB data store. This section also provides any clarifications and/or profiles of those requirements. The majority of the requirements referenced in this section are detailed in Volume 1: OGC CDB Core Standard: Model and Physical Data Store Structure (hereafter known as CDB Core). In the GeoPackage in CDB standard, requirements are specified that themselves reference specific requirements in the Core

### 7.3.1. CDB Vector Data Core Requirements

The following requirement profiles the CDB general data representation core requirements class: Requirements Class - General Data Representation (Requirements 6-10). Please note that the profile excludes the raster imagery compression requirement. The related CDB conformance class documented in Annex A of the CDB Core standard is A.1.1 General CDB Data Store and Implementation.

| | |
|---|---|
| **Requirement** 7 CDB general data requirements for GeoPackage: Profile | http://www.opengis.net/spec/cdb/1.0/geopackage/cdb-geopackage-data<br><br>Any CDB structured GeoPackage that encodes vector features *SHALL* be compliant with CDB Requirements 7 through 10. These requirements are documented in CDB Requirements Class **General Data Representation Requirements**. Please note that Requirement 6 image compression is excluded. |

### 7.3.2. CDB Tiling and LoD Rules for Structuring GeoPackages

The following requirement references the CDB requirements for the rules for tiling the vector data in a GeoPackage and determination of the level of detail (LoD) in which the GeoPackage should be stored. Complete details for the CDB tiling and LoD storage model are found in CDB Volume 1 and clause "CDB Concepts".

| | |
|---|---|
| **Requirement 8** CDB Tiles/Geocells and LoD relationships for GeoPackage | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-geopackage-tiles-and-lods<br><br>Any CDB structured GeoPackage that encodes vector features *SHALL* be compliant with CDB Tiles/Geocells and LoD Requirements 11 through 16 inclusive and CDB Requirement 41. These requirements are documented in CDB Requirements Class **Tiles/Geocells and LoD relationships (11-16 and 41)**. |

### 7.3.3. GeoPackage in CDB - Tiled Vector Data Sets

Clause 3.6 - Tiled Data Sets specifies the requirements and provides detailed supporting information on the tiling structure and LoD structure for tiled vector data. Note: All vector data sets are tiled in a CDB structured data store. The level-of-detail organization for GeoPackage vector datasets mimics the concept of map scaling commonly found in cartography. Please especially note Requirement 64 in the Core. Also pay careful attendtion to the tables and equations for calculating the latitude and longitude elements of the file names for the GeoPackages.

| | |
|---|---|
| **Requirement 9** vector tiled data | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-core-vector-tiled-data<br><br>Any CDB structured GeoPackage that encodes vector features *SHALL* be compliant with CDB Core Tiled Data Requirements 64 through 67 inclusive. These requirements are documented in the CDB Core Requirements Class Tiled Datasets (64-67). |

### 7.3.4. CDB GeoPackage Vector Datasets

#### CDB Vector data general requirements

This clause references requirements in section 5.7 Vector Datasets in the Volume 1 CDB Core Model Standard. The OGC suggests that clause 5.7 of the Core CDB standard be read in its entirety prior to developing capabilities to transform any non-CDB data source into a CDB structured GeoPackage that meets the mandatory requirements and would pass the CDB compliance tests.

From Volume 1: *All of the information that is needed to instance features is organized in accordance to the CDB tile structure. All the tiled Vector dataset files are located in the same directory. The dataset's second component selector (CS2) is used to differentiate between files with the same extension or with the same Vector features.*

Further, the developer also needs to understand the feature coding system used in a CDB data store.

From Volume 1: *The Vector dataset concept and the feature concepts overlap somewhat; some of the Vector datasets are generalizations or specializations of feature codes. Section 1.5 CDB Data Dictionary provides a recommended mapping of the feature attributes across the CDB compliant datasets. Note that the same feature should not have two representations.* More specifically, visit section 3.3.8.1 Feature Classification for more details. Basically, for the file path and naming requirements to be properly implemented, the correct feature codes need to be used. See http://schemas.opengis.net/cdb/1.1/Feature_Data_Dictionary.xml and http://schemas.opengis.net/cdb/1.1/Feature_Data_Dictionary.xsd for a complete list of feature codes. The .xml file is actually located in http://schemas.opengis.net/cdb/cdb-1_1_0.zip .

The first set of tiled vector data requirements relates to coordinates for lights and the structure of vertex coordinates.

| **Requirement 10**<br>Tiled Vector Datasets | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-core-tiled-vector-datasets<br><br>Any CDB structured GeoPackage that encodes vector features SHALL be compliant with CDB Core Tiled Vector Datasets Requirements 107 through 111 inclusive. These requirements are documented in the CDB Core Requirements Class Tiled Vector Datasets (107-111). |
|---|---|

**CDB Attribution**

Attributes are used to describe one or more real or virtual characteristics of a feature. Features can be assigned a variable number of attributes. The following requirements from the core document and related informative discussion in Volume 1 Core describe the CDB compliant usage of attributes in a CDB datastore.

Important Note: Each attribute is uniquely defined by an attribute identifier. In database terminoly this would be a column name or heading in a database table. This attribute identifier is a “case-sensitive” character string of 10 characters or less. Further, the 10 character must be a unique literal string so that all columns in a table have unique names.

| **Requirement 11**<br>Tiled Vector Datasets Attribution | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-core-tiled-vector-datasets-attribution<br><br>Any CDB structured GeoPackage that encodes vector features *SHALL* be compliant with CDB Core Tiled Vector Datasets Requirements 112 through 116 inclusive. These requirements are documented in the CDB Core Requirements Class 5.7.1.2 CDB Attribution (112-116). |
|---|---|

**Topology**

If the vector data layer is topologically structures, such as road networks, then the following requirement SHALL apply for structuring these data layers in a CDB GeoPackage container. Please reference 5.7.1.6 Handling of Topological Networks in Volume 1 for detailed information on this set

of requirements.

| | |
|---|---|
| **Requirement 13**<br>Tiled Vector Topological Networks | http://www.opengis.net/spec/cdb/1.2/geopackage/cdb-gpkg-topology<br><br>Any CDB structured GeoPackage that encodes topologically structured vector data layers, such as roads, railroads, or hydrology, *SHALL* implement CDB Requirements 117 through 119 inclusive. |

# Annex A: Conformance Class Abstract Test Suite (Normative)

## A.1. Conformance Test Class: OGC CDB Core Standard

This section describes conformance tests for the the optional extension to the CDB standard for structuring and storing any vector data as a GeoPackage container. These abstract test cases describe the conformance criteria for verifying the structure and content of any data store or database claiming conformance to the CDB 1.2 standard.

The conformance class base id is “http://www.opengis.net/spec/cdb/1.2/” and all of the other conformance tests URLs are created in this path. Each conformace class then appends: "/conf/cdb-core/geopackage" to this base ID. Another issue that the reader should pay attention to is the test method. When the test method is assigned with “Visual”, it means that the purpose of the test should be “visually” investigate the file contents, image, or other content.

### A.1.1. Only one vector data encoding/format per version.

The following conformance class verifies that each "version" in a CDB data store contains one and only one vector encoding/format.

| | | |
|---|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage** | |
| **Requirements** | /req/geopackage/cdb-core | |
| **Dependency** | Volume 1: CDB Core - versioning. | |
| **Test 1** | **Test purpose** | Verify that any version in a CDB data store contains one and only one vector data format. |
| | **Test method** | Pass there is only one vector data type in a version. |
| | **Test type** | Conformance |

### A.1.2. All instances of a given feature code SHALL be of the same geometry type.

| | | |
|---|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage** | |
| **Requirements** | /req/geopackage/vector-geom-rule | |
| **Dependency** | Volume 1: CDB Core - geometry rule. | |
| **Test 1** | **Test purpose** | Verify that all instances of a given feature code are of the same geometry type. |

| | | |
|---|---|---|
| | **Test method** | Pass if there is only one geometry type for a given feature code. |
| | **Test type** | Conformance |

### A.1.3. Verify that implementations support the literal case rules as specified in the CDB standard

| | | |
|---|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage/cdb-gpkg-literal-case** | |
| **Requirements** | /req/geopackage/cdb-gpkg-literal-case | |
| **Dependency** | Volume 1: CDB Core - literal. | |
| **Test 12** | **Test purpose** | Verify that implementations support the literal case rules as specified in the CDB standard. Additionally, any name SHALL have its first 10 characters as unique. Finally, the GeoPackage file extension .gpkg SHALL always be lower case. |
| | **Test method** | Visual |
| | **Test type** | Conformance |

### A.1.4. Verify that GeoPackage Requirements 1 through 16 are implemented

| | | |
|---|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage/geopackage-core** | |
| **Requirements** | /req/geopackage/geopackage-core | |
| **Dependency** | GeoPackage Requirements 1 through 16 inclusive and GeoPackage ATS. | |
| **Test 3** | **Test purpose** | Any CDB structured GeoPackage SHALL be compliant with GeoPackage Requirements 1 through 16 inclusive. |
| | **Test method** | As per the GeoPackage Abstract Test Suite. |
| | **Test type** | Conformance |

### A.1.5. Verify that the CRS used is WGS-84 (2D or 3D)

| | |
|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage/cdb-geopackage-core-crs** |
| **Requirements** | /req/geopackage/cdb-geopackage-core-crs |

| | | |
|---|---|---|
| **Dependency** | Volume 1: CDB Core - CDB Requirement 8 and GeoPackage Requirements 10 and 11. | |
| **Test 4** | **Test purpose** | As per CDB Requirement 8, any CDB structured GeoPackage SHALL specify geographic locations using WGS-84 (World Geodetic System 1984), equivalent to EPSG (European Petroleum Survey Group) code 4326 (2 dimensions) or EPSG code 4979 (3 dimensions). If a geographic location also has an altitude, the altitude SHALL be expressed relative to the WGS-84 reference ellipsoid. |
| | **Test method** | Verify that the CRS used are correct. |
| | **Test type** | Conformance |

### A.1.6. Verify that GeoPackage Requirements 18 through 33 are implemented

| | | |
|---|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage/geopackage-features** | |
| **Requirements** | /req/geopackage/geopackage-features | |
| **Dependency** | GeoPackage Requirements 18 through 33 inclusive and GeoPackage Requirements 146 and 150 and the GeoPackage ATS. | |
| **Test 5** | **Test purpose** | Any CDB structured GeoPackage that encodes features SHALL be compliant with GeoPackage Requirements 18 through 33 inclusive and GeoPackage Requirements 146 and 150. |
| | **Test method** | As per the GeoPackage Abstract Test Suite. |
| | **Test type** | Conformance |

### A.1.7. Verify that polygon data is "clean"

| | |
|---|---|
| **Conformance Class** | **/conf/cdb-core/geopackage/polygon-rules-reader** |
| **Requirements** | /req/geopackage/polygon-rules-reader |
| **Dependency** | CDB Vector Data Requirement. |

| | | |
|---|---|---|
| **Test 6** | **Test purpose** | GeoPackage readers SHALL handle the error cases with proper error handling and reporting for Polygon geometries. |
| | **Test method** | Visual. |
| | **Test type** | Conformance |

# Annex B: GeoPackage and Shapefile Geometry types cross walk.

| Value | GeoPackage geometry type | Shapefile Vector type | Fields |
|---|---|---|---|
| 0 | | Null shape | None |
| 1 | Point | Point | X, Y |
| 3 | Linestring | Polyline | MBR, Number of parts, Number of points, Parts, Points |
| 5 | Polygon | Polygon | MBR, Number of parts, Number of points, Parts, Points |
| 8 | MultiPoint | MultiPoint | MBR, Number of points, Points |
| 11 | Point | PointZ | X, Y, Z *Optional*: M |
| 13 | Linestring | PolylineZ | *Mandatory*: MBR, Number of parts, Number of points, Parts, Points, Z range, Z array *Optional*: M range, M array |
| 15 | Polygon | PolygonZ | *Mandatory*: MBR, Number of parts, Number of points, Parts, Points, Z range, Z array *Optional*: M range, M array |
| 18 | Multipoint | MultiPointZ | *Mandatory*: MBR, Number of points, Points, Z range, Z array *Optional*: M range, M array |
| 21 | Point | PointM | X, Y, M |

| Value | GeoPackage geometry type | Shapefile Vector type | Fields |
|---|---|---|---|
| 23 | Linestring | PolylineM | *Mandatory*: MBR, Number of parts, Number of points, Parts, Points *Optional*: M range, M array |
| 25 | Polygon | PolygonM | *Mandatory*: MBR, Number of parts, Number of points, Parts, Points *Optional*: M range, M array |
| 28 | MultiPoint | MultiPointM | *Mandatory*: MBR, Number of points, Points *Optional Fields*: M range, M array |
| 31 | Surface | MultiPatch | *Mandatory*: MBR, Number of parts, Number of points, Parts, Part types, Points, Z range, Z array *Optional*: M range, M array |

# Annex C: Revision History

| Date | Release | Editor | Primary clauses modified | Description |
|---|---|---|---|---|
| 2019-11-06 | 1.2 | C. Reed | all | initial version |

# Annex D: Bibliography

**NOTE**

*Example Bibliography (Delete this note).*

The TC has approved Springer LNCS as the official document citation type.

Springer LNCS is widely used in technical and computer science journals and other publications

- For citations in the text please use square brackets and consecutive numbers: [1], [2], [3]

– Actual References:

[n] Journal: Author Surname, A.: Title. Publication Title. Volume number, Issue number, Pages Used (Year Published)

[n] Web: Author Surname, A.: Title, http://Website-Url

[1] OGC: OGC Testbed 12 Annex B: Architecture. (2015).