

OGC CDB Version 2.0 GeoPackage  
Profile for CDB 1.x Datastores  
Standard

# Table of Contents

1. Scope .....	5
2. Conformance .....	7
3. References .....	8
4. Terms and Definitions .....	9
4.1. <b>coordinate reference system</b> .....	9
4.2. <b>coordinate system</b> .....	9
4.3. <b>dataset</b> .....	9
4.4. <b>feature</b> .....	9
4.5. <b>feature collection; collection</b> .....	9
4.6. <b>Height</b> .....	9
4.7. <b>GeoPackage file</b> .....	10
4.8. <b>tile</b> .....	10
4.9. <b>Valid GeoPackage</b> .....	10
4.10. <b>vector and vector data</b> .....	10
4.11. <b>vector geometry</b> .....	10
4.12. <b>Version</b> .....	10
4.13. Abbreviations .....	10
5. Conventions .....	11
5.1. Identifiers .....	11
6. Overview and Background .....	12
6.1. What is a GeoPackage? .....	13
6.2. What is CDB (or a CDB)? .....	13
6.3. Why use GeoPackages in a CDB datastore? .....	13
6.4. CDB and GeoPackage Geometry and Geometry Types .....	13
7. Requirements for structuring GeoPackage Containers based on the CDB 2.0 Abstract Core, CDB 1.x, and GeoPackage .....	15
7.1. General requirements .....	15
7.2. Compliance with GeoPackage Standard .....	16
7.3. Requirements and restrictions based on the OGC CDB 1.2 and later Standard. ....	17
7.4. Restriction of CDB Version 2: core Folder Requirements Class .....	17
7.5. Restriction of CDB Version 2: Core CRS Requirements Class .....	18
7.6. Restriction of CDB Version 2: Core Geometry Requirements Class .....	19
7.7. Restriction of CDB 2 Metadata Requirements Class .....	20
7.8. Restriction of CDB Version 2: Core Tiling Requirements Class .....	22
7.9. Restriction of CDB Version 2: Core Topology Requirements Class .....	23
7.10. Restriction of CDB Version 2: Core Attributes Requirements Class .....	24
7.11. Literal restriction for file names etc. ....	24
8. Media Types .....	25

9. Security Considerations .....	26
9.1. Security related to CDB and CDB Resources .....	26
9.2. Security consideratuons related to GeoPackage .....	26
Annex A: Conformance Class Abstract Test Suite (Normative) .....	27
A.1. Conformance Class Vector Format, .....	27
A.2. Conformance: Compliance with CDB 2.0 CRS Requirements Class .....	27
A.3. Conformance: Compliance with GeoPackage Requirements Class .....	28
A.4. Conformance: Compliance with CDB 1.3 Requirements Class .....	28
A.5. Conformance: Compliance with CDB 2.0 Folders Requirements Class .....	28
A.6. Compliance with CDB 2.0 Geometry Requirements Class .....	29
A.7. Compliance with CDB 2.0 Metadata Requirements Class .....	29
A.8. Compliance with CDB 2.0 Tiling Requirements Class .....	30
A.9. Compliance with CDB 2.0 Topology Requirements Class .....	30
A.10. Compliance with CDB 2.0 Attributes Requirements Class .....	30
Annex B: Revision History .....	32

# Open Geospatial Consortium

Submission Date: <2023-05-20>

Approval Date: <yyyy-mm-dd>

Publication Date: <yyyy-mm-dd>

External identifier of this OGC® document: <http://www.opengis.net/doc/IS/ogccdb-gpkg/1.0>

Internal reference number of this OGC® document: 23-035

Version: 1.0.2-SNAPSHOT (Editor's draft)

Latest Published Version:

Category: Draft OGC® Implementation Standard

Editors: Carl Reed, PhD

## OGC CDB Version 2.0 Simple GeoPackage Profile

### Copyright notice

Copyright © 2023 Open Geospatial Consortium

To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>

### 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: Interface

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 NON-INFRINGEMENT 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.

## i. Abstract

The CDB Version 2.0 GeoPackage Profile for CDB 1.x Datastores Standard profiles CDB Version 2.0: Core requirements for creating and using OGC GeoPackage compliant storage containers in a CDB 1.2 and later datastore. Requirements specified in the GeoPackage Profile are restrictions of requirements specified in the CDB Version 2.0: Core Requirements Modules and Classes. Other requirements, such as for file naming and folder structure, are derived from CDB version 1.2 and later Requirements.

**NOTE** For the sake of brevity, in the remainder of this document, the title is shortened to **GeoPackage Profile**.

**NOTE** Implementation of this Standard requires knowledge of the CDB 1.2 requirements for vector data, tiled vector data, LoDs, attribution, file naming, and layers.

## ii. Keywords

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

ogcdoc, OGC document, CDB, GeoPackage, Profile

## iii. Preface

### Background of the CDB Version 2.0 GeoPackage Profile for CDB 1.x Datastores Standard

The original requirement for storing GeoPackage structured content in a CDB datastore is documented in [OGC Change Request 545](#). This OGC change request was submitted based on work performed in [OGC Testbed 13](#). That OGC Testbed activity and the 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.

The research and implementation work continued in May 2020 when [SOFWERX](#) announced the 3D Geospatial Series Tech Sprint II – OGC CDB 2.0. In support of Special Operations Forces (SOF) Future Concepts, this effort focused on accelerating evolution of the OGC CDB Standard to meet the needs of planning, rehearsal, and Mission Command systems providing decision support to SOF and enabling SOF tactical and operational advantage. Additionally, there was a desire to better align aspects of the CDB standard with the U.S. Army One World Terrain activity was expressed during the initial workshop. One of the five work items was to develop guidelines, rules, and requirements for storing CDB structured vector and coverage content in one or more GeoPackage compliant containers.

The results of these various activities was the definition and approval of the [OGC CDB Volume 13 19-054 CDB Vector Data in Geopackage Standard](#). This optional CDB Version 2.0 GeoPackage Profile is the next step in the evolution of using GeoPackage(s) in a CDB datastore.

**NOTE**

This Profile, based on CDB Version 2.0: Core requirements, is designed to be consistent and backwards compatible with CDB version 1.3 and later CDB compliant datastores. As such, this Standard is part of the CDB 1.2 and later family of CDB Standards.

**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 ( <i>editor</i> )	Carl Reed & Associates

# Chapter 1. Scope

This optional CDB Version 2.0 GeoPackage Profile for CDB 1.x Datastores Standard specifies the behavior and requirements for encoding vector data in a GeoPackage container for use in a CDB 1.x and later data store. The requirements and related guidance are grounded in the CDB 2.0 Fundamental Core Requirements Classes, CDB 1.3 Requirements for file naming and attribution, and the GeoPackage Standard core requirements for vector data. As such, any GeoPackage that is to be referenced/used in a CDB data store must be:

- Compliant with the CDB 2.0 Fundamental Core requirements as referenced.
- Compliant with CDB 1.3 Requirements as referenced.
- Compliant with the GeoPackage core requirements for encoding vector data as referenced.

Specifically, this GeoPackage Profile references and restricts the following CDB Version 2.0 Fundamental Core Requirements Classes.

Requirements Class	M/O	Description
<a href="#">Coordinate Reference System</a>	M	Requirements for the coordinate reference system of a CDB datastore.
<a href="#">File naming</a>	M	Requirements for naming assets in a CDB datastore.
<a href="#">Geometry</a>	M	Requirements for specifying geometry in a CDB datastore.
<a href="#">Links</a>	M	Specifies requirements for how links are structured.
<a href="#">Media Types</a>	O	Specifies enumerations of CDB 2.0 Media Types.
<a href="#">Metadata</a>	M	Specifies requirements classes and requirements for global and resource metadata used in a CDB datastore.
<a href="#">Tiling (Abstract)</a>	M	Specifies requirements classes and requirements for tiling a CDB datastore at the abstract level.
<a href="#">Tiling Extension: CDBGlobal Grid</a>	M	Requirements module for implementing the CDBGlobalGrid tiling extension.



<b>Requirements Class</b>	<b>M/O</b>	<b>Description</b>
Topology	O	Specifies requirements for topology primitives used in a CDB datastore.
Versioning	O	Specifies requirements for versioning used in a CDB datastore implementation.

# Chapter 2. Conformance

This document defines requirements classes, requirements and conformance classes applicable to the implementation of the CDB Version 2.0 GeoPackage Profile for CDB 1.x Datastores Standard.

The standardization targets of all conformance classes are "GeoPackages in a CDB 1.2 and later datastore grounded in CDB 2.0 Abstract Core Requirements".

The main requirements class is:

```
<<cdb2-gpkgap-normative,CDB-2-GeoPackage-AP>>.
```

The CDB-2-GeoPackage-AP requirements class specifies requirements that *SHALL* be implemented for all GeoPackages that are to be stored in a CDB 1.2 and later compliant datastore. The requirements defined in the GeoPackage Vector Data in a CDB 1.2 and later Datastore and a combination of profiles of the CDB 2.0 Abstract Core classes and CDB 1.3 and later requirements classes.

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.

In order to conform to this OGC® Standard, a software implementation SHALL implement the mandatory conformance class specified in Annex A (normative). Each Conformance Class has an associated Requirements Class. The Requirements Classes define the functional requirements which will be tested through the associated Conformance Class.

# 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.

- Open Geospatial Consortium (OGC). OGC 22-002: **OGC® Volume 1: CDB Core Standard: Model and Physical Data Store Structure version 1.3** [online]. Edited by Carl Reed. 2023 [viewed 2023-03-07]. Available at <to be added once approved>
- Open Geospatial Consortium (OGC). OGC 22-002: **OGC® CDB Version 2.0: Core Standard** [online]. Edited by Carl Reed. TBD [viewed 2023-01-07]. Available at <to be added once approved>
- Open Geospatial Consortium (OGC). OGC 2-128r18: **OGC® GeoPackage Encoding Standard** [online]. Edited by Jeff Yutzler. 2022 [viewed 2023-01-07]. Available at <https://www.geopackage.org/spec131/index.html>
- Internet Engineering Task Force (IETF). RFC 3339: **Date and Time on the Internet: Timestamps** [online]. Edited by G. Klyne, C. Newman. 2002 [viewed 2020-03-16]. Available at <https://www.rfc-editor.org/rfc/rfc3339.html>

# Chapter 4. Terms and Definitions

In addition to the terms defined below, 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.

## NOTE

Terms and definitions as defined in the CDB 2.0 Requirements classes listed below also apply to the normative clauses of this Standard.

## 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. 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.

## 4.8. 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).

## 4.9. 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.

## 4.10. 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]

## 4.11. vector geometry

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

## 4.12. 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.

## 4.13. Abbreviations

**GPKG** GeoPackage

# Chapter 5. Conventions

This section 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

The normative provisions in this standard are denoted by the URI namespace

```
http://www.opengis.net/spec/cdb/2.0/gpkgap
```

All requirements that appear in this document are denoted by partial URIs which are relative to the namespace shown above.

For the sake of brevity, the use of “req” in a requirement URI denotes:

```
http://www.opengis.net/spec/cdb/2.0/gpkgap/req
```

An example might be:

```
req/gpkgap/crs
```

All conformance tests that appear in this document are denoted by partial URIs which are relative to the namespace shown above.

For the sake of brevity, the use of “conf” in a requirement URI denotes:

```
http://www.opengis.net/spec/cdb/2.0/gpkgap/conf
```

For example, the conformance test for the Coordinate Reference System (CRS) Requirements class is

```
/conf/cdb/gpkgap/crs
```

# Chapter 6. Overview and Background

This Clause provides information for understanding the role of using GeoPackage containers to encode vector feature data in a CDB data store. One key and overriding requirement to consider when implementing this CDB Application Profile is that the structure if the content of the vector data in GeoPackage is also dictated by the requirements as specified in the CDB Version 2.0: Core Standard. Examples would be for restrictions related to the coordinate reference system (CRS) allowed and geometries supported.

The GeoPackage Profile is based on the results and recommendations of:

- The OGC [OGC CDB Vector Data in GeoPackage Interoperability Experiment](#) (2019).
- The SOCOM/SOFWERX [3D Geospatial Series Tech Sprint II – OGC CDB 2.0](#) (2020).
- The [OGC CBD Volume 13 CDB Rules for Encoding CDB Vector Data using GeoPackage Standard](#) (2020).

## NOTE

More specifically, this GeoPackage Profile is based on Option 1, sub-options C and D, as defined in the Vector Data in the GeoPackage Engineering Report.

Option 1 (Experiment 2 in the IE) focused on approaches to replacing each Shapefile with a corresponding GeoPackage in an existing CDB data store thereby consolidating the three geometry files into a single GeoPackage. Sub-options C and D added the additional constraints related to "flattening" the attribute schema and how to process attribute extensions. The goal was to ensure full compatibility with existing implementations of GeoPackage processing and visualization software. Therefore, this Application Profile defines a structure conceptually similar to the use of Shapefiles in CDB version 1.3 and earlier.

Characteristics of this approach are:

- There is a 4:1 to 7:1 reduction in the number of files.
- There is one layer (table) per GeoPackage.
- The Feature Class and Extended Attributes are populated for each feature.
- Utilizes a standard normalized relational database design, utilizing foreign keys.
- Table (related tables) for extended attributes
- This approach under-utilizes the capabilities of GeoPackage.

In order for existing GeoPackage viewers to consume and display CDB structured GeoPackages, the attribute schema needs to be "flattened". In terms of the CDB 1.x series of standards, this is defined in clause [7. Flattening the Schema](#). This issue was identified in the Interoperability Experiment.

Please note that there are other, more efficient approaches to to using GeoPackage for storing all geospatial data in a compliant CDB 2.0 datastore. These approaches may be defined in later versions of this Standard.

## 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 datastore?

The vector 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. CDB and GeoPackage Geometry and Geometry Types

When working with vector feature data, one of the properties is the feature’s geometry. At a basic level, a feature 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 as well as the concept of a geometry collection. Other types may be used but their definition and use requires use of a GeoPackage extension. This CDB 2.0 GeoPackage Application Profile is based on the GeoPackage core geometry types and does not require any extensions to support additional geometries. Reference Table 1 of the [CDB 2.0 Core Geometry Requirements Class](#).

Further, the OGC GeoPackage Standard supports 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 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.

# Chapter 7. Requirements for structuring GeoPackage Containers based on the CDB 2.0 Abstract Core, CDB 1.x, and GeoPackage

This section documents the mandatory requirements for having GeoPackage containers containing structured vector data in a CDB 1.2 and later compliant data store. Many of the requirements are *by reference* to specific requirements in both the GeoPackage and CDB 1.x standards. The referenced requirements are known as *dependent* requirements. For example, the GeoPackage Standard specifies that any valid GeoPackage *SHALL* have a file extension of **.gpkg**. The optional CDB Version 2.0 GeoPackage Profile for CDB 1.x Datastores Standard specifies the same requirement.

<b>Requirements Class - GeoPackage Vector Data in a CDB Datastore.</b>	
/req/core/data-representation	
Target type	Vector data in a CDB datastore
Dependency	<a href="#">Folder/File Structure and Hierarchy</a>
Dependency	<a href="#">CRS</a>
Dependency	<a href="#">Geometry</a>
Dependency	<a href="#">Tiling</a>
Dependency	<a href="#">Metadata</a>
Dependency	<a href="#">Topology</a>
Dependency	<a href="#">Attribution</a>
Requirement 1	<a href="#">/req/cdb2/gpkgap/vector-format</a>
Requirement 2	<a href="#">/req/cdb2/gpkgap/geopackage-requirements</a>
Requirement 3	<a href="#">/req/cdb2/gpkgap/cdb-1-x-requirements</a>
Requirement 4	<a href="#">/req/cdb2/gpkgap/folder-structure</a>
Requirement 5	<a href="#">/req/cdb2/gpkgap/crs</a>
Requirement 6	<a href="#">/req/cdb2/gpkgap/geometry</a>
Requirement 7	<a href="#">/req/cdb2/gpkgap/metadata</a>
Requirement 8	<a href="#">/req/cdb2/gpkgap/tiling</a>
Requirement 9	<a href="#">/req/cdb2/gpkgap/topology</a>
Requirement 10	<a href="#">/req/cdb2/gpkgap/attributes</a>
Requirement 11	<a href="#">/req/cdb2/gpkgap/literal</a>

## 7.1. General requirements

The following are the general requirements.

Requirement 1	/req/cdb2/gpkgap/vector-format
A	Any given version in a CDB data store <i>SHALL</i> contain one and only one vector data format.
B	All instances of a given feature code <i>SHALL</i> be of the same geometry type. While the GeoPackage model supports encoding of 8 different types that can be stored in the same GeoPackage, this CDB 2.0 Profile 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).

General requirement A states that any given version in a CDB datastore has only one allowed vector format. However, a CDB data store may have multiple versions. Each Version may have a different vector encoding/format but within any given version only a single vector encoding/format is allowed. This requirement is consistent and backwards compatible with CDB 1.x version data stores.

General requirement B is consistent and backwards compatible with CDB 1.x version data stores.

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.

## 7.2. Compliance with GeoPackage Standard

The following requirements states compliance of a GeoPackage in a CDB 1.x and later data store with the [GeoPackage Standard version 1.3.1](#).

Requirement 2	/req/cdb2/gpkgap/geopackage-requirements
A	Any CDB 2 and later structured GeoPackage <i>SHALL</i> be compliant with GeoPackage Requirements 1 through 16 inclusive. Please see Requirement CRS of this Standard for a restriction (profile) on GeoPackage Requirements 10 and 11 - Spatial Reference Systems (aka coordinate reference systems in CDB).
B	Any CDB structured GeoPackage that encodes features <i>SHALL</i> be compliant with . Please see Requirement 20 and <a href="#">Table 21</a> of the GeoPackage Standard for additional clarification on vector feature geometry types. These requirements are included in <a href="#">Clause 2.1 Features</a> of the GeoPackage Standard.

## 7.3. Requirements and restrictions based on the OGC CDB 1.2 and later Standard

The following are requirements and restrictions defined in the OGC CDB Standard, version 1.2 and later. These are additional requirements that are not defined in other requirements documented in this section. For example, CDB 1.x Requirement 7 states that all units of measure shall be in meters. This requirements is defined in the [Restriction of CDB Version 2: Core CRS Requirements Class](#) for height and [Restriction of Metadata Requirements Class](#) for storage of measurements such as length and distance.

Requirement 3	/req/cdb2/gpkgap/cdb-1-x-requirements
A	Any CDB structured GeoPackage that encodes vector features <i>SHALL</i> be compliant with CDB 1.2 (and later) Tiled Data Requirements 64 through 67 inclusive. These requirements are documented in the CDB Core Requirements Class Tiled Datasets <a href="#">64-67</a> .
B	Any CDB structured GeoPackage that encodes vector features <i>SHALL</i> be compliant with CDB 1.2 (and later) Tiled Vector Datasets Requirements 107 through 111 inclusive. These requirements are documented in the CDB Core Requirements Class Tiled Vector Datasets (107-111). NOTE: Some of these requirements are also specified in requirements defined in this section.

## 7.4. Restriction of CDB Version 2: core Folder Requirements Class

The following requirement restricts the CDB Version 2: Core cdb-core-file-structure.adoc[Folder] requirements class.

Requirement 4	/req/cdb2/gpkgap/folder-structure
A	Any CDB 2 and later structured GeoPackage <i>SHALL</i> be compliant with the Requirements as specified in the CDB Version 2: Core <a href="#">File Structure</a> Requirements Class with the following restrictions.
B	<a href="#">Restriction of CDB Core File Structure Recommendation File1</a> . The CDB data store <i>SHALL</i> have a root folder location (landing page) of /CDB.

C	All GeoPackages in a CDB 1.2 and later datastore <i>SHALL</i> be stored in a subfolder labeled /CDB/Tiles. This is consistent with the CDB version 1.x series of Standards .
---	--

## 7.5. Restriction of CDB Version 2: Core CRS Requirements Class

The following requirements restrict the CDB Version 2: Core [CRS Requirements Class](#). The following requirement covers requirements 7 through 10 in the CDB 1.2 or later Standard.

Requirement 5	/req/cdb2/gpkgap/crs
A	<a href="#">Restriction of CDB Version 2: Core Recommendation 1.</a> Coordinates in CDB <i>SHALL</i> be expressed using WGS-84 (World Geodetic System 1984), equivalent to EPSG (European Petroleum Survey Group) code 4326 (2 dimensions) and EPSG code 4979 (3 dimensions).
B	If a geographic location also has an altitude, height or depth, the altitude, height or depth <i>SHALL</i> be expressed relative to the WGS-84 reference ellipsoid.
C	<a href="#">Restriction of CDB Version 2: Core CRS Requirement 6.</a> Coordinates <i>SHALL</i> be described using the decimal degree format without the “°” symbol. The values of latitude and longitude <i>SHALL</i> be bounded by $\pm 90^\circ$ and $\pm 180^\circ$ respectively. Positive latitudes are north of the equator, negative latitudes are south of the equator. Positive longitudes are east of the Prime Meridian; negative longitudes are west of the Prime Meridian. The axis order <i>SHALL</i> be Latitude and longitude are expressed in that sequence.
D	<a href="#">Restriction of CDB Version 2: Core CRS Requirement VCRS 3.</a> Vertical units such as for altitude/height/elevations <i>SHALL</i> be meters.
E	<a href="#">Restriction of CDB Version 2: Core CRS Requirement 6.</a> The CRS Metadata (WKT 2 definition) <i>SHALL</i> be stored in the global metadata folder as defined in the <a href="#">Metadata restriction class</a> specified below.
F	The CRS metadata file <i>SHALL</i> be labeled <code>crs_metadata.json</code>

For information, the [OGC WKT 2](#) encoding for description of a compound WGS 84, EPSG 4326 and

EPSG 3855. (2 dimensions):

```
GEOGCRS["WGS 84",
  ENSEMBLE["World Geodetic System 1984 ensemble",
    MEMBER["World Geodetic System 1984 (Transit)"],
    MEMBER["World Geodetic System 1984 (G730)"],
    MEMBER["World Geodetic System 1984 (G873)"],
    MEMBER["World Geodetic System 1984 (G1150)"],
    MEMBER["World Geodetic System 1984 (G1674)"],
    MEMBER["World Geodetic System 1984 (G1762)"],
    MEMBER["World Geodetic System 1984 (G2139)"],
    ELLIPSOID["WGS 84",6378137,298.257223563,
      LENGTHUNIT["metre",1]],
    ENSEMBLEACCURACY[2.0]],
  PRIMEM["Greenwich",0,
    ANGLEUNIT["degree",0.0174532925199433]],
  CS[ellipsoidal,2],
    AXIS["geodetic latitude (Lat)",north,
      ORDER[1],
      ANGLEUNIT["degree",0.0174532925199433]],
    AXIS["geodetic longitude (Lon)",east,
      ORDER[2],
      ANGLEUNIT["degree",0.0174532925199433]],
  USAGE[
    SCOPE["Horizontal component of 3D system."],
    AREA["World."],
    BBOX[-90,-180,90,180]],
  ID["EPSG",4326]]
```

For the JSON encoding of EPSG 4326, please refer to [ANNEX B](#).

## 7.6. Restriction of CDB Version 2: Core Geometry Requirements Class

The following requirements restrict the CDB Version 2: Core [Geometry Requirements Class](#)

Requirement 6	<a href="/req/cdb2/gpkggap/geometry">/req/cdb2/gpkggap/geometry</a>
A	Any CDB 1.2 and later structured GeoPackage <i>SHALL</i> be compliant with the Requirements as specified in the CDB Version 2: Core <a href="#">Geometry Requirements Class</a> with the following restriction.

B	<p><b>Restriction of CDB Version 2: Core Geometry Requirement Geom2.</b> To be consistent with both CDB 1.2 and later GeoPackages and with GeoPackage Geometry types as specified in Table 30 of <a href="#">Annex G: Geometry Types</a> (Normative) in the GeoPackage Standard, only Geometry types 0 through 7 <i>SHALL</i> be used.</p>
---	--

**NOTE**

Restriction B makes the Geometry types specified in a CDB Version 2 and later GeoPackage consistent with the the GeoJSON Format [IETF RFC 7946](#)

## 7.7. Restriction of CDB 2 Metadata Requirements Class

The following requirements restrict the CDB Version 2: Core [Metadata Requirements Class](#)

Requirement 7	/req/cdb2/gpkgap/metadata
A	Any CDB datastore implementing this GeoPackage Profile <i>SHALL</i> be compliant with the Metadata Requirements as specified in the CDB Version 2: Core <a href="#">Metadata</a> Requirements Class with the following restrictions.
B	<b>Restriction of CDB Version 2: Core Metadata Requirement Metadata3</b> Global metadata <i>SHALL</i> be accessible from a folder labeled <code>metadata</code> .
C	<b>Restriction of CDB Version 2: Core Core File Structure Requirement File6.</b> The <code>metadata</code> folder <i>SHALL</i> be located as a sub-folder of the CDB datastore root folder <code>/CDB</code> . Example: <code>/CDB/metadata</code>
D	All metadata in the global <code>metadata</code> folder <i>SHALL</i> be encoded as JSON where each file <i>SHALL</i> have a <code>.json</code> extension. The exception is if the CRS metadata is encoded as a WKT2 file which which would then have a <code>.txt</code> file.
E	Information on the metadata standard used, the version of the CDB Standard, and the datastore UoM <i>SHALL</i> be encoded in a <code>version.json</code> file with a path (or link) of <code>/CDB/metadata/version.json</code> . See example below.
F	Global metadata elements <i>SHALL</i> be labled as shown in Table 1 below.
G	Global metadata content <i>SHALL</i> be stored in a file labled <code>global_metadata</code> in the <code>/CDB/metadata</code> folder.

An example CDB Version 2.0 version.json file that is compliant with the above requirements.

```
{
  "Version": {
    "PreviousIncrementalRootDirectory": "/CDB/root",
    "Comment": "For consistency with CDB 1.x versions, this version.json file contains
elements with the same meaning as in CDB version 1.2",
    "Specification": "version=2.0",
    "Metadata-standard": "DCAT",
    "uom": "M",
    "Extension": "CDB 2.0 GeoPackage Profile"
  }
}
```

The following Table 1 specifies the element (property) name, description, and whether that metadata element is mandatory.

Metadata element Name	Description	Mandatory or Optional
ID	A unique identifier for the entire CDB data store instance. This identifier is persistent and is considered global metadata. For example, this could be a <a href="#">Digital Object Identifier (DOI)</a> .	Mandatory
title	Title by which the datastore is known. For global metadata describing a CDB data store, this would be a name given to the entire data store. For example, “Yemen demonstration CDB data store.”	Mandatory
description	Detailed multi-line description to fully explain the datastore.	Mandatory
contactPoint	Name of the person, position, or organization responsible for the resource. This is a text string. An example of a point of contact could be “Flight Safety” or “CAE.”	Mandatory
created	A date which is used to help identify the datastore. For global metadata, this is the date that the CDB data store was created or issued. See Requirement Metadata6 above	Mandatory



language	The language and character set used for the datastore (if a language is used). NOTE: Please see <a href="#">IANA Language subtag registry</a> for language codes.	Mandatory
update	Date of last update to the CDB datastore. Note: Date gives values for year, month and day. Character encoding of a date is a string that <i>SHALL</i> follow the format for date as specified in CDB 2.0 Core Requirement <a href="#">Metadata6</a> .	Optional element
temporal	The temporal extent of the datastore. For a CDB datastore, this would be the temporal range of when the data store was initially created to the point where the most recent content was created.	Optional element
accessRights	Security restrictions on the access and use of the datastore. These would be constraints for an entire CDB data store. This could be information necessary to generate an EDH compliant encoding.	Optional element
license	A sub-class of all access constraints. These legal constraints include copyright, patent, patent pending, trademark, license, Intellectual Property Rights, restricted, and other. At the global level, these are legal constraints applicable to an entire CDB data store.	Optional element

## 7.8. Restriction of CDB Version 2: Core Tiling Requirements Class

The following requirements restrict the CDB Version 2: Core [Tiling Requirements Class](#). The following requirement covers requirements 11 through 16 and 41 in the CDB 1.3 Standard.

**NOTE** Implementation of the following requirements will create a CDB tiled datastore for vector (and image) data consistent with tiled content in the CDB 1.x series of standards.

Requirement 8	/req/cdb2/gpkgap/tiling
A	Any CDB implementation of the CDB GeoPackage Profile <i>SHALL</i> be consistent with and conform to all requirements specified in the <a href="#">Tiling Requirements Module (Abstract)</a> .
B	Any CDB implementation of the CDB GeoPackage Profile <i>SHALL</i> implement and conform with the <a href="#">CDB1 Global Grid extension</a> . Note: This is a restriction of the above Requirement. Sub-requirement <b>B</b> is consistent with and conforms to sub-requirement <b>A</b> .
C	Any CDB structured GeoPackage that encodes vector features <i>SHALL</i> be compliant with CDB 1.2 or later Tiles/Geocells and <a href="#">LoD Requirements 11 through 16</a> inclusive and CDB Requirement 41. These requirements are documented in CDB 1.2 or later Requirements Class Tiles/Geocells and LoD relationships (11-16 and 41).

**NOTE** The CDB1 Global Grid extension is defined in Annex H of the [OGC Two Dimensional Tile Matrix Set and Tile Set Metadata Standard](#).

## 7.9. Restriction of CDB Version 2: Core Topology Requirements Class

The following requirements restrict the CDB Version 2: Core [Topology Requirements Class](#)

Requirement 9	/req/cdb2/gpkgap/topology
A	Any CDB 1.2 and later structured GeoPackage that requires topology <i>SHALL</i> be compliant with the Requirements as specified in the CDB Version 2: Core <a href="#">Topology</a> Requirements Class with the following restriction.
B	<b>Restriction of CDB Version 2: Core Geometry Requirement Face Topology - Winding Order.</b> All faces (polygons) <i>SHALL</i> be encoded using a counterclockwise winding order for the "parent" polygon.

## 7.10. Restriction of CDB Version 2: Core Attributes Requirements Class

The following requirements restrict the CDB Version 2: Core [Attributes Requirements Class](#)

Requirement 10	/req/cdb2/gpkggap/attributes
A	Any CDB structured GeoPackage that encodes vector features <i>SHALL</i> be compliant with CDB Core Tiled Vector Datasets Requirements 112 through 116 inclusive. These requirements are documented in the CDB 1.2 and later Requirements Class CDB Attribution (112-116).
B	Any CDB structured GeoPackage that encodes topologically structured vector data layers, such as roads, railroads, or hydrology, <i>SHALL</i> implement CDB Requirements 117 through 119 inclusive.

## 7.11. Literal restriction for file names etc.

The following requirement is for backwards compatability with CDB versions 1.2 and later GeoPackages

Requirement 11	/req/cdb2/gpkggap/literal
A	Implementations <i>SHALL</i> support the literal case rules as specified in the CDB 1.2 and later standard.
B	Regardless of case, any name such as “house” <i>SHALL</i> have the same semantic meaning
C	Additionally, any name <i>SHALL</i> have its first 10 characters as unique.
D	The GeoPackage file extension .gpkg <i>SHALL</i> always be lower case.

# Chapter 8. Media Types

GeoPackage

- `application/geopackage+sqlite3` for GeoPackages. (<https://www.iana.org/assignments/media-types/media-types.xhtml>)

JSON media types that would typically be used in a server that supports JSON are:

- `application/json` for resources encoded as JSON, typically metadata.

XML media types that would typically occur in a server that supports XML are:

- `application/xml` for resources such as metadata.

# Chapter 9. Security Considerations

===

## 9.1. Security related to CDB and CDB Resources

Security considerations for CDB datastore implementations are in the domain of the implementing application, deployment platform, operating system, and networking environment.

Further, the CDB Standard specifies optional constraints on resource access and use (dct:accessRights, MD\_SecurityConstraints). These are security restrictions on the access and use of the resource. These would be constraints for an entire CDB data store. This could be information necessary to generate an Enterprise Data Header (EDH) compliant encoding.

The CDB Standard also specified optional constraints on resource access and use (dct:license, MD\_LegalConstraints): A sub-class of all access constraints. These legal constraints include copyright, patent, patent pending, trademark, license, Intellectual Property Rights, restricted, and other. At the global level, these are legal constraints applicable to an entire CDB data store.

### NOTE

If a community requires that such constraints are mandatory, then a profile of the CDB Standard should specify that these metadata elements are mandatory.

## 9.2. Security considerations related to GeoPackage

The GeoPackage Standard does not place any constraints on application, platform, operating system level, or network security.

Since GeoPackage is dependent on SQLite, implementors should monitor for security alerts related to SQLite and respond accordingly.

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

This Annex 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 2.0 standard.

The conformance class base id is “<http://www.opengis.net/spec/cdb/2.0/>” and all of the other conformance tests URLs are created in this path. Each conformance class then appends: “/conf/cdb/geopackage-ap” 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. Conformance Class Vector Format,

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

<b>Abstract Test 1</b>	<b><a href="#">/conf/gpkgap/vector-format</a></b>
Test Purpose	Validate that a CDB 2.0 datastore version has one and only one vector format.
Requirement	<a href="#">/req/cdb2/gpkgap/vector-format</a>
Test Method	<ol style="list-style-type: none"><li>1. Automated test to ensure all extensions end with <b>.gpkg</b>.</li><li>2. SQL Query to verify that all instances of a given feature code <i>SHALL</i> be of the same geometry type.</li></ol>

## A.2. Conformance: Compliance with CDB 2.0 CRS Requirements Class

<b>Abstract Test 2</b>	<b><a href="#">/conf/cdb/gpkgap/crs</a></b>
Test Purpose	Validate that all spatial geometries are in the CRS or CRS84h coordinate reference system unless otherwise requested by the client.
Requirement	<a href="#">/req/cdb2/gpkgap/crs</a>

Test Method	<ol style="list-style-type: none"> <li>1. SQL Query to determine that all coordinates are using WGS-84 (World Geodetic System 1984), equivalent to EPSG (European Petroleum Survey Group) code 4326 (2 dimensions) or EPSG code 4979 (3 dimensions).</li> <li>2. Validate retrieved spatial data using the CRS84 reference system (for 2D geometries) or the CRS84h reference system (for 3D geometries).</li> </ol>
-------------	--

### A.3. Conformance: Compliance with GeoPackage Requirements Class

<b>Abstract Test 3</b>	<b><a href="#">/conf/gpkgap/gpkg-compliance</a></b>
<b>Test Purpose</b>	Verify conformance with GeoPackage Requirements 1 through 16 inclusive, GeoPackage Requirements 18 through 33 inclusive, and GeoPackage Requirements 146 and 150, and GeoPackage ATS.
<b>Requirements</b>	<a href="#">/req/cdb2/gpkgap/geopackage-requirements</a>
<b>Test method</b>	As per the GeoPackage Abstract Test Suite.

### A.4. Conformance: Compliance with CDB 1.3 Requirements Class

<b>Abstract Test 4</b>	<b><a href="#">/conf/gpkgap/cdb-1-3</a></b>
<b>Test Purpose</b>	Validate that a CDB GeoPackage Application Profile implementation is compliant with CDB Core Tiled Data Requirements 64 through 67 inclusive and with CDB Core Tiled Vector Datasets Requirements 107 through 111 inclusive.
<b>Requirement</b>	<a href="#">/req/cdb2/gpkgap/cdb-1-3-requirements</a>
<b>Test Method</b>	As per the CDB 1.3 Abstract Test Suite

### A.5. Conformance: Compliance with CDB 2.0 Folders Requirements Class

<b>Abstract Test 5</b>	<b><a href="#">/conf/gpkgap/folder-structure</a></b>
------------------------	--

Test Purpose	Validate that a CDB GeoPackage Application Profile implementation is compliant with the Requirements as specified in the CDB 2 <a href="#">File Structure</a> Requirements Class with restrictions B and C as specified in the GeoPackage Application Profile.
Requirement	<a href="#">/req/cdb2/gpkgap/folder-structure</a>
Test Method	Visual inspection of the file system for the CDB datastore.

## A.6. Compliance with CDB 2.0 Geometry Requirements Class

<b>Abstract Test 6</b>	<b><a href="#">/conf/gpkgap/geometry</a></b>
Test Purpose	Validate that a CDB GeoPackage Application Profile implementation complies with the Geometry Requirements class and appropriate restrictions.
Requirement	<a href="#">/req/cdb2/gpkgap/geometry</a>
Test Method	SQL Query to determine geometry types used in an CDB structured GeoPackage.

## A.7. Compliance with CDB 2.0 Metadata Requirements Class

<b>Abstract Test 7</b>	<b><a href="#">/conf/gpkgap/metadata</a></b>
Test Purpose	Validate that a CDB GeoPackage Application Profile implementation complies with the <a href="#">Metadata Requirements Class</a> and appropriate restrictions.
Requirement	<a href="#">/req/cdb2/gpkgap/metadata</a>
Test Method	Visual inspection of content in <code>/cdb/global_metadata</code> folder for mandatory elements.



## A.8. Compliance with CDB 2.0 Tiling Requirements Class

Abstract Test 8	<a href="#">/conf/gpkgap/tiling</a>
Test Purpose	Validate that a CDB GeoPackage Application Profile implementation complies with the <a href="#">CDB 2.0,Abstract Tiling Requirements Class</a> , <a href="#">link../../cdb-tiling-CDB1GlobalGrid-extension.adoc</a> [CDB1 Global Grid extension] and <a href="#">appropriate restrictions</a> .
Requirement	<a href="#">/req/cdb2/gpkgap/tiling</a>
Test Method	Visual inspection of the folder hierarchy structure of the CDB datastore.

## A.9. Compliance with CDB 2.0 Topology Requirements Class

Abstract Test 9	<a href="#">/conf/gpkgap/topology</a>
Test Purpose	Validate that a CDB GeoPackage Application Profile implementation complies with the Requirements as specified in the CDB 2 Topology Requirements Class with the restriction for winding order.
Requirement	<a href="#">/req/cdb2/gpkgap/topology</a>
Test Method	SQL Query that returns a topologically structured vector dataset for display with nodes highlighted.

## A.10. Compliance with CDB 2.0 Attributes Requirements Class

Abstract Test 10	<a href="#">/conf/gpkgap/attributes</a>
Test Purpose	Validate that a CDB GeoPackage Application Profile implementation complies with the Requirements as specified in the CDB 2 Attribute Requirements Class as restricted by this Application Profile's Attributes requirements class.

Requirement	<a href="/req/cdb2/gpkgap/attributes">/req/cdb2/gpkgap/attributes</a>
Test Method	SQL Queries that check validity of attribution and compliance with the requirements class.

# Annex B: Revision History

<b>Date</b>	<b>Release</b>	<b>Editor</b>	<b>Primary clauses modified</b>	<b>Description</b>
2023-04-11	2.0	C. Reed	all	Initial version