

CDB X Conceptual Model with Prototyping Examples and Recommendations

Table of Contents

1. Subject	4
2. Executive Summary	5
2.1. Recommendations	5
2.1.1. Attribution	6
2.2. Document contributor contact points	6
2.3. Foreword	7
3. References	8
4. Terms and definitions	9
4.1. Abbreviated terms	11
5. Overview	13
6. CDB 2.0 - Background and History	14
6.1. What is CDB?	14
6.2. OGC CDB 1.x History	15
6.3. OGC CDB 2.0	16
6.4. Attribution - An OGC Testbed 13 CDB Activity in	16
7. CDB X Design Goals, Use Cases, and Conceptual/Logical Models	18
7.1. CDB X Design Goals and Objectives	18
7.2. Use Cases	18
7.3. Definition of Concepts	19
7.3.1. Foundation Data	19
7.3.2. Reference Systems	20
7.4. 3D Models	20
7.5. Attribution	20
7.6. Tiling/Coverages/Imagery	20
7.7. Vector Data	20
8. CDB-X Attribution	21
8.1. Current CDB 1.x Attribution Approach	21
8.2. CDB-X Discussion and Requirements for CDB-X Experimentation	22
8.2.1. Experimentation Backround	22
8.2.2. Some questions to be addressed during experimentation	22
8.2.3. Phase 3, Day 3	23
8.2.4. Phase 3, Day 4	23
8.2.5. Phase 3, Day 5	25
Plan for Experimentation in subsequent phases	28
9. CDB X All Vector Data in GeoPackage	29
9.1. General	29
9.2. Question to be answered	29
10. 3D Models and other 3D Content	30

10.1. Questions to be addressed	30
10.2. CDB X Requirements for 3D Model Components	31
10.3. Short and Medium Term Objectives	32
10.4. 3D Experiment Object Model	32
10.4.1. The data	32
10.4.2. Attribute changes explored	32
10.5. Description of 3D formats/encodings and References for this section:	33
10.5.1. OpenFlight	33
10.5.2. GL Transmission Format (glTF) 2.0	33
10.6. CDB OpenFlight and glTF Feature Analysis	34
11. Supporting more than CDB 1.X:	44
11.1. Original Prototyping Experiments - Keeping for now but will delete when this section is more mature.	44
12. CDB X Metadata and Integration	46
Summary of history of CDB X metadata discussions.	47
.1. Geospatial Metadata – Guidance	49
Phase 3 Consensus, Day 3	53
Phase 3, Day 4	54
Phase 3, Day 5	55
Plan for Experimentation in subsequent phases	56
13. Tiling and Tiled Coverages	57
14. CDB X Tiling Design Principles, Requirements, Tiled Data Conceptual Model and GeoPackage	58
14.1. General	58
14.1.1. OGC Core Tiling Conceptual and Logical Models for 2D Euclidean Space	58
14.1.2. OGC Two Dimensional Tile Matrix Set Standard	58
14.2. Design Objectives for CDB X Tiling and LoDs and Layers	59
14.3. Proposed Tiling Logical Model for CDB X	59
14.4. Proposed CDB X Tiling Structure	60
14.5. Additional CDB X Recommendations	61
14.5.1. Elevation min/max	61
14.5.2. Image Compression - JPEG	61
14.5.3. Materials	61
14.6. Findings from experiments	61
14.7. CDB X Tiling/GeoPackage Experiment #1 (Ecere)	62
14.7.1. Data and GeoPackage Structure	62
14.7.2. OGC API access demo	63
14.7.3. Visualization	63
14.7.4. Live Cesium JS / 3D Tiles demonstation	64
14.7.5. Next steps (this section will change as new content is provided)	64
14.8. FlightSafety GeoPackage Tiling Experiments	65

14.8.1. FlightSafety Experiment 1	65
14.8.2. FlightSafety Experiment 2	66
14.8.3. Observations	66
Annex A: Revision History	68
Annex B: Bibliography	69

Publication Date: YYYY-MM-DD

Approval Date: YYYY-MM-DD

Submission Date: YYYY-MM-DD

Reference number of this document: <Something - could be OGC XX-XXX>

Reference URL for this document: <http://www.opengis.net/doc/PER/CDB> X Study-ID

Category: OGC Public Engineering Report

Editors: David Graham, Carl Reed

Title: CDB X Conceptual Model with Prototyping Examples and Recommendations

OGC Public Engineering Report

COPYRIGHT

WARNING

This document is not an OGC Standard. This document is an OGC Public Engineering Report created as a deliverable in an OGC Interoperability Initiative and is not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, any OGC Public Engineering Report should not be referenced as required or mandatory technology in procurements. However, the discussions in this document could very well lead to the definition of an OGC Standard.

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.

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the

Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Chapter 1. Subject

This Engineering Report documents a CDB Experimental (CDB X) Conceptual Model as well as the results of CDB X rapid prototyping activities conducted during the 3D Geospatial Series Tech Sprint II - OGC CDB 2.0. This activity was performed in support of Special Operations Forces (SOF) Future Concepts. This effort hopes to accelerate evolution of the OGC CDB standard to meet the needs of planning, rehearsal, and Mission Command systems providing decision support to Special Operations Forces and enabling SOF tactical and operational advantage. OGC industry standards enable interoperability of geospatial data across systems and applications that SOF Operators and analysts use across warfighting functions.

CDB X - Meeting the requirements for tactical GEOINT that can be used across warfighting functions.

Chapter 2. Executive Summary

Since 2018 the CDB community has discussed and considered what CDB version 2.0 might provide in terms of capabilities. These discussion also touched on architecture and related logical models. Many of these CDB 2.0 discussion have occurred as part of the OGC standards work. More recently, other organizations such as SOCOM and NGA, have vigorously engaged in these discussions. However, progress has been relatively slow.

Therefore, in order to accelerate the development of CDB 2.0, in May 2020 SOFWERX announced the [3D Geospatial Series Tech Sprint II – OGC CDB 2.0](https://events.sofwerx.org/3dgeots/) [<https://events.sofwerx.org/3dgeots/>]. In support of SOF Future Concepts, this effort hopes to accelerate evolution of the OGC CDB standard to meet the needs of planning, rehearsal, and Mission Command systems providing decision support to Special Operations Forces and enabling SOF tactical and operational advantage.

The CBD Tech Sprint was comprised of five (5) key work phases:

Phase 1: 22 June 2020 to 26 June 2020 Virtual Concept Development: In a series of virtual workshop conferences, participants discussed CDB 2.0 requirements and design objectives with a focus on defining conceptual and logical models for the CDB 2.0 standard. The discussion focused on five key topics: Attribution, 3D Models, Vector Data/GeoPackage, Coverage Tiling (and LoDs and general tiling), and Top Level Metadata. This models and related topics discussions served as the basis for technical experimentation in Phase 2.

Phase 2: 06 July 2020 to 31 July 2020 Virtual Tech Sprint: Participants, and/or their organizations, will provide prototype implementations of the conceptual model documented in Phase 1. Through technical integration experimentation, the model will be updated and refined.

Phase 3: 03 August 2020 to 28 August 2020 Engineering Report (ER): Participants will complete the project Engineering Report detailing the conceptual model and results of the technical integration experiments.

Phase 4: 07 September 2020 to 25 September 2020 Proposed Draft Specification: Participants will create a proposed draft specification for “CDB 2.0” based on work in the previous phases, to be submitted to the appropriate working groups within OGC, Khronos, and/or SISO.

Phase 5: 05 October 2020 to 30 October 2020 Final Report: Participants will write a final report summarizing activities to date, recommendations for the OGC SWG and other standards bodies to consider, and contributions of each participant.

2.1. Recommendations

The following are the recommendations for changes, enhancements, and design principals for not just CDB 2.0 but also CDB version 1.3. The version 1.3 changes are viewed as important enhancement that resolve both existing user specified needs as well as helping to provide a transition path the version 2.0

2.1.1. Attribution

Recommendation 1 Extended Attributes - Version 1.3 The subgroup discussion on this topic is titled: [Should Extended Attributes be preserved at the logical data model level?](https://github.com/sofwerx/cdb2-concept/issues/25) [https://github.com/sofwerx/cdb2-concept/issues/25] The suggestion is that the CDB SWG discuss this issue and possible solution as a possible change for CDB version 1.3. Some additional testing may be required to determine if this capability can be added to version 1.3 or not.

Recommendation 2 Attribute default values - Version 1.3 The subgroup discussion on this topic is titled: [Attribute Default Values #32](https://github.com/sofwerx/cdb2-concept/issues/32) [https://github.com/sofwerx/cdb2-concept/issues/32]. The recommendation is that Defaults.xml can be used to define global attribute defaults as well as per-dataset defaults. Doing per-entity defaults would be a straight forward extension that could be proposed for CDB 1.3 as a transition path. The subgroup suggests that the CDB SWG discussion this for possible inclusion in version 1.3. A change request for this approach to specifying default values is also suggested.

Recommendation 3 Attribute Terms - Version 1.3 The subgroup discussion on this topic is titled: [Capture Attribute Terms \(Enumerants\) in Metadata #31](https://github.com/sofwerx/cdb2-concept/issues/31) [https://github.com/sofwerx/cdb2-concept/issues/31]. Attributes describing qualitative values are present in CDB 1.2 and the list of valid values for each attribute are documented in the human-readable specification with both the vocabulary term name and its integer numeric value (index). However, the machine-readable XML metadata does not contain any of this information and treats these attribute types as raw integers with only a minimum and maximum value constraint. It may make sense as a transition path to update CDB 1.3 to define additional XML elements in a backward compatible way to capture these definitions from the existing specification into the machine-readable XML metadata. The conceptual model in the CDB 1.2 specification does align with how GGDM treats such attributes, so there is no fundamental incompatibility, and the proposed CDB X dictionary design accounts for properly tracking the terms for qualitative attributes in a machine-readable way in SQLite.

2.2. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Contacts

Name	Organization	Role
David Graham	Eaglecapsystems	Editor
Carl Reed, PhD	Carl Reed & Associates	Editor
Kevin Bentley	Cognitics	Contributor
Holly Black	CAE	Contributor
Hermann Bressard	Presagis	Contributor
Patrick Cozzi	CESIUM	Contributor
Brian Ford	FlightSAfety	Contributor
Ryan Franz	FlightSafety	Contributor
Jay Freeman	CAE	Contributor

Name	Organization	Role
Jérôme Jacovella-St-Louis	Ecere	Contributor
Michala Hill	Cognitics	Facilitator/Contributor
Greg Peele	Geometric Progress	Contributor
Vaughn Whisker	ARL PSU	Contributor
Tracey Birch	CloudLake/USSOCOM SOF AT&L	Emeritus

2.3. Foreword

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.

Chapter 3. References

The following normative documents are referenced in this document.

NOTE: Only normative standards are referenced here, e.g. OGC, ISO or other SDO standards. All other references are listed in the bibliography.

- OGC: 15-113r5 [Volume 1: CDB Core Standard: Model and Physical Data Store Structure](https://portal.opengeospatial.org/files/15-113r5) [https://portal.opengeospatial.org/files/15-113r5]
- OGC: 16-007r4 [Volume 11: CDB Core Standard Conceptual Model](https://portal.opengeospatial.org/files/16-007r4) [https://portal.opengeospatial.org/files/16-007r4]
- OGC: 12-128r15 [Geopackage Encoding Standard - with Corrigendum version 1.2.1](https://portal.opengeospatial.org/files/12-128r15) [https://portal.opengeospatial.org/files/12-128r15]
- OGC: 18-000 [GeoPackage Related Tables Extension](http://docs.opengeospatial.org/is/18-000/18-000.html) [http://docs.opengeospatial.org/is/18-000/18-000.html]
- OGC: 17-066r1 [GeoPackage Extension for Tiled Gridded Coverage Data](http://docs.opengeospatial.org/is/17-066r1/17-066r1.html) [http://docs.opengeospatial.org/is/17-066r1/17-066r1.html]
- Khronos [GL Transmission Format \(glTF\) version 2.0](https://github.com/KhronosGroup/glTF/tree/master/specification/2.0) [https://github.com/KhronosGroup/glTF/tree/master/specification/2.0]

Chapter 4. Terms and definitions

For the purposes of this report, the definitions specified in Clause 4 of the OWS Common Implementation Standard [OGC 06-121r9](https://portal.opengeospatial.org/files/?artifact_id=38867&version=2) [https://portal.opengeospatial.org/files/?artifact_id=38867&version=2] shall apply. In addition, the following terms and definitions apply.

4.x

authoritative data

Officially recognized data that can be certified and is provided by an authoritative source.

(SOURCE: Authority and Authoritative Sources: Clarification of Terms and Concepts for Cadastral Data. FGDC/NIST 2008)

4.x

authoritative data source

An information technology (IT) term used by system designers to identify a system process that assures the veracity of data sources. These IT processes should be followed by all geospatial data providers. The data may be original or it may come from one or more external sources all of which are validated for quality and accuracy.

(SOURCE: Authority and Authoritative Sources: Clarification of Terms and Concepts for Cadastral Data. FGDC/NIST 2008)

4.x

conceptual model

description of common concepts and their relationships, particularly in order to facilitate exchange of information between parties within a specific domain. A conceptual model is explicitly chosen to be independent of design or implementation concerns.

(SOURCE: CEN ENV 1613:1995)

4.x

coverage

feature that acts as a function to return values from its range for any direct position within its spatio-temporal domain

4.x

coordinate reference system

coordinate system that is related to the real world by a datum

(SOURCE: ISO 19111:2019 Geographic information — Referencing by coordinates)

4.x

dataset

A dataset is a collection of data, published or curated by a single agent. Data comes in many forms including numbers, words, pixels, imagery, sound and other multi-media, and potentially other types, any of which might be collected into a dataset.

Note: There is an important distinction between a dataset as an abstract idea and a distribution as a manifestation of the dataset

(SOURCE: [W3C Data Catalog Vocabulary \(DCAT\) - Version 2](https://www.w3.org/TR/vocab-dcat-2/) [<https://www.w3.org/TR/vocab-dcat-2/>], 2020)

4.x

Data Store

A data store is a repository for persistently storing and managing collections of data which include not just repositories like databases, but also simpler store types such as simple files, metadata, models, etc.

(SOURCE: <https://www.information-management.com/glossary/d.html>:2020)

4.x

elevation

Synonym for “height”

(SOURCE: [OGC Abstract Topic 6: Schema for coverage geometry and functions](https://portal.opengeospatial.org/files/?artifact_id=19820) [https://portal.opengeospatial.org/files/?artifact_id=19820])

4.x

feature data dictionary

A Feature data dictionary is a collection of descriptions of the Feature objects or items in a Feature data model for the benefit of programmers and others who need to refer to them.

(SOURCE: OGC Testbed-13: CDB Engineering Report - 17-042)

4.x

foundation data

authoritative data and models that have been curated and stored in a CDB repository. Foundation data uses known and agreed to controlled vocabularies for attribution, has been curated based on the requirements as stated in the CDB standard, and supported by the required metadata.

4.x

geospecific model

A Geospecific model is instanced once and only once within a CDB. Geospecific models usually correspond to unique (in either shape, size, texture, materials or attribution), man-made, real world 3D features. NOTE: Being discussed for possible evolution as part of CDB 2.0

4.x

geotypical model

A Geotypical model is instanced multiple times within a CDB data store. Geotypical models correspond to representative (in shape, size, texture, materials and attribution) models of real-world manmade or natural 3D features.

4.x

height

Distance of a point from a chosen reference surface measured upward along a line perpendicular to that surface. 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.

(SOURCE: ISO 19111:2019 Geographic information — Referencing by Coordinates)

4.x

Metadata

information that captures the characteristics of a resource to represent the ‘who’, ‘what’, ‘when’, ‘where’, ‘why’, and ‘how’ of that resource.

(SOURCE: National System for Geospatial Intelligence (NSG) Metadata Foundation (NMF) Version 3.0 [https://nsgreg.nga.mil/doc/view?i=4252&month=10&day=22&year=2019])

4.x

repository

a place that holds CDB data, makes CDB data available to use, and organizes CDB data in a logical manner. (Network of the National Library of Medicine) 2020

4.x

resource

identifiable asset or means that fulfils a requirement

NOTE

A web resource, or simply resource, is any identifiable thing, whether digital, physical, or abstract.

(SOURCE: ISO:19115-1:2014 Geographic information — Metadata — Part 1: Fundamentals)

- **tile**

geometric shape with known properties that may or may not be the result of a tiling (tessellation) process. A tile consists of a single connected "piece" without "holes" or "lines" (topological disc).

NOTE

"tile" is NOT a packaged blob of data to download in a chunky streaming optimization scheme!

- **tiling**

in mathematics, a tiling (tessellation) is a collection of subsets of the space being tiled, i.e. tiles that cover the space without gaps or overlaps.

4.1. Abbreviated terms

ATAK	Android Tactical Assault Kit
COTS	Commercial Off The Shelf
CRS	Coordinate Reference System
DGIM	Disaster Geo-Information Management
GGDM	Ground-Warfighter Geospatial Data Model
GPKG	GeoPackage
glTF	GL Transmission Format
FACC	Feature Attribute and Coding catalog

FDD	Feature Data Dictionary
FSC	Feature Sub-code
LoD	Level of Detail
MC	Mission Command
NAS	NSG Application Schema
NSG	National System for Geospatial-Intelligence
OTW	Out the Window
OWT	One World Terrain
PBR	Physically-Based Rendering (PBR)
SOF	Special Operations Forces
STAC	SpatioTemporal Asset Catalog
TIFF	Tagged Image File Format
TMS	Tile Map Service
TMS	Tile Matrix Set

Chapter 5. Overview

Section 5 provides the background and history of CDB to provide the context for the CDB X work described in this Engineering Report.

Section 6 Documents any models, such as the OGC/ISO Simple Features geometry model that are germane to the CDB 2.0 work.

Section 7 Attribution in CDB 1.3 and 2.0: Section 7 discusses the Attribution Topic and the work and recommendations of the Attribution Subgroup.

Section 8 Vectors-GeoPackage:

Section 9 3D

Section 10 top-level-metadata

Section 11 Tiling coverages in CDB X

Chapter 6. CDB 2.0 - Background and History

This section of the ER provides background information on the history and development of version 1 of the CDB standard. This background information provides the basis for the decision for the SOFWERX/SOCOM sponsored CDB X Conceptual Modelling activity.

6.1. What is CDB?

The existing CDB standard defines a standardized model and structure for a single, versionable, virtual representation of the earth. A CDB structured data store provides 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. In this case, a CDB is plug-and-play interoperable between CDB-compliant simulators. A CDB can be readily used by existing simulation client-devices (legacy Image Generators, Radar simulator, Computer Generated Forces, etc.) through a data publishing process that is performed on-demand in real-time.

The application of CDB to future simulation architectures will significantly reduce runtime-source level and algorithmic correlation errors, while reducing development, update and configuration management timelines. With the addition of the [High Level Architecture](https://en.wikipedia.org/wiki/High-level_architecture) [https://en.wikipedia.org/wiki/High-level_architecture] - Federation Object Model (HLA/FOM) and [Distributed Interactive Simulation](https://en.wikipedia.org/wiki/Distributed_Interactive_Simulation) [https://en.wikipedia.org/wiki/Distributed_Interactive_Simulation] (DIS) protocols, the application of the CDB standard provides a common environment to which inter-connected simulators share a common view of the simulated environment.

The CDB standard defines an open format for the storage, access and modification of a synthetic environment database. A **synthetic environment** is a [computer simulation](https://en.wikipedia.org/wiki/Computer_simulation) [https://en.wikipedia.org/wiki/Computer_simulation] that represents activities at a high level of realism, from simulation of theaters of war to factories and manufacturing processes. These environments may be created within a single computer or a vast distributed network connected by local and wide area networks and augmented by super-realistic special effects and accurate behavioral models. SE allows visualization of and immersion into the environment being simulated see "[Department of Defense Modeling and Simulation \(M&S\) Glossary](#)", [DoD 5000.59-M](#), [<https://www.msco.mil/MSReferences/Glossary/MSGlossary.aspx>].

This standard defines the organization and storage structure of a worldwide synthetic representation of the earth as well as the conventions necessary to support all of the subsystems of a full-mission simulator. The standard makes use of several commercial and simulation data formats endorsed by leaders of the database tools industry. A series of associated OGC Best Practice documents define rules and guidelines for data representation of real world features.

The CDB synthetic environment is a representation of the natural environment including external features such as man-made structures and systems. A CDB data store can include terrain relief, terrain imagery, three-dimensional (3D) models of natural and man-made cultural features, 3D models of dynamic vehicles, the ocean surface, and the ocean bottom, including features (both natural and man-made) on the ocean floor. In addition, the data store can include the specific

attributes of the synthetic environment data as well as their relationships.

6.2. OGC CDB 1.x History

The CDB specification was initially authored by CAE Inc. on November 2005 under a contract administered by the US Army Program Executive Office for Simulation Training and Instrumentation (PEO STRI) to meet requirements set by US Special Operations Command (USSOCOM) for interoperable, high-performance mission rehearsal and simulation federations. CAE along with other companies (e.g., Flight Safety Inc., New York, NY, USA, Presagis Inc., Montréal, QC, Canada, etc.) have delivered hundreds of simulation channels based on CDB to customers in the US, Canada, UK, Germany, Turkey, Israel, Singapore, Australia, Brunei and other countries.

To improve the efficiency of M&S analysis, facilitate data and services' interaction and communications, and reduce operational cost and complexity, the CDB Community led by CAE submitted the industry specification version 3.2 into the OGC for consideration as an OGC Best Practice. Beginning in 2013, CDB was discussed and demonstrated at OGC Technical Committee meetings. The industry specification was approved by the OGC Members and released as an [OGC Best Practice](https://portal.opengeospatial.org/files/?artifact_id=61935) [https://portal.opengeospatial.org/files/?artifact_id=61935]. The OGC CDB SWG then split the two Best Practice documents into 12 volumes, removed unnecessary content by providing links to authoritative sources, and began evolving terminology to be consistent with OGC/ISO terms and definitions. CDB Version 1.0 was approved by the OGC membership in October 2016 as a new standard.

The CDB SWG continued the evolution of CDB. Version 1.1 of the CDB Standard and related Best Practices was approved in August 2018. One of the main enhancements for version 1.1 was guidance on how to encode and include global and local spatial metadata in a CDB data store. Version 1.1 also moved to having file extensions being generic in all CDB Requirements clauses. This was done to allow inclusion of new (additional) file types in a CDB data store. Further, a number of CDB User community changes requests were included in version 1.1.

Version 1.2 of the CDB Standard was approved as an official OGC Standard in August 2020. The major enhancement in CDB version 1.2 was the inclusion of the ability to add OGC GeoPackage formatted containers into a CDB data store. There are two new CDB volumes that explicitly provide requirements and guidance on using GeoPackages in a CDB data store. Version 1.2 also includes two substantive changes. First, the way that CDB's Primary Alternate Terrain Elevation dataset was defined in CDB Version 1.1 and earlier causes problems with standard open source libraries used to read and process these data. The SWG agreed to changes address the issues that ground simulation has with CDB gridded terrain meshes. The second substantive change is that CDB 1.1 and earlier supported a single (hard-coded) file format per dataset. To allow other file formats to be used in a CDB Data Store, the need to explicitly specify the file format that is used to physically store the components of a given dataset is required. The current CDB metadata and controlled vocabulary definitions (See Clauses 1.4.3, 3.1, and 5.1 in CDB Volume 1: Core) has a file called Datasets.xml listing all possible datasets that can be used in a CDB data store. The file has been expanded to indicate the encoding format used to encode the dataset and its components.

The CDB community anticipates that additional standardization work will be required to prescribe content appropriate to targeted simulation applications, new use cases, and application in new domains. For example, in its current form, the CDB standard does not mandate synthetic

environmental richness, quality and resolution.

6.3. OGC CDB 2.0

Beginning in late 2020, the OGC CDB SWG began discussion on what CDB version 2.0 might have as an architecture, what additional encodings and/or formats might be supported, and so forth. These discussions also included initial thoughts on a CDB 2.0 conceptual model. However, progress was slow - in part due to other OGC Member commitments and resource constraints.

NOTE In the OGC, a major version number indicates that the new version is not backwards compatible with previous versions of the standard. The use of major revisions allows for a major evolution and enhancement to any OGC Standard.

A number of the major requirements for version 2.0 have been discussed for some time and have been captured in various OGC documents including formal Change Requests and OGC Interoperability Program Engineering Reports.

Some of the major enhancements that have been discussed and documented are:

1. Develop an approach (and model?) for using any attribution controlled vocabulary in a CDB data store.
2. Determine if the current tiling scheme is optimal or should be enhanced/replaced with a better tiling approach.
3. Describe explicitly how the CDB model may or may not align with the OGC DGGS standard;
4. Provide best practice details on how to use other OGC standards such as WMS and WFS as well as the new OGC APIs to access existing CDB data stores. This work may require Interoperability Experiments to better understand the implications of these decisions;
5. Extend the supported encodings and formats for a CDB structured data store to include the use of the CityGML, and InDoorGML standards as well as other broadly used community encoding standards, such as glTF and GeoTIFF. This work may require performing OGC Interoperability Experiments to better understand the implications of these decisions.
6. Further align CDB terminology to be fully consistent with OGC/ISO terminology.

Making these enhancements may allow the use and implementation of a CDB structured data store for application areas in addition to aviation simulators.

6.4. Attribution - An OGC Testbed 13 CDB Activity in

The [CDB Engineering Report](http://docs.opengeospatial.org/per/17-042.html) [http://docs.opengeospatial.org/per/17-042.html] (ER) summarizes the CDB sub-thread activity. The CDB activity was divided into three main work packages:

- A feasibility study;
- The implementation of data models and schemas mapping that are based on the feasibility study results; and
- A set of OGC web services that implement the CDB in the form of WFS and WCS (Web Coverage

Service) instances.

This Testbed 13 Engineering Report describes the approach taken and the results of the experimentation:

- The conceptual model of an OGC CDB 1.0 datastore as a UML (Unified Modeling Language) diagram to show different datasets (the 3D models, vector features and coverages) structure;
- How to process and use a NAS-based Profile as a CDB feature/attribute data model or a GML-SF0 application schema;
- How to access, navigate and visualize a CDB dataset using OGC web services (such as WFS and WCS).

The Testbed 13 CDB activity also resulted in a formal [OGC Change Request Proposal](http://ogc.standardstracker.org/show_request.cgi?id=544) [http://ogc.standardstracker.org/show_request.cgi?id=544] that was submitted for consideration in April 2018. The Change Request provides:

- Recommendations for replacing FACC feature code and indexing structure to be consistent with the application schemas (e.g. NAS data model) under discussion for the OGC CDB 2.0;
- Recommendations for supporting application schemas in CDB (level of complexity: Esri Geodatabase, GML-Simple Features Level 0 application schema) are being discussed for the OGC CDB 2.0;
- A method to expand the supported encodings and formats for an OGC CDB compliant datastore;
- Guidance on generating a coherent attribute schema for CDB 1.0 based on the "CDB_Attribute.xml" file.

Chapter 7. CDB X Design Goals, Use Cases, and Conceptual/Logical Models

This section documents draft design objectives, the target use cases, and the CDB X Conceptual and Logical Models.

7.1. CDB X Design Goals and Objectives

The following are the design goals and objectives for a tiled CDB X repository.

- Be fast :-)
- Geometry model: All geometry for any vector feature data shall be compliant with the OGC/ISO Simple Features Geometry Model.
- Correlation Requirement: CDB-T, CDB-S, CDB-G and CDB-A profiles shall correlate to the data defined in CDB-F
- Derivative and Optional structures: CDB-T, CDB-S, CDB-G and CDB-A profiles shall encode its rules and mechanisms for optimization in a self-describing form.
- Holistic Foundation: CDB-T, CDB-S, CDB-G and CDB-A structures shall all be generated from the data defined in CDB-F
- Deployment: CDB-F shall be able to be deployed on the cloud, in relational databases, and on a file system.

7.2. Use Cases

The following are summaries of key use cases identified during this project.

Tactical Use Case: <need short description> Possible impact: Define a profile of the foundation data in a form most suitable for tactical devices (CDB-T, where T is for tactical), such as ATAK.

Gaming Use Case <need short description> Possible impact: Define a derivative and optional structure of the .foundation data in a form most suitable for gaming devices (CDB-G, where G is for gaming), such as Unreal or Unity.

M&S Use Case <need short description> Possible impact: Define a profile of the foundation data in a form most suitable for simulators (CDB-S, where S is for simulators)

Analytics Use Case <need short description> Possible impact: Define a derivative and optional structure of the datasets in a form most suitable for analytics using AI/ML (CDB-A, where A is for analytics)

Data Transport <need short description> Possible impact: Optimize the speed data synchronization between all echelons and data systems for CDB-F and CDB-T

DDIL Deployment CDB-F, CDB-T, and CDB-A must be able to deploy in DDIL environments.

NOTE

Optimization For Use Cases: CDB-S, CDB-G and CDB-T may tile/generate LODs and derivative data types (assuming they are OGC and/or Khronos standards) in a manner to optimize efficiency and performance leveraging a small, finite set of tiling schemes (Profile should include Tiling structure, LOD structure, optimization, etc.) CDB-A may generate topology, extended attributes and derivative data types (assuming they are OGC and/or Khronos standards) in a manner to optimize efficiency and performance leveraging a small, finite set of relationship schemes

This is future work and actually would not be part of a core CDB-X standard Configuration Management. Define an abstract mechanism for configuration management applicable to multiple deployment strategies (file systems, RDMS and cloud) relational databases for CDB-F, CDB-S, CDB-G, CDB-T and CDB-A. Support Versioning, attribution, synchronization

7.3. Definition of Concepts

7.3.1. Foundation Data

Foundation data are the authoritative data and models that have been curated per the rules defined in the CDB standard, and stored in a CDB repository. Foundation data utilizes known and consensus controlled vocabularies for attribution, has been curated based on the requirements as stated in the CDB standard, and the required metadata has been documented.

For example, as per the current CDB standard (and industry best practice), a road network (RoadNetwork dataset in CDB 1.2) could be considered as foundation data. In CDB 1.2, there are a set of requirements associated with how the roads data are structured and attributed and stored in a CDB repository.

The specification of any foundation data layer (such as RoadNetwork) at the logical model level can be 100% independent from the storage environment, end user use case(s) and so on while at the same time also having the requirements based on industry knowledge and use case requirements.

For example:

Requirement: The RoadNetwork shall be considered as a foundation data layer in a CDB repository.

Sub-requirement: All roads in a roads layer SHALL use a known road classification system to identify the roads type. This could be the US DoT classification, the current FACC classification, or the newer GDDM.

Sub-requirement: The RoadNetwork SHALL be topologically structured. (This enables network analysis etc). NOTE: The how is not specified at the logical model level.

Sub-requirements: The RoadNetwork Dataset SHALL be used to specify all of the road networks..

Notice that there is no mention of the actual storage structure, enabling software, storage formats, tiling and so forth.

7.3.2. Reference Systems

7.4. 3D Models

7.5. Attribution

7.6. Tiling/Coverages/Imagery

7.7. Vector Data

Simple Features Geometry Types

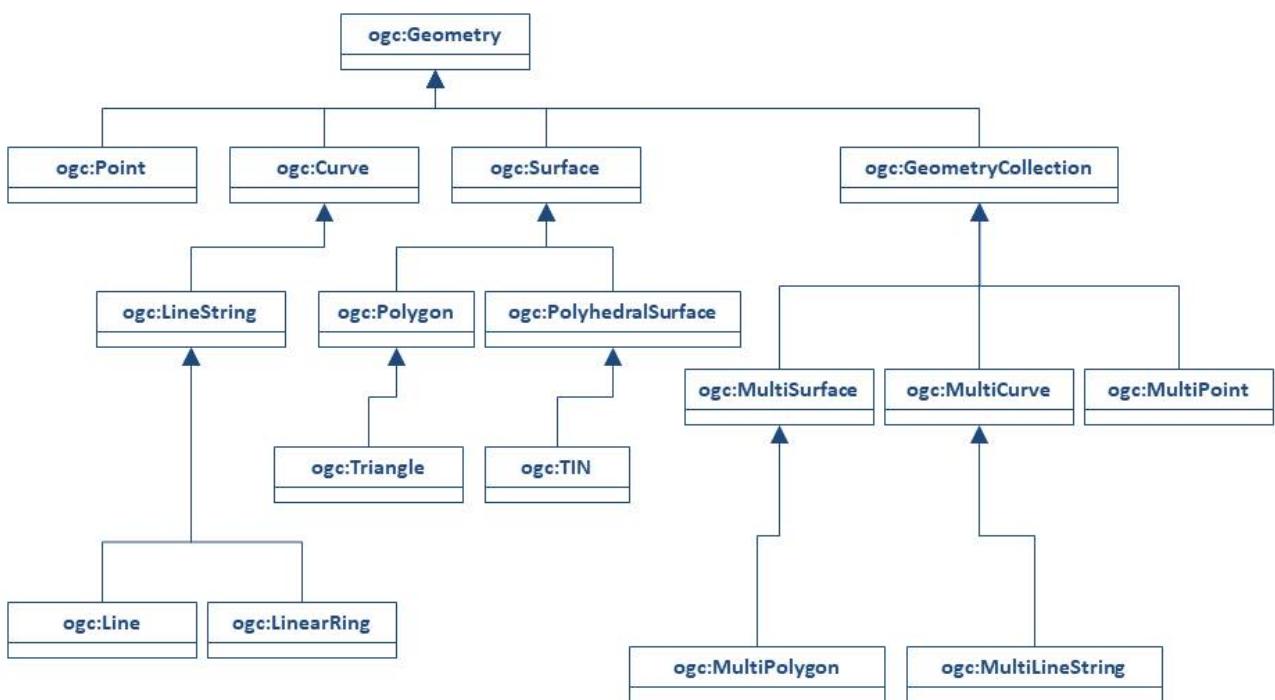


Figure 1. Simple Features Geometry Model.

Chapter 8. CDB-X Attribution

For several years, there have been discussions concerning how to modify the CDB standard to support attribute coding schemes (aka controlled vocabularies) other than the current approach that is based on a Feature and Attribute Coding Catalog (FACC). However, at this time any changes to the CDB standard to accommodate other coding schemes have not occurred. This section provides 1.) information on the current approach in CDB 1.x, 2.) requirements for evolving the CDB standard to accommodate other coding schemes and 3.) the experimental framework and results from the work in the CDB-X activity.

8.1. Current CDB 1.x Attribution Approach

OGC CDB versions 1.x use a CDB Feature Data Dictionary (FDD) as the controlled vocabulary for attribution. An Excel spreadsheet version of the CDB 1.X Feature Data Dictionary is available on the public [CDB wiki](https://external.ogc.org/twiki_public/pub/CDBswg/WebHome/CDB_FDD.xlsx) [https://external.ogc.org/twiki_public/pub/CDBswg/WebHome/CDB_FDD.xlsx].

The CDB V1.x feature codes and subcodes are primarily based on the [DIGEST Feature and Attribute Coding Catalog 2.1](https://gwg.nga.mil/documents/asfe/DGIWG_FACC.htm) [https://gwg.nga.mil/documents/asfe/DGIWG_FACC.htm] (FACC) with additional concepts from a few other sources. From that website: *FACC specifies geospatial information concepts used by member nations of the multi-national Defence Geospatial Information Working Group (DGIWG) community. These concepts characterize aspects of real-world entities (or objects) and related properties, including those that are not necessarily visible or have a tangible physical form (e.g., airspace). The DGIWG FACC is a comprehensive dictionary and coding scheme for feature types, feature attributes (properties or characteristics associated with features), and attribute values (domain of feature attributes).*

In the FACC, each major feature category is further divided into subcategories identified by the second character of the five-character code containing an alphabetic value from "A" to "Z". Finally, the third, fourth, and fifth characters of the five-character feature code are a numeric value from 000 to 999. This value provides unique feature type identification within categories. Thus, all features must be identified by all five alphanumeric characters; e.g., the feature named "Building" is denoted by the code "AL015". Feature codes are listed in Annex A of the FACC Data Dictionary.

Each feature is associated with a textual description, which provides a human readable dictionary definition for the feature. Each Feature Code is also associated with a short human readable name.

Code	Name	Definition
AA010	Mine	An excavation made in the earth for the purpose of extracting natural deposits. (See also AQ090)
BB150	Landing Place	A place on shore where landing from the sea is possible.

DB180	Volcano	A mountain or hill, often conical, formed around a vent in the earth's crust through which molten rock, ash, or gases are or have been expelled.
FA001	Administrative Area	An area controlled by administrative authority.
GB075	Taxiway	A prepared surface providing access to/from runways and the aircraft parking area, terminal area, or service area, etc.

An OGC [Change Request Proposal](#) [http://ogc.standardstracker.org/show_request.cgi?id=544] (CRP) related to CDB attribution was generated during OGC Testbed 13:

NOTE: An overview of the Testbed 13 activity is provided in the [Background Section](#) section of this ER.

Note: Most of the images in this section of the report are snippets of the Miro whiteboard used for collaboration throughout Phase 3 of the Tech Sprint. The board was archived at various points through the spring by the moderator. At the time of the drafting of this report, the link to the Miro whiteboard is at: https://miro.com/app/board/o9J_kqG1Jw0/

8.2. CDB-X Discussion and Requirements for CDB-X Experimentation

8.2.1. Experimentation Backround

There was early consensus during Phase 3 of the CDB-X tech sprint to experiment with and potentially adopt an approach to CDB attribution generally called "GGDM".

Greg Peele, one of the Tech Sprint participants, has described his work as a sub-contractor on the team recommending a Data Model for the U.S. Army One World Terrain (OWT) effort in multiple presentations to the OGC Interopable Simulation and Gaming Domain Working Group. Greg's most recent presentation, entitled "**Entities, Attributes, and Enumerants, Oh My! Applying GGDM for Interoperable One World Terrain Semantics**" to the ISG DWG may be found here: https://portal.ogc.org/files/?artifact_id=93666

8.2.2. Some questions to be addressed during experimentation

1. GGDM/NAS define standard layers that subdivide vector features into semantic sets (i.e. hydrology, industry, extraction, etc.). Is this meaningful to CDB? Or is it irrelevant?
2. GGDM can be cross walked to NGA TDS 7.1. What are the missing attributes between M&S and TDS?
3. Is there a challenge on the Sim for reading Feature Codes Vs Attribution?

4. What are the format implications: vector and mesh formats must support GGDM attribution and entity types?
5. Metadata: GGDM defines metadata as attributes. Encoding decision: separate metadata table referenced from features via foreign key? Or flattened metadata attributes present on every feature?
6. GGDM does not have mandatory attribution fields. All fields are optional. What are the implications?
7. What attribution is missing from GGDM for CDB?

8.2.3. Phase 3, Day 3

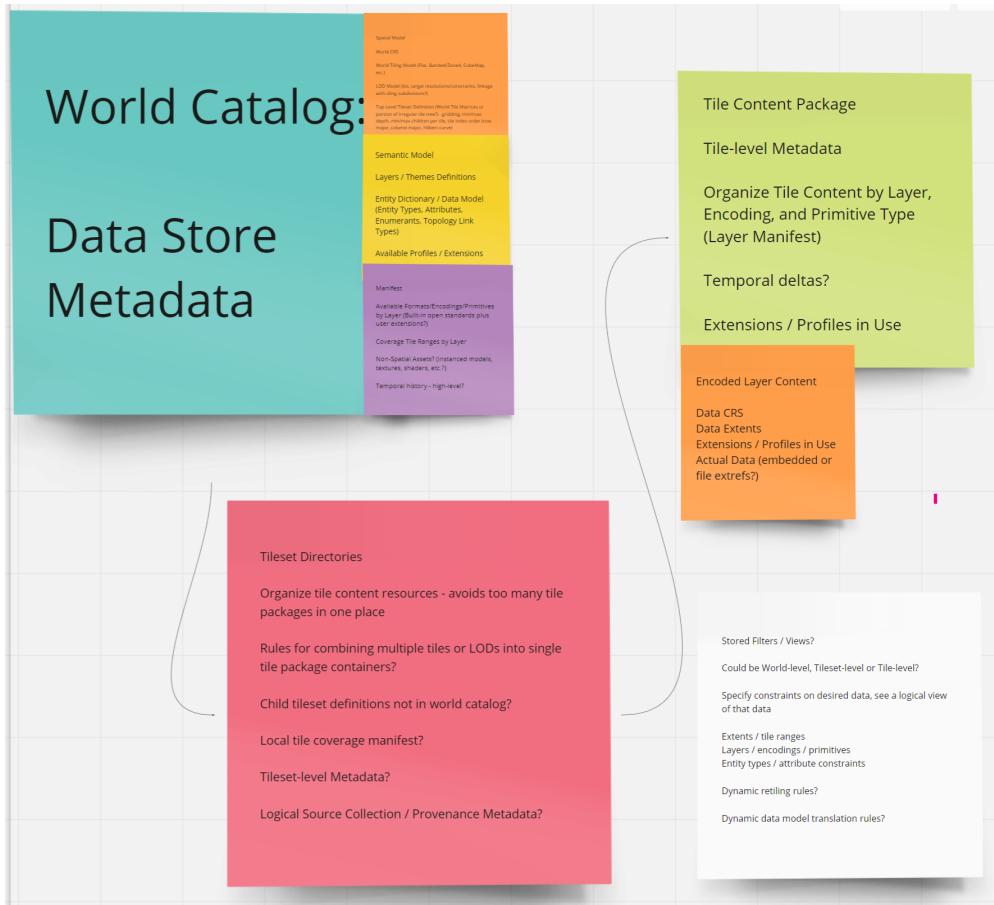


Figure ATTPh3 - 2. Greg Peele's Whiteboard from Phase 3 Day 3.

8.2.4. Phase 3, Day 4



Figure ATTPh3 - 3. 'Attribution will be in GGDM, Prove me Wrong' Day 4 Sign.

We will use GGDM for attribution

Holly
Greg
Randi

Feature Attribute Metadata
Feature Attribute Metadata is collected as attributes on each feature; these attributes represent information necessary to understand the feature source, currency, usability scale, schema, as well as classification and dissemination information.

Geometry Enabled Metadata
Geometry Enabled Metadata is collected as surface features or tables to represent a collection of features with the same metadata. The GGDM contains four geometry enabled feature groups for metadata: ResourceSrf feature class with DATASET_S feature, DATASET_T table, MetadataSrf feature class with ENTITY_COLLECTION_METADATA_S feature, and ENTITY_COLLECTION_METADATA_T table.

GGDM Drawback: Denormalized Definitions

- GGDM (as normally presented) duplicates entity types and attributes
 - Entity types have duplicate definitions for different geometry types
 - Same attributes are duplicated across different entity types, sometimes with different code values e.g. Cable Type is "CAB" on Cable but "AT005_CAB" on Pylon
 - Multi-valued attributes are mapped as three separate attributes (max value count of 3 attributes for min/max and an attribute for range closure)
 - Same enumerants are duplicated across many different attribute types with different code values depending on the attribute
 - Material types are particular offenders – "Structural Material Type" vs. "Vertical Construction Material" vs. "Route Surface Composition" and so on
 - GGDM approach decoheres the consistent hierarchy definitions from NGDC
 - As a practical matter, this makes the "standard" GGDM spreadsheet difficult to use

OGC®

GGDM can be cross walked to NGA TDS 7.1

what are the missing attributes between M&S and TDS

Is there a challenge on the Sim for reading Feature Codes Vs Attribution?

GGDM/NAS define standard layers that subdivide vector features into semantic sets (i.e. hydrology, industry, extraction, etc.)

Is this meaningful to CDB? Or is it irrelevant?

Format implications: vector and mesh formats must support GGDM attribution and entity types

To Be Considered:

Metadata:
GGDM defines metadata as attributes

Encoding decision: separate metadata table referenced from features via foreign key? Or flattened metadata attributes present on every feature?

GGDM does not have mandatory attribution fields; all optional

What attribution is missing from GGDM for CDB?

TDS - Table Root Name	GGDM Composite	GGDM Feature Name	CDB/FACC-FSC	CDB-FSC	CDB/FACC-FSC Label
BUILDING	StructurePnt	BUILDING_P	AL013-000	0	Building
	StructureSrf	BUILDING_S	AL013-000	0	Building
NON_BUILDING_STRUCTURE	StructurePnt	NON_BUILDING_STRUCTURE_P	AL014-000	0	Non_Building_Structure
	StructureSrf	NON_BUILDING_STRUCTURE_S	AL014-000	0	Non_Building_Structure
			AL015-000	0	Building_Generic
			AL015-001	1	Fab_Structure
			AL015-002	2	Gov_Building
			AL015-003	3	Capitol_Building
			AL015-004	4	Castle
			AL015-005	5	Gov_Admin_Building
			AL015-006	6	Hospital

Figure ATTPh3 - 4. Attribution Day 4 Whiteboard.

8.2.5. Phase 3, Day 5

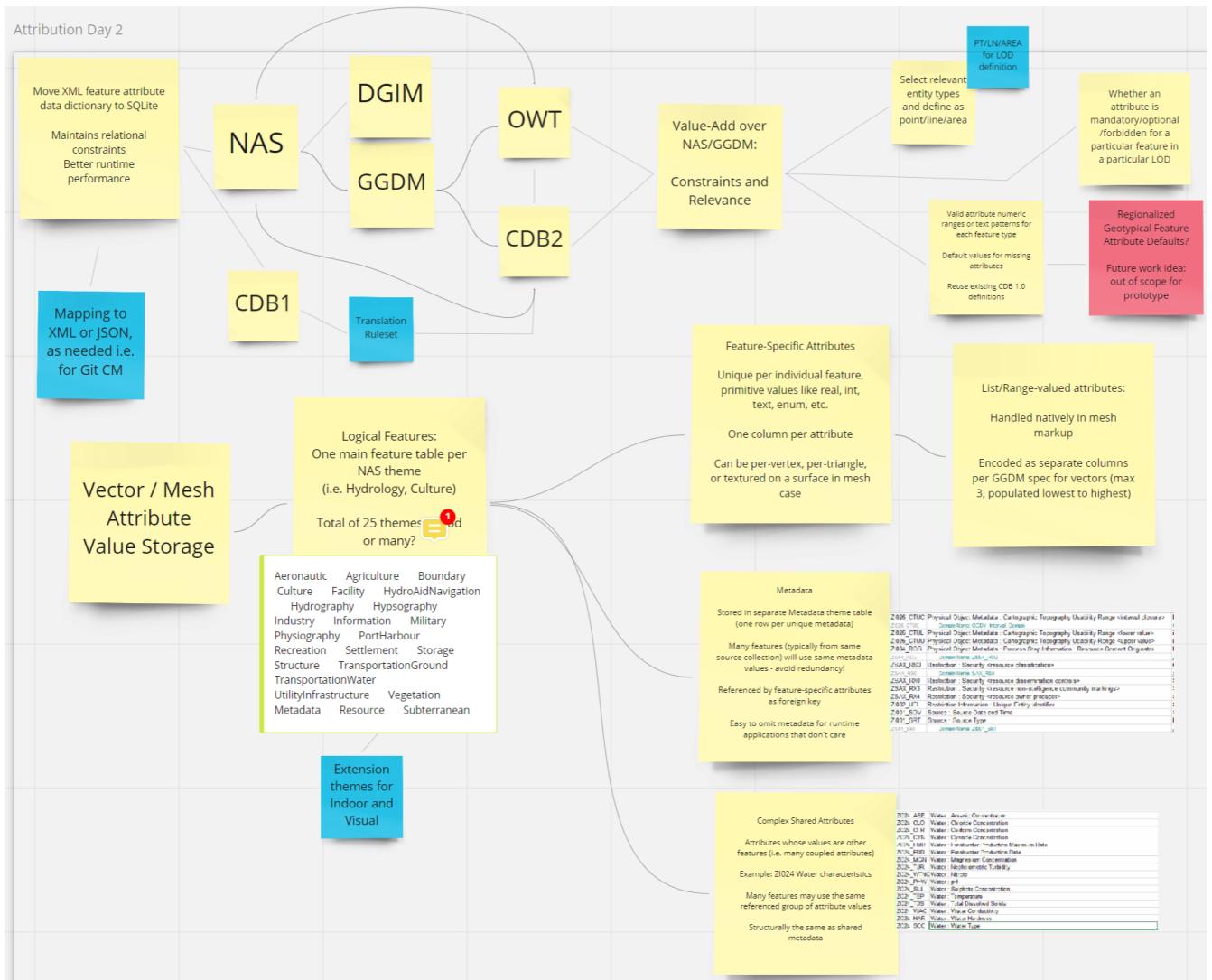


Figure ATTPh3 - 5. Attribution Day 5 Work in Progress Whiteboard One.

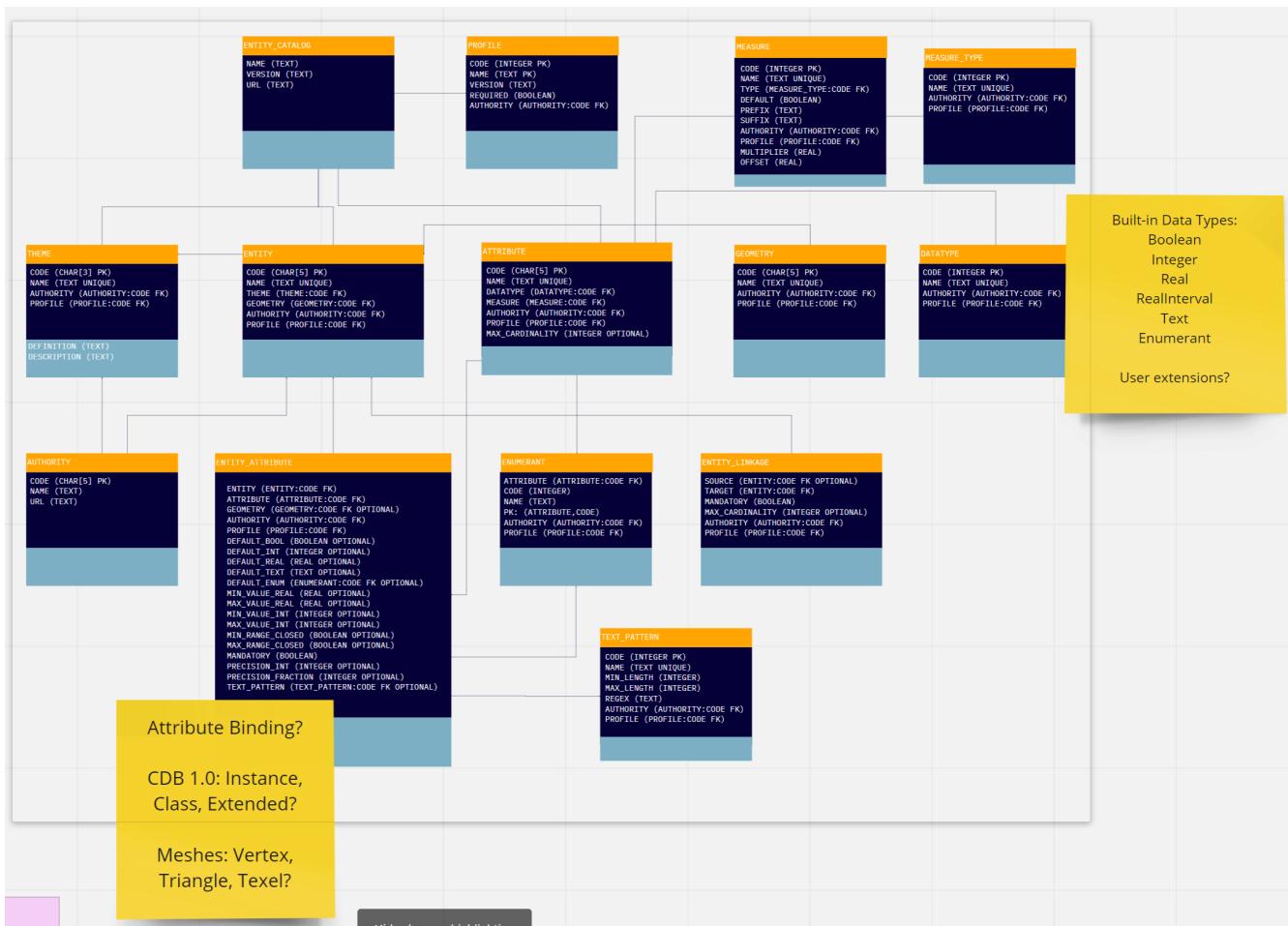


Figure ATTPh3 - 6. Attribution Day 5 Work in Progress Whiteboard Two.

Plan for Experimentation in subsequent phases

Chapter 9. CDB X All Vector Data in GeoPackage

This clause provides discussion using GeoPackage for vector data storage including requirements and the results of the prototyping activities.

9.1. General

9.2. Question to be answered

- Is the GeoPackage tiling scheme enough for the whole world with dense data, or is there some CDB tiling structure outside that uses multiple GeoPackages?
- How does GeoPackage work with the need to update data layers dynamically? SQLite has a write model that only allows a single writer lock that prevents any reads to the database.
- Lots of discussion on locking. Main sticky note states: PENDING lock means that the process holding the lock wants to write to the database as soon as possible and is just waiting on all current SHARED locks to clear so that it can get an EXCLUSIVE lock. No new SHARED locks are permitted against the database if a PENDING lock is active, though existing SHARED locks are allowed to continue.

- Discussion:

- Once an EXCLUSIVE lock is acquired, no read/SHARED locks are permitted.
- Can't that exclusive lock be released once the writing is done though?
- Absolutely, it is held as short as possible. What are the SOCOM requirements for dynamic updates? And how often might they occur?
- Also, this is about multi-processs right... A process that has an exclusive lock can both read & write from the DB.
- Reading and writing can't happen at the same time, and mostly transparent from the calling process. Also, I am assuming that reading is from one client, and the update is coming from another process.
- It could be possible for a single process to act as both the reading and proxy for the applying any updates. ***That is one method. From a flight sim perspective, there might be multiple devices that want direct access to the data. Or maybe those updates come through other channels (DIS/HLA) and not the CDB
- What I'm trying to say is if you wanted a PostgreSQL equivalent for a SQLite DB, you would implement your own server handling multiple connections on top of it and that could be considered the "direct" access to the data.
- So at that point, you are no longer interacting with the CDB, but a service
- A local service yes, but much as if you used another database like PostgreSQL (meant as a counter-argument to reasons not to chose SQLite over a heavier RDBMS).

Chapter 10. 3D Models and other 3D Content

This section discusses the proposal to use glTF to encode and store 3D models in the CDB X data store. This section also describes the prototype testing performed during the the <activity>. In CDB, 3D models are such phenomenon as trees (static), transmission towers, or helicopters (dynamic).

Currently, the OpenFlight specification is used to define all the 3D models in a CDB data store. The current OGC CDB 1.x core standard and [Volume 6 CDB Rules for Encoding Data using OpenFlight](#) [<https://portal.opengeospatial.org/files/16-009r4>] specifies all of the rules and requirements for using OpenFlight models in a CDB data store. The early advantage OpenFlight held over many 3d geometry model file formats (.obj, .dxf, .3ds) was its specific real-time 3d graphics industry design.

For CDB X, the participants in the 3D Geospatial Series Tech Sprint II – OGC CDB 2.0 activity agreed that a more modern 3D model encoding and transmission format should be explored. Based on wide adoption and use in many domains, the Khronos glTF 2.0 specification was selected for further research and prototyping. A key part of this activity was to perform a detailed analysis of glTF functionality as compared to OpenFlight capabilities. The participants agreed that no (or as little as possible) functionality should be lost. NOTE: glTF will not be a replacement for OpenFlight in a CDB data store. Instead, glTF is recommended as the preferred encoding for CDB 3D Models. There are a few OpenFlight to glTF conversion tools.

The remainder of this section discusses the findings of the comparison as well as lessons learned from the prototyping activity.

10.1. Questions to be addressed

Following are some of the questions that the 3D Modelling Subgroup initially identified as needing further discussion and possible experimentation. These formed the basis for the specific questions and approaches uses in the CDB-X 3D Prototyping and Experimentation.

1. Models be stored in GeoPackage (Editors Note: *Is this a good idea? One could store the glTF in the CDB data store without using GeoPackage - just as OpenFlight today. Need to think about also perhaps having I3S, 3D Tiles, CityGML and other encodings.*)
2. Models preferred to be glTF (others allowed) (glb package?)
3. Textures shall be xxxx format (png, jp2) Power of 2 enforced?
4. UV Map with model?
5. naming convention tbd - should have some metadata use - indexing? directory path? layer? featureID?
6. LOD Structure? (3d tiles? MS LOD extenxtion for glTF) (Editors note: *Shouldn't the same LoD structure be used throughout CDB-X? Otherwise, things get confusing.*)
7. Describe model placement
 - a. point features
 - b. Terrain Reconstruction
 - c. Elevation

- d. Imagery
 - e. Areal features. (Editor's note: We stopped using "aerial" in CDB 1.1 - changed to polygon.)
 - f. linear features
8. Is there a way to map glTF Material vs CDB physical material?

CDB catalog service for 3D tile inside a tile

10.2. CDB X Requirements for 3D Model Components

Based on the questions and followup discussion, the CDB X participants derived the following requirements for CDB-X model components.

Requirement	Objectives
Improve interoperability with OWT	<ul style="list-style-type: none"> * Align 3D model geometry attribution in order to support conversion between the two standards. Test that conversion in both directions. * Support similar encoding (glTF) to support convergence as standards evolve.
Serve archiving/repository use case and runtime usage	<ul style="list-style-type: none"> * Package 3D models LODs in an easy to edit manner while allowing fast runtime access of independent LODs. * Increase 3D tools (edit and view) interoperability. * Support multiple encoding with metadata to identify which one is used (Starting with OpenFlight and glTF).
Modernize CDB	<ul style="list-style-type: none"> * Support newer 3D encoding formats (glTF initially) * Support newer attribution (NAS/GGDM) in point feature and in 3D geometry * Provide flexibility to extend 3D models components and attribution * Review LOD “bins” in context of today’s GPU capabilities * Added constructs for better game engine support (navigation and collision mesh, PBR etc..) *Review lightpoint (Dataset and in 3D model), light source, lightmap.
Support SOF use cases : Same dataset for M&S and C2 including the mission command and TAK devices	<ul style="list-style-type: none"> * ATAK: Support 3D view and attribution * ATAK: Support building interior 3D encoding (and 2D floorplan?) * ATAK: Support packing 3D models in small memory footprint * ATAK: Support LOD selection “per model or per zone” to transfer only the required data * Mission command: Support merging of CDB (Versioning) to cover large area * Mission command: Support 3D WEB based map steaming

10.3. Short and Medium Term Objectives

The following are the short and medium term objectives for the 3D Model Prototyping and Experimentation. These design objectives are based on the initial set of [questions](#) posed by the CDB-X participants.

1. Support more 3D model encoding with metadata detailing which is used. Preserve OpenFlight (and potentially improve its usage), add glTF with extensions
2. Reduce file count and package models better while providing enough indexing data to allow readers to quickly extract only what they need. Metadata/catalog - support editing/replace use case
3. Address GS, GT, model instancing
4. Support Building interior with navigation mesh and collision mesh
5. Encode 3D models in glTF
 - a. Node grouping, LOD grouping in json
 - b. Define CDB extensions for glTF
 - c. Look at tools to allow CDB attribution editing
6. Model packing with MetaData

10.4. 3D Experiment Object Model

10.4.1. The data

This experiment uses a GeoPackage comprised of a point feature table with attribution.

id	A01	CNAM	RTAI	AHGT	BBH	BBL	BBW	BSR	CMIX	CNAM	FACC	FSC	HGT	MLOD	MODL	NIS	NIX	NNL	NTC	NTX	NVT	SSC	SSR
1	12.3653	AL241000-39-12U12R39-0	0	FALSE	15.624	40.598	35.833	28.179	2	AL015000-38-12U12R38-0	AL015	0	15.62	2	Building_-5616_-12643	1	0	0	0	0	225	0	0

Figure ATTPh3 - 7. 3D Experiment 1 attribute table for point features.

10.4.2. Attribute changes explored

The following current CDB attribute changes were explored.

- CNAM - no longer needed
- FACC - FSC - Change to GGDM - impact on field type
- MLOD - (LOD of the model to use) This is closely linked to tiling and LODs of vector where each point features would point to a model at a given LOD (should be lower than feature vector LOD). Presume at the moment we keep this.
- MODL - (Name of the model to use). In this case, we should point to a metadata file for the model. Alternatively, the content of the metadata could be encoded in GeoPackage - which is bad for tool operability.

A key challenge to resolve is how to link the model data with the feature data. Currently, the MLOD

and MODL are composed and lookup into the tile index of the point feature. Is that functional capability (approach?) preserved? Do we use a model table in Geopackage with a primary key? As decided, models would remain separate files and not encoded in the GeoPackage. So, a unique file name is required for this experiment.

10.5. Description of 3D formats/encodings and References for this section:

10.5.1. OpenFlight

OpenFlight (or .flt) is an open, freely available 3d geometry model file format originally developed by Software Systems Inc. for its MultiGen real-time 3d modeling package. OpenFlight is a format supported widely in modeling and simulation community for dynamic and static 3D model. In CDB 1.x, OpenFlight is used for the representation of 3D dynamic models and RGB format for the 3D model's textures. However, while widely used and there are extensive .flt model libraries, OpenFlight is long in the tooth.

[OpenFlight v16](https://portal.opengeospatial.org/files/90663) [https://portal.opengeospatial.org/files/90663]

10.5.2. GL Transmission Format (glTF) 2.0

glTF is a royalty-free specification for the efficient transmission and loading of 3D scenes and models by applications. glTF uses the JSON standard. glTF is an API-neutral runtime asset delivery format developed by the Khronos Group 3D Formats Working Group.

[glTF 2.0](https://github.com/KhronosGroup/glTF/tree/master/specification/2.0) [https://github.com/KhronosGroup/glTF/tree/master/specification/2.0] GitHub repo and description.

[KHR_lights_punctual](https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Khronos/KHR_lights_punctual) [https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Khronos/KHR_lights_punctual] is an extension that defines a set of lights for use with glTF 2.0. Lights define light sources within a scene.

[MSFT_lod](https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/MSFT_lod) [https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/MSFT_lod] is an extension that adds the ability to specify various Levels of Detail (LOD) to a glTF asset.

[EXT_mesh_gpu_instancing](https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/EXT_mesh_gpu_instancing) [https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/EXT_mesh_gpu_instancing] is an extension that is specifically designed to enable GPU instancing which renders many copies of a single mesh at once using a small number of draw calls. It's useful for things like trees, grass, road signs, etc.

[FB_geometry_metadata](https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/FB_geometry_metadata) [https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Vendor/FB_geometry_metadata] is an extension that annotates glTF scene objects with a summary of the cumulative geometric complexity and scene-space extents of the scene's associated scene graph. **Editors note: While the computed total vertex and primitive count are metadata this is very limited metadata and may not meet the needs of the CDB X community.**

[KHR_materials_unlit](https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Khronos/KHR_materials_unlit) [https://github.com/KhronosGroup/glTF/tree/master/extensions/2.0/Khronos/KHR_materials_unlit] is an extension that defines an unlit shading model for use in glTF 2.0 materials, as an alternative to the Physically Based Rendering (PBR) shading models provided by the core specification.

[KHR_texture_transform](#) [https://github.com/KhronosGroup/gltf/tree/master/extensions/2.0/Khronos/KHR_texture_transform] is an extension that adds offset, rotation, and scale properties to textureInfo structures. These properties would typically be implemented as an affine transform on the UV coordinates.

10.6. CDB OpenFlight and glTF Feature Analysis

==

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Node Hierarchy	<p>Req 2 - Global Zone</p> <p>Req 3 - Model Zone</p> <p>Chap 6.5.6.1 - Model Landing Zone</p> <p>Req 17 - Model Footprint Zone</p> <p>Req 18 - Model Footprint Hierarchy</p> <p>Req 19 - Model Cutout Zone</p> <p>Req 20 - Model Cutout Geometry</p> <p>Req 21 - Model Pseudo-Interior Zone</p> <p>Req 22 - Model Interior Zone</p> <p>Req 4 - Layers</p>	<p>Chap 6.2 - Using hierarchy to organize models helps identify components of interest</p>	<p>Node Hierarchy fully supported</p>	No	Easy	Yes

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Comment Attribute	Req 14 - Zone Names Chap 6.5.3.1 - Zone Material Index Req 16 - Hot Spot Temperature Req 23 - Model Point (Damage) Req 24 - Model Point (DIS Origin) Req 25 - Model Point (View Point) Chap 6.6.2.3 - Model Attach Point Chap 6.6.2.4 - Model Anchor Point Chap 6.6.2.5 - Model Center of Mass Req 29 - Damage State (Levels) Req 36 - Model Vendor Attributes	Comment field is used to extend OpenFlight node for CDB specialization for many uses A number of constructs are available in the OF spec that could be natively encoded in the format instead of using the comment field.	"extras" property allows storage of an object with any number of named properties	No	Easy	

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Node Attribution	Req 13 - Model Zone Bounding Box	<p>Group node has bounding box attribute</p> <p>Other attributes available on the node:</p> <ul style="list-style-type: none"> Roof Flag Culture Footprint 	<p>Primitives have min/max properties.</p> <p>For general nodes, we could store this in the extras property</p>	No	Easy	
External Referencing	<p>Chap 6.2.4 - Model Master File</p> <p>Req 6 “Use XRef node to reference other files</p> <p>Chap 6.2.5.1 - Models straddling multiple files</p> <p>Chap 6.2.5.2 - Models with multiple LODs</p>	<p>Model Master File (comprised of xrefs) ensures a Model is seen as a single "object" even though its constituents are stored in separate files</p>	No external referencing	Yes	<p>Difficult</p> <p>The converter can parse external references and combine them into one gltf file (using mesh instancing for repeated references)</p>	<p>Partial</p> <p>The converter can parse external references and combine them into one gltf file (using mesh instancing for repeated references)</p>

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Level of Detail Node	<p>Chap 6.2.5.2 - Models with multiple LODs</p> <p>Req 26 - Significant Size</p> <p>Chap 6.8.1 - LOD node Exchange LODs</p> <p>Chap 6.8.2 - Additive LODs</p>	<p>LOD node contained in XRefs in master file</p> <p>Significant Size attribute on LOD</p> <p>LOD node support exchange and additive</p> <p>LOD strategies</p>	<p>Requires extension (MSFT_lod)</p> <p>No sig size.</p> <p>Works with switch in/out</p>	Yes	Medium	<p>Yes</p> <p>Uses MSFT_lod extension</p>
Switch Node	<p>Req 27 - Switch Masks (one per state)</p> <p>Req 28 - Switch Mask Names</p> <p>Req 29 - Damage State (Transition)</p> <p>Req 31 - Blur State (Transition)</p>	Switch node supports multiple masks. Each mask can be named.	No native support for this.	Yes	Medium	

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Degree of Freedom Node	Req 32 - Articulation Req 33 - Gimbal Limits	DOF node supports min/max limits for each degree of articulation (translation, scale, rotation)	Skins, Joints, Animations. glTF is more flexible/complex than OF when it comes to animated models	No	Medium	
Light Point Node	Req 35 - Model Light Points	Light Point node can represent individual points or light â€ stringâ™	No native support. Existing extensions for light sources but these are a different concept than light points	Yes	Difficult	
Projection	Req 1 - Specify Projections	Required projections for GTModel, GSModel, MModel and T2DModel	Could be specified in extras property at the scene level	No	Easy	

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Coordinate System	Req 7 - X (left/right), Y (front/back), Z (bottom/top) Req 8 - Origin (0,0,0)	These are native OpenFlight conventions Would recommend keeping glTF's axis system and adjusting the standard if needed.	glTF 2.0 uses a right-handed coordinate system, with x point right, y point up and z backward Changing this requires an extension and would reduce performance and interoperability	No	N/A	
Local Coordinate System	Chap 6.3.1.2	Transformation Matrix is used to specify LCS	Transformations on nodes	No	Easy	Yes
Units	Char 6.3.1.3	Header attribute is used to specify Units	Can be specified in extras property	No	Easy	

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Instancing	Req 11 - Avoid repeating identical pieces of geometry	Efficiency - smaller database size	Multiple nodes can instantiate the same mesh. However, there is no concept of node instancing. OF is more flexible	No	Medium	Yes
Mesh	Req 11 - Favor mesh over polygons	Efficiency - smaller database size, fewer graphics states	Mesh is supported and highly recommended over polygons. In OF many models use individual polygon nodes, but this would be inefficient in glTF. May lead to large geojson files.	No	Easy	Yes Polygon nodes at the same level with the same textures are merged into one mesh
Vertex Ordering	Req 11 - CCW order of verts define polygon front		GLTF uses CCW ordering of vertices	No	Easy	Yes

OpenFlight Construct	CDB Requirement/Guidelines (all from Chapter 6)	Notes	glTF Equivalent	New glTF extension required	Mapping OF to glTF difficulty	Implemented in converter prototype
Relative Priority	Req 12 - Layers of coplanar geometry	Relative Priority attribute at : <ul style="list-style-type: none">• Face• Mesh• Object• Group	Not supported natively. Could be stored in "extras"	No	Easy	
Textures	Req 37 - Textures stored in separate files from models Req 41 - Relative Texture Paths Req 42 - Object Shadow Attribute	Loading efficiency	Textures supported. Materials in glTF are similar to extended materials in OF, but not all layers from openflight exist in glTF. Ex: Light map, specular map, reflection map. Material textures are not a concept in gltf. Would require extension.	Yes	Hard	

==

==

==

==

Chapter 11. Supporting more than CDB 1.X:

OpenFlight capabilities that could be leveraged:

- Extensions
- Extended Materials
- Hotspots
- LOD Transitions
- Cultural Footprint
- Point Nodes (Model Points instead of using transforms)

OpenFlight to glTF converter

30 july 2020:

- Command line tool converting OpenFlight file into glTF
- Node hierarchy support and ported into glTF json file
- Polygons are converted to mesh on a per group basis
- No texture support yet
- LOD: using MSFT_LOD extension
 - LOD attribution for Significant size is not present
 - LOD attribution â€¢.

[image][C:\Users\hermann\AppData\Local\Microsoft\Windows\INetCache\Content.Word\gltfNodes.PNG]

glTF import in Blender

[image]

- LOD nodes (Extension) are not supported in Blender â€“ imported as just a node, with all 3 LODs visible at the same time (but could be separated as they are in different nodes)
- No node attribution

11.1. Original Prototyping Experiments - Keeping for now but will delete when this section is more mature.

Two profiles.

Profile 1 Storing the mesh in as 3D tile: Experiments

1. Question: How to store LODs? In glTF extensions, In 3D tiles, in separate glTF

2. Question: Are there missing glTF constructs?

Profile 2 Storing the mesh in as 3D tile (Editor's note: Why just 3D Tiles? Why not a more general approach that allows other encodings/approaches?)

1. Question: What is the ability to source edit the mesh?
2. Question: Mixing GeoPackage with 3D tiles?
3. Question: CDB catalog service for 3D tile inside a tile

Both Profile 1 and Profile 2

1. Question: What are the efficient format/packing for each different end-points:
 - a. ATAK
 - b. Mission command (wms/wfs)
 - c. Modsim
 - d. Gaming

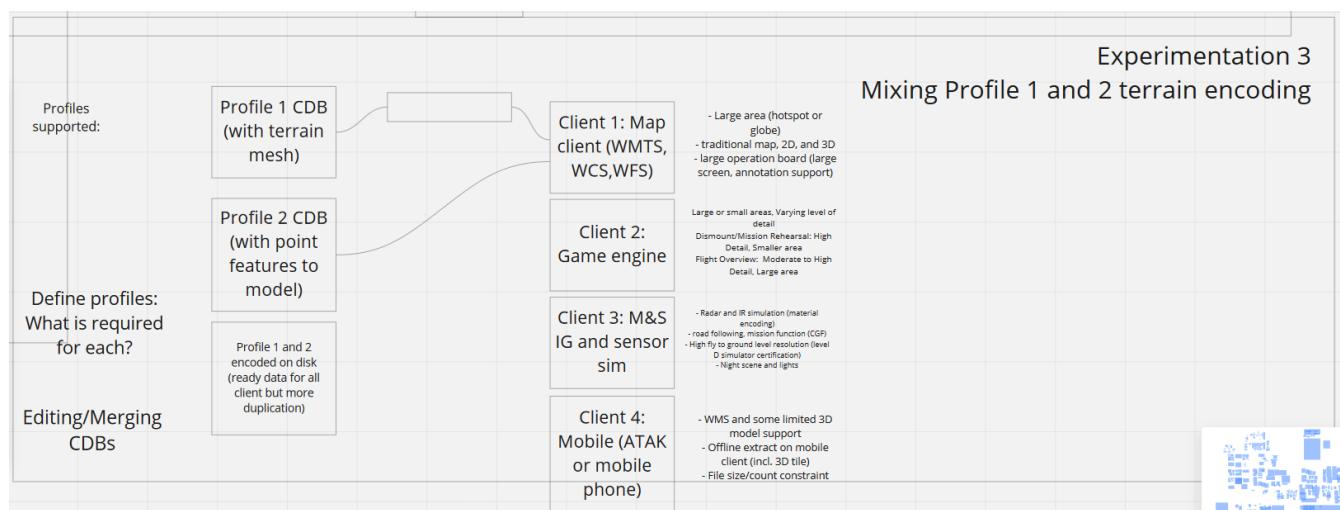


Figure ATTPh3 - 8. Profiles and Experiments for CBD-X 3D Models.

Chapter 12. CDB X Metadata and Integration

This section discusses metadata in CDB X

Summary of history of CDB X metadata discussions.

The earlier 'industry' versions, and OGC CDB 1.x versions of CDB contain a 'metadata' directory. Despite the directory name, this is a collection of .XSD (XML Schema Definition) and associated .XML (eXtensible Markup Language) files that are essentially controlled vocabularies for elements of the CDB standard.

In the OGC CDB Github "Schema" repo, the contents of the root level are:

	.git	4/28/2020 1:54 PM	File folder	
<input checked="" type="checkbox"/>	Metadata	2/23/2020 4:50 PM	File folder	
	Base_Material_Table.xsd	9/23/2019 10:48 ...	XSD File	4 KB
	Composite_Material_Table.xsd	9/23/2019 10:48 ...	XSD File	6 KB
	Configuration.xsd	9/23/2019 10:48 ...	XSD File	5 KB
	Datasets.xsd	2/23/2020 4:50 PM	XSD File	4 KB
	Defaults.xsd	9/23/2019 10:48 ...	XSD File	2 KB
	DIS_Country_Codes.xsd	9/23/2019 10:48 ...	XSD File	2 KB
	Feature_Data_Dictionary.xsd	9/23/2019 10:48 ...	XSD File	7 KB
	Lights.xsd	9/23/2019 10:48 ...	XSD File	3 KB
	Lights_Tuning.xsd	9/23/2019 10:48 ...	XSD File	3 KB
	Model_Components.xsd	9/23/2019 10:48 ...	XSD File	2 KB
	Model_Metadata.xsd	9/23/2019 10:48 ...	XSD File	12 KB
	Moving_Model_Codes.xsd	9/23/2019 10:48 ...	XSD File	3 KB
	OpenFlight_Model_Extensions.xsd	9/23/2019 10:48 ...	XSD File	5 KB
	README.md	9/10/2019 5:34 PM	MD File	1 KB
	ReadMe-cdb-1_1_0.txt	9/23/2019 10:48 ...	Text Document	2 KB
	Vector_Attributes.xsd	4/28/2020 1:41 PM	XSD File	8 KB
	Version.xsd	9/26/2019 11:57 ...	XSD File	7 KB
	xs3p.xsl	9/23/2019 10:48 ...	XSL Stylesheet	350 KB

Figure Mtd_Ph3 - 9. OGC CDB Schema Github Repo Root level files.

and the contents of the Metadata subdirectory are:

 Stylesheet	2/23/2020 4:50 PM	File folder	
 CDB_Attributes.xml	9/23/2019 10:48 ...	XML Document	64 KB
 CMT_Example.xml	9/23/2019 10:48 ...	XML Document	5 KB
 Configuration.xml	9/23/2019 10:48 ...	XML Document	3 KB
 Datasets.xml	2/23/2020 4:50 PM	XML Document	14 KB
 Defaults.xml	9/23/2019 10:48 ...	XML Document	10 KB
 DIS_Country_Codes.xml	9/23/2019 10:48 ...	XML Document	12 KB
 Feature_Data_Dictionary.xml	9/23/2019 10:48 ...	XML Document	783 KB
 Lights.xml	10/30/2019 4:12 ...	XML Document	75 KB
 Lights_Client.xml	9/23/2019 10:48 ...	XML Document	1 KB
 Materials.xml	9/23/2019 10:48 ...	XML Document	18 KB
 Model_Components.xml	9/23/2019 10:48 ...	XML Document	16 KB
 Moving_Model_Codes.xml	9/23/2019 10:48 ...	XML Document	19 KB
 Version.xml	9/26/2019 11:57 ...	XML Document	1 KB

Figure Mtd_Ph3 - 10. OGC CDB Schema Github Repo Metadata sub-directory files.

OGC CDB V1.1 is a minor revision of OGC CDB that provides for optional global and local metadata. The description of the new geospatial metadata provisions is in the Volume 1 Core document, in Section 5.1:

5. CDB Datasets

5.1. Controlled Vocabularies and Metadata Datasets used by CDB

5.1.1. Global Metadata

5.1.2. Local Metadata

5.1.3. Light Data Hierarchy Controlled Vocabulary

5.1.3.1. Client Specific Lights Definition Metadata

5.1.4. Model Components Definition File

5.1.5. Base Materials Table

5.1.6. Default Values Definition Table

5.1.7. Version Metadata

5.1.8. CDB Attributes Metadata

5.1.8.1. Definition of the <Attribute> Element

5.1.8.2. Definition of the <Unit> Element

5.1.8.3. Definition of the <Scaler> Element

5.1.8.4. Example of CDB_Attributes.xml

5.1.9. Geomatics and Vendor Attributes Metadata

5.1.10. Geospatial Metadata – Guidance

5.1.10.1. Suggested Global Geospatial Metadata Elements

5.1.10.2. Suggested Local Geospatial Metadata Elements

5.1.10.3. Where are local metadata files stored?

5.1.11. Configuration Metadata

5.1.11.1. A Note about Folder Path

5.1.11.2. Example

Figure Mtd_Ph3 - 11. OGC CDB V1.2 Volume 1 Table-of-Contents Excerpt.

For ease of reference, sections 5.1.10 thru 5.1.12 from Volume 1 of OGC CDB V1.2 are repeated here:

.1. Geospatial Metadata – Guidance

These are optional metadata files. This file is not included with the CDB distribution schema package.

Most metadata standards specify dozens of possible elements, such as author, that can be specified in a metadata encoding. This is why in many communities there are profiles that are applicable to the information sharing and discovery requirements of that community. For example, there are numerous profiles of ISO 19115:2013 Geographic information – Metadata. These include the INSPIRE, Defence NSG Geospatial Core metadata, and FGDC profiles. As such, the CDB standard

does not specify mandatory and/or optional metadata elements. Instead, a suggested set of minimal metadata elements are provided. The two lists – one for global and one for local – are based on an evaluation of mandatory elements in eight widely implemented metadata standards that are used in the geospatial and simulation communities. The one requirement is that all local metadata in a CDB data store provides the same mandatory elements as defined in the metadata standard specified in the Version metadata.

These following two sub-clauses recommend the metadata elements for global and local metadata. The use of F.1 refers to Table F.1 in ISO 19115-1:2014. Each element is identified by a general string followed by two element names. The first name is the DCAT name followed by the ISO 19115:2014 element name.

Suggested Global Geospatial Metadata Elements

Resource Identifier (dct:identifier, MD_Metadata.metadataIdentifier): 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 Digital Object Identifier (DOI). The **DOI** system provides a framework for persistent identification of electronic resources management of intellectual content, managing metadata, linking customers with content suppliers, facilitating electronic commerce and enable automated management of media.

Resource Title (dct:title [http://purl.org/dc/terms/title], CI_Citation.title): Title by which the resource is known (Table F.1). For global metadata for a CDB data store, this would be a name given to the entire data store. For example, this could be “Yemen demonstration CDB data store.”

Resource point of contact (dcat:contactPoint, (MD_Metadata.contact/CI_ResponsibleParty)): Name of the person, position, or organization responsible for the resource. (Table F.1). This is a text string. An example of a resource point of contact could be “Flight Safety” or “CAE.”

Resource reference date (dct:issued, CI_Citation.date): A date which is used to help identify the resource. (Table F.1). For global metadata, this is the date that the CDB data store was created or issued.

Resource Language (dct:language, PT_Locale): The language and character set used in the resource (if a language is used). (Table F.1) NOTE: We should recommend use of ISO 639-2 . For example, for English, the code would be “ENG.”

Geographic Location (dct:spatial, EX_GeographicBoundingBox): Geographic description or coordinates (latitude/longitude) which describes the location of the resource. Note: I think for the CDB standard that the definition should be narrowed to the bounding box of the contents of the data store. (Table F.1). We should also follow guidance from OGC OWS Common. See also 19115 annex B.3.1.2 Geographic extent information.

Resource abstract (dct:description ^[1], MD_DataIdentification.abstract): A brief description of the content of the resource (Table F.1).

Metadata date stamp (dct:issued, MD_Metadata.dateInfo): Reference date(s) for the metadata, especially creation. (Table F.1). Note: Date gives values for year, month and day. Character encoding of a date is a string which shall follow the format for date specified by ISO 8601. This class is documented in full in ISO/TS 19103.

Temporal Extent information for the dataset (dct:temporal, EX_TemporalExtent): The temporal extent of the resource. For a CDB data store, this would be the temporal range of when the data store was initially created to the point where the most recent content was created.

Constraints on resource access and use (dct:accessRights, MD_SecurityConstraints): 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 EDH compliant encoding.

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.

Suggested Local Geospatial Metadata Elements

Local Geospatial metadata can be stored in a number of different folder locations based on the data resource (data set) for which the metadata is associated. For instance, metadata for vector data will be stored at the LoD/tile level. Metadata for a moving model would be stored in the same folder using the same path name as the actual model definition. See Clause 5.1.2 above for examples.

- + While the same metadata elements are recommended for both global and local geospatial metadata, there are some differences that should be considered.
- + *Metadata Reference Information* (dct:identifier, MD_Metadata.metadataIdentifier) This is a unique identifier for the dataset. In CDB, this could be the pathname to the dataset or the tile. These pathnames are unique. Using such identifiers would facilitate development of a RESTful API for discovery and access of CDB resources.
- + *Resource Title* (dct:title [<http://purl.org/dc/terms/title>], CI_Citation.title): Title by which the data set is known (Table F.1). For local metadata, this could be a name given to a layer or model in the data store. In a CDB data store, at the dataset or tile level this would be a name given to the resource, such as “county soils.”
- + *Resource point of contact*: Name of the person, position, or organization responsible for the resource. This is a text string. An example of a resource point of contact for the content for a given layer and tile could be “Ordnance Survey.”
- + *Resource reference date* (dct:issued , CI_Citation.date): A date which is used to help identify the resource. For local metadata, this could be the date that the tile content was created in the CDB data store or the date a moving model was added to the data store
- + *Spatial Resolution Information* (No equivalent, MD_Identification.spatialResolution): The nominal scale and/or spatial resolution of the resource. This description can include LoD information. Note: This is not precision! Precision is more about the number of decimal places and not the accuracy of the resource.

Where are local metadata files stored?

Typically, local metadata files will be stored with the physical data. For GTModel geotypical data sets, the metadata file would be stored along with the model XML file. If the model is stored in

multiple LoDs, the metadata would also be stored at each LoD. For tiled vector data, the local metadata would be stored with the vector files at the tile level. Please see the examples in Section 3.0 for more detail.

Phase 3 Consensus, Day 3

CDB will be self-describing
in a top-level metadata file
(and possibly at sub-levels
also)

Tracey
David
Mike

Figure Mtd_Ph3 - 12. Day 3 Consensus and formation of the Metadata / Integration sub-team.

Phase 3, Day 4

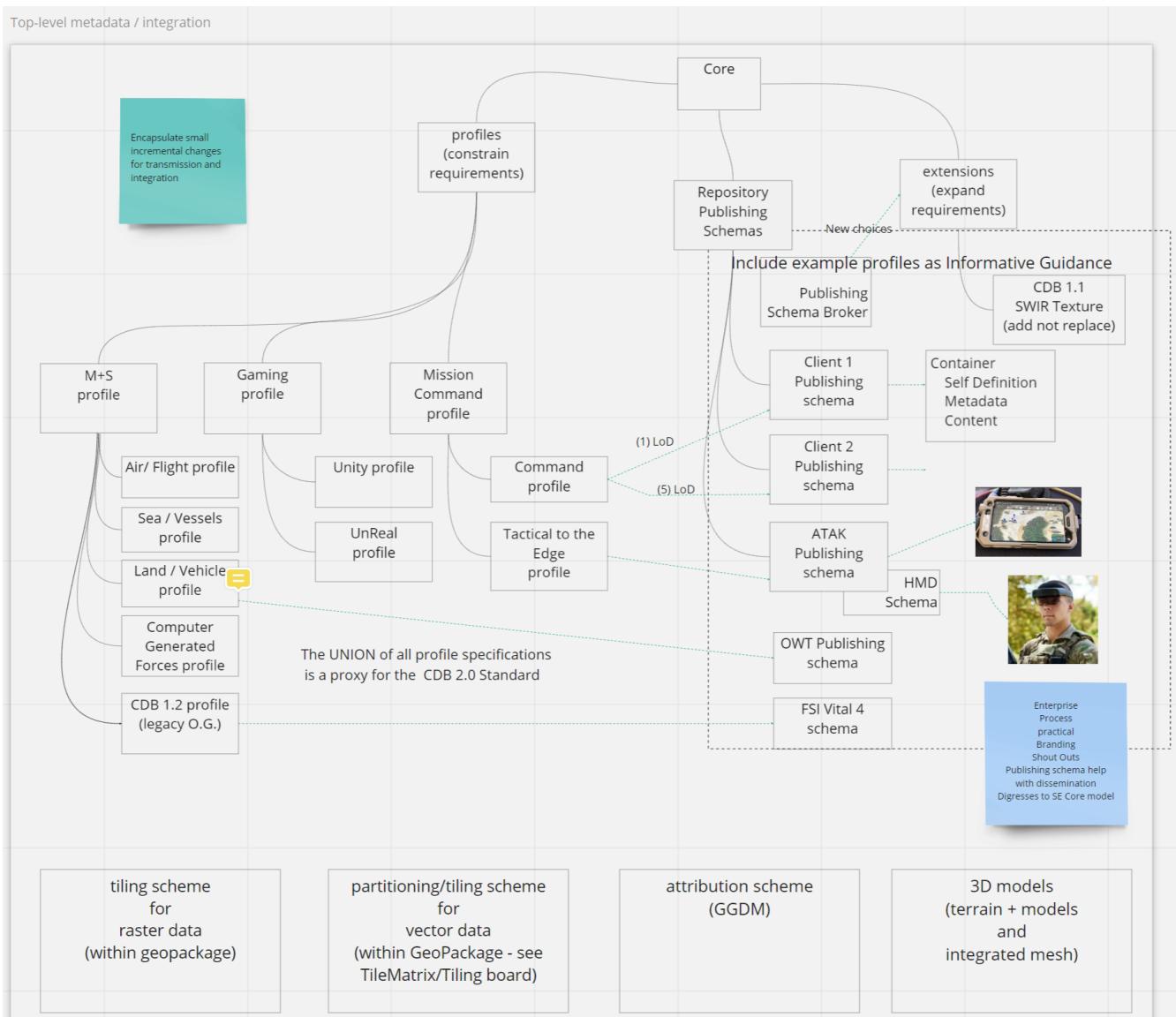


Figure Mtd_Ph3 - 13. Metadata Day 4 Whiteboard.

Phase 3, Day 5

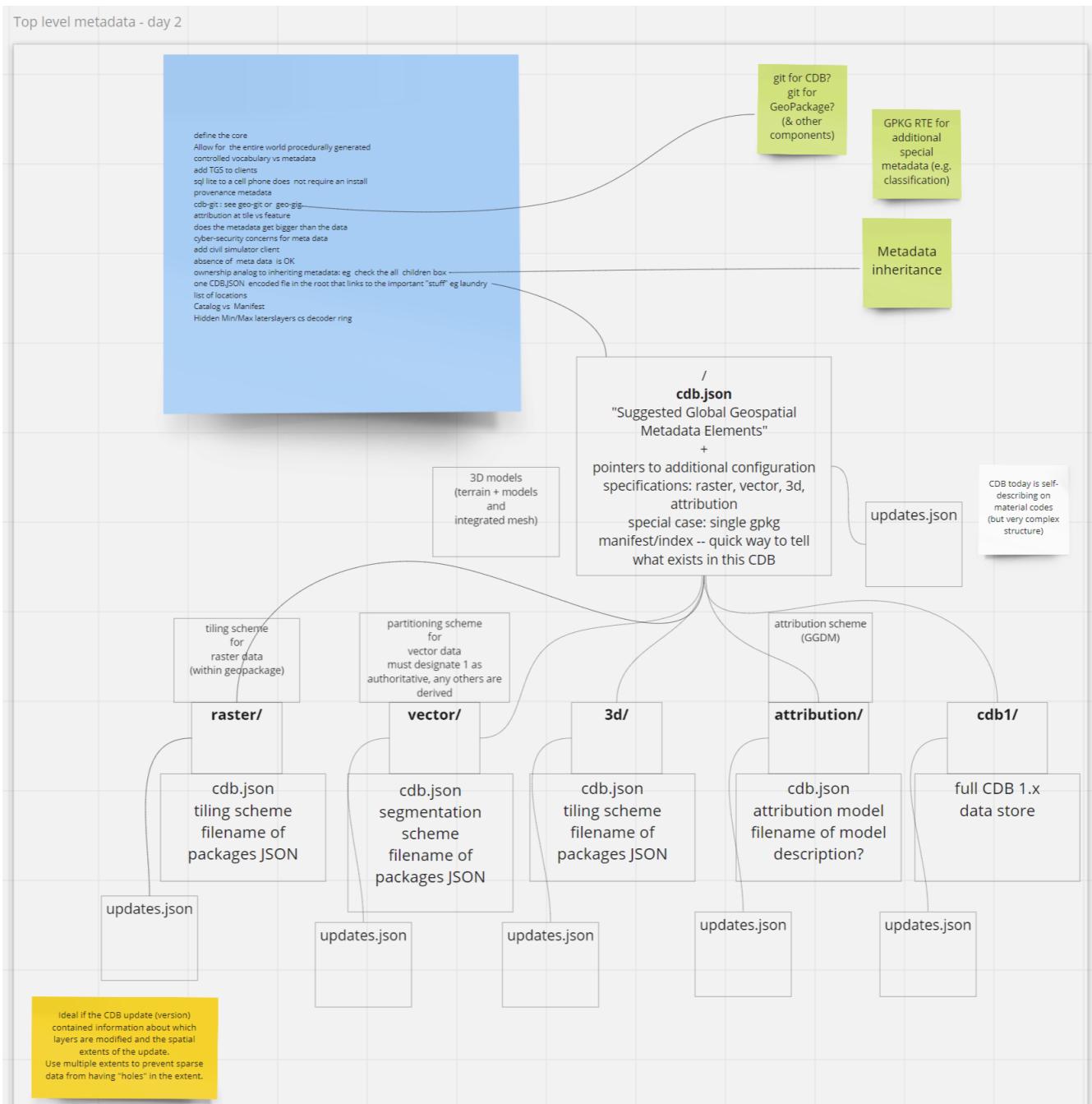


Figure Mtd_Ph3 - 14. Metadata Day 5 Whiteboard.

Plan for Experimentation in subsequent phases

Chapter 13. Tiling and Tiled Coverages

[1] DCAT does not have a concept “abstract”. Use description instead.

Chapter 14. CDB X Tiling Design Principles, Requirements, Tiled Data Conceptual Model and GeoPackage

This clause provides discussion of tiled data and GeoPackage requirements and the results of the prototyping activities.

14.1. General

Any tiling schemes specified in a CDB X data store (repository) SHALL be based on and consistent with the:

- [OGC Core Tiling Conceptual and Logical Models for 2D Euclidean Space](https://portal.ogc.org/files/?artifact_id=92962&version=1) [https://portal.ogc.org/files/?artifact_id=92962&version=1] (19-014r3)
- [OGC Two Dimensional Tile Matrix Set Standard](https://www.ogc.org/standards/tms) [https://www.ogc.org/standards/tms] (17-083r2)

14.1.1. OGC Core Tiling Conceptual and Logical Models for 2D Euclidean Space

This OGC Abstract Specification consists of two Parts: A General Tiling Conceptual Model and, based on the Conceptual Model, a Logical Model for the Tessellation (Tiling) of 2D Euclidean Space. Tiling of 2D Euclidean space is the most commonly known approach to partitioning space in traditional geospatial technology. However, there are also common elements and/or semantics for any approach to partitioning space in any dimension. The logical model in this document defines a set common required elements and then follows with more specific requirements for the two dimensional case.

Part 1 of the Abstract Specification describes a general tiling conceptual model. The conceptual model is applicable to any dimension. The conceptual model makes no assumptions regarding content, use cases, implementation scenarios, or how the space is to be tessellated (tiled). The conceptual model is abstract and cannot be implemented as is.

Part 2 of the Tiling Abstract Specification defines a detailed logical model for the tessellation of 2D Euclidean Space. One or more logical models are required to provide the requirements and structure necessary for implementation. Therefore, in addition to the conceptual model, this Abstract Specification also specifies a core logical model for the 2D planar (Euclidean) use case.

14.1.2. OGC Two Dimensional Tile Matrix Set Standard

The OGC Tile Matrix Set standard defines the rules and requirements for a tile matrix set as a way to index space based on a set of regular grids defining a domain (tile matrix) for a limited list of scales in a Coordinate Reference System (CRS) as defined in [OGC 08-015r2] Abstract Specification Topic 2: Spatial Referencing by Coordinates. Each tile matrix is divided into regular tiles. In a tile matrix set, a tile can be univocally identified by a tile column a tile row and a tile matrix identifier. This document presents a data structure defining the properties of the tile matrix set in both UML

diagrams and in tabular form. This document also presents a data structure to define a subset of a tile matrix set called tile matrix set limits. XML and JSON encodings are suggested both for tile matrix sets and tile matrix set limits. Finally, the document offers practical examples of tile matrix sets both for common global projections and for specific regions.

14.2. Design Objectives for CDB X Tiling and LoDs and Layers

The following are the key design principals for a CDB X tiling, levels of detail, and layering.

- OGC GeoPackage structured containers are a primary storage format for derived vector data layers, coverages, and other data types as identified (**TBD**). The two OGC Standards of relevance are [OGC GeoPackage version 1.1](https://portal.opengeospatial.org/files/12-128r15) [<https://portal.opengeospatial.org/files/12-128r15>] and [OGC GeoPackage Extension for Tiled Gridded Coverage Data](http://docs.opengeospatial.org/is/17-066r1/17-066r1.html) [<http://docs.opengeospatial.org/is/17-066r1/17-066r1.html>].
- Have metadata for whole dataset listing how data layers are regrouped in GeoPackages (LOD grouping and data layer grouping)
- Store imagery in GeoPackage according to the tiling scheme specified below (**need anchor**)
- Store coverage data in GeoPackage according to the tiling scheme specified below (**need anchor**). Suggested coverage types based on current CDB standard and user specified requirements: (Spherical body surface): Elevation models, imagery, multi-spectral, raster (such as classified satellite imagery). Should be extensible to support other types.
- Coordinate Reference System is WGS 84 with epoch encoding (same as current CDB Standard except for epoch).
- Need to keep the various encodings that CDB 1.x already has:
 - Integer Elevation
 - 8-bit Raster Material Mixtures
 - Imagery Compression:
- Must enable a "relatively" easy migration path from the CDB 1.1/1.2 tiling/LoD structure into the CDB X structure.
- CDB X must support integer compressed heights (elevations)
- CDB X must support GeoTIFF as an encoding for images.

14.3. Proposed Tiling Logical Model for CDB X

The following figure, based on the Tiling Abstract Specification Logical Model and the CDB 1.x Conceptual Model, shows the properties by class (concept) and the relationships between the classes.

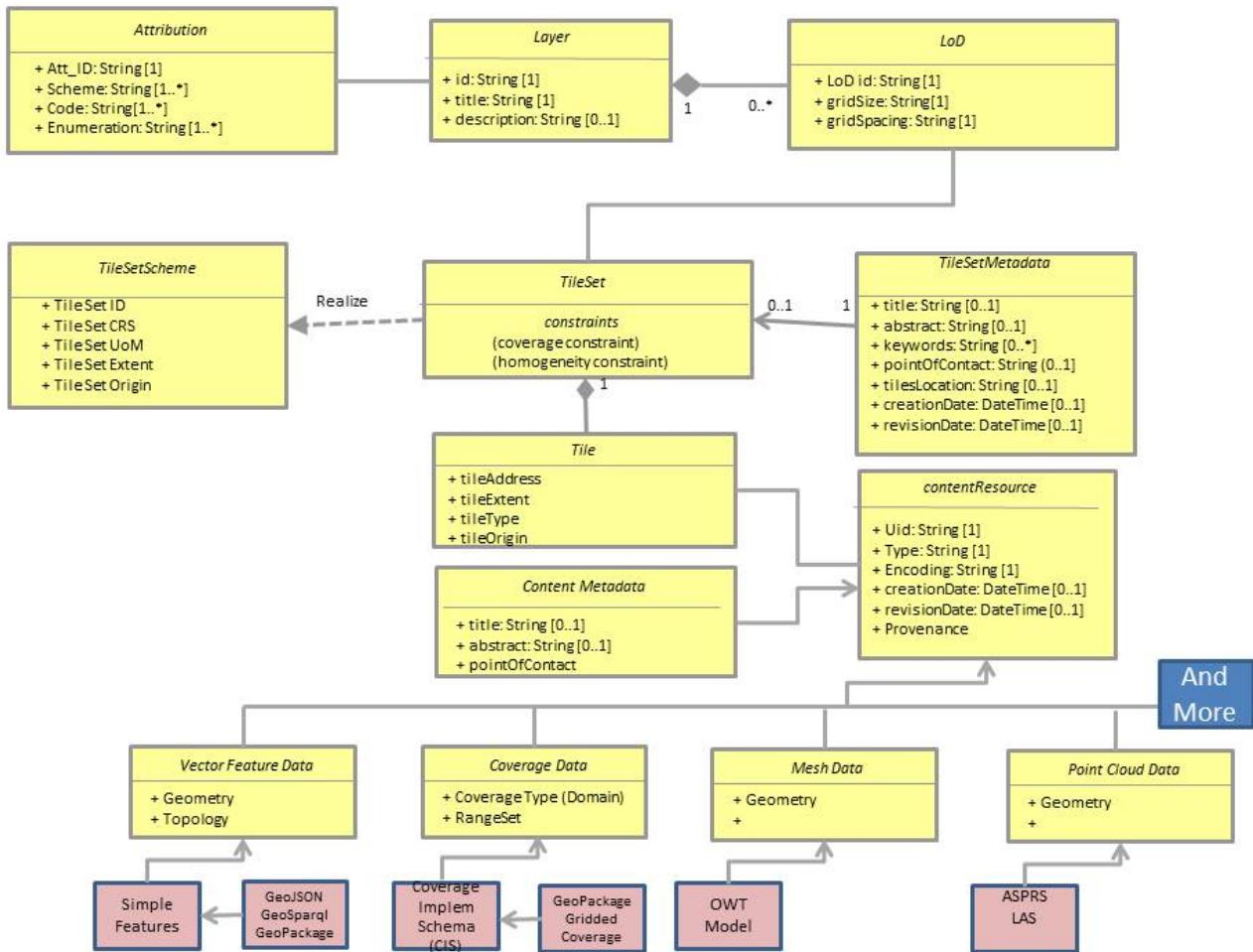


Figure Mtd_Ph3 - 15. Logical Model for partitioning based on tiles in CDB X.

14.4. Proposed CDB X Tiling Structure

This section describes the proposed tiling structure and LoD (levels?) structure in CDB X. (Probably need small section on consistency with version 1.x?)

Using GPKG Tiled Gridded Coverage standard. Discussed "materials" and their requirements as coverages. Suggest a change request to allow 8, 16 bit and 32 TIFF (integer or float). (Not only for materials, but also heights (aka elevation))

For any CDB or data layer, pick one of: CDB 1.x grid or GNOSISGlobalGrid (better equal area approximation for polar regions, constant tile size even at overview LODs)

Current GPKG TMS draft: [GPKG TMS Extension \(Draft\)](#) [<https://gitlab.com/imagemattersllc/ogc-vtp2/-/tree/master/extensions>]

- TileMatrixSet compatibility

(topLeftCorner, Variable width)

- Reference same TMS from multiple layers

Sample GeoPackage using TMS / GNOSISGlobalGrid for both elevation & vector data tiles: https://portal.ogc.org/files/?artifact_id=92565

14.5. Additional CDB X Recommendations

The following are recommendations and suggested additional discussion topics. These recommendations and discussion topics resulted from the Tiling sub-groups discussion on an enhanced tiling model for CDB X and the potential impacts on the various data types (layers) in the current CDB standard and existing CDB data stores.

14.5.1. Elevation min/max

CDB X needs to continue supporting the Min/Max Elevation component concept. In order to reduce the number of files and complexity, the recommendation is to move the minimum and maximum elevation values for the gridded coverage contained in a tile to the tile metadata. Note: The MinElevation and MaxElevation components are part of the MinMaxElevation dataset whose purpose is to provide a CDB conformant data store with the necessary data and structure to achieve a high level of determinism in computing line-of-sight intersections with the terrain. The values of each component are with respect to WGS-84 reference ellipsoid.

14.5.2. Image Compression - JPEG

Recommendation: That loss-less image compression solutions be explored for use in CDB X. Any such solutions are not viewed as a replacement for JPEG 2000 but instead as alternatives. This could be accomplished by submitting a change request for the OGC GeoPackage standard that provides guidance and requirements for support of other image formats beyond PNG and JPG. The subgroup identified a potential candidate: [FLIF - Free Lossless Image Format](https://flif.info/) [<https://flif.info/>]. NOTE: JPEG-2000 has very high compression, even in lossless mode, and there are multiple open-source implementations. However, performance can be extremely slow and non-optimal for all use cases.

14.5.3. Materials

Recommendation: CDB X needs to support material data to provide the same functionality as CDB 1.x. To also reduce the number of files, this can be accomplished by putting all the raster material data (including material table) in a single CDB data layer in GeoPackage, perhaps using the related tables extension. The subgroup did have some discussion on what "materials" means in the CDB 1.x context. Materials in current CDB have to do with reflectance in wavelengths other than what the human eye senses. These are for non-visualization use cases or special visualization such as IR. The subgroup did also discuss for the possible need for CDB X to provide guidance on using Physically-Based Rendering (PBR) to support the visualization/rendering use case. glTF, I3S, and 3D Tiles all support PBR.

14.6. Findings from experiments

The following graph compares the CDB 1.x zones with the Gnosis grid and shows how the GNOSIS algorithm helps to keep the typical tile closer to a “square” than CDB’s zones.

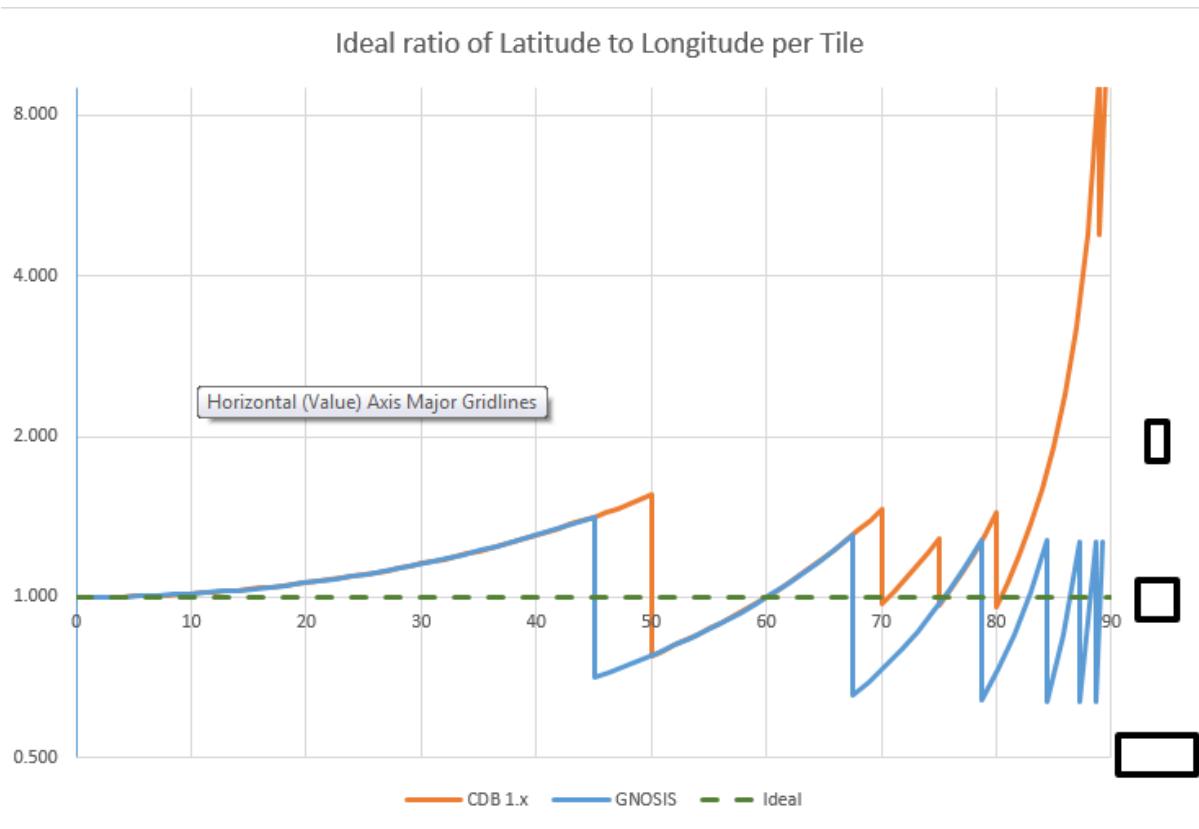


Figure Mtd_Ph3 - 16. CDB 1.x to GNOSIS Comparison - Ratio.

14.7. CDB X Tiling/GeoPackage Experiment #1 (Ecere)

14.7.1. Data and GeoPackage Structure

A 1.4 GB GeoPackage of the Camp Pendleton sample CDB from Presagis (originally used in OGC Testbed 13), along with an accompanying `cdb.json` can be found at:

<http://maps.ecere.com/CDBX/X1/>

In the first Ecere experiment, the camp Pendleton data was stored in a single GeoPackage using the "null grouping" mode, i.e. everything stored in a single GeoPackage. Potentially even the `cdb.json` could be included inside as metadata (using the GeoPackage metadata extension) to make this GeoPackage very portable.

Inside the GeoPackage, all layers were tiled using the Ecere [GNOSIS Global Grid](https://maps.ecere.com/ogcapi/tileMatrixSets/GNOSISGlobalGrid) [<https://maps.ecere.com/ogcapi/tileMatrixSets/GNOSISGlobalGrid>]. This approach was implemented for the experiment using the proposed (and early draft) [GeoPackage Two Dimensional Tile matrix Set \(TMS\) extension](https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/14-tile-matrix-set.adoc) [<https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/14-tile-matrix-set.adoc>]. This extension was defined and initially tested in the [2019 OGC Vector Tiles Pilot Phase 2](https://www.ogc.org/projects/initiatives/vtp2) [<https://www.ogc.org/projects/initiatives/vtp2>].

The content includes imagery data, stored as JPEGs, and terrain data stored as GeoTIFFs. This was accomplished using an implementation of the [OGC GeoPackage Tiled Gridded Coverage Extension](https://docs.opengeospatial.org/is/17-066r1/17-066r1.html) [[http://docs.opengeospatial.org/is/17-066r1/17-066r1.html](https://docs.opengeospatial.org/is/17-066r1/17-066r1.html)].

The content also included the Natural and Environmental light features vector layers. For this experiment, this content was stored using Mapbox Vector Tiles. This included points intended to

reference 3D models such as "Man-made point features" and "Tree point features".

The early draft GeoPackage Tiled Vector Data (vector tiles) extensions were used for this:

1. [GeoPackage Tiled Vector Data Attributes Extension](https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/4-vtae.adoc) [https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/4-vtae.adoc] This extension defines a relationship between features contained in a tiled layer and tiles containing those features.
2. [GeoPackage Tiled Vector Data](https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/1-vte.adoc) [https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/1-vte.adoc] The GeoPackage Tiled Vector Data extension defines the rules and requirements for encoding tiled feature data (aka "vector tiles") into a GeoPackage data store.
3. [MapBox Vector Tiles extension](https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/2-mvte.adoc) [https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/2-mvte.adoc] The GeoPackage Mapbox Vector Tiles extension defines the rules and requirements for encoding vector tiles in a GeoPackage data store as Mapbox Vector Tiles.

NOTE Ecere is planning to add the 3D models in the next experiments — for geo-specific, both one glTF for a whole tile, as well as individual models to be referenced by the points.

For the case of points referencing the 3D models (best suited for geo-typical), those glTF files would be stored in a single 3D models table, as well as a textures table (if the models share rather than embed textures). The related table extension would be used to relate the features attributes.

For the geo-specific / one glTF for the whole tile, the glTF could potentially be stored in a tiles table instead, and the models constituting the payload (much like raster or vector tiles).

14.7.2. OGC API access demo

At this address: <https://maps.ecere.com/ogcapi/collections/CDBX:X1:CampPendletonX1.gpkg>

You can directly access this CDB X Experiment #1 GeoPackage through our GNOSIS Map Server, including rendering maps, downloading coverages, accessing as tiles in different tiling schemes, accessing individual vector features, retrieving them as (re-merged) GeoJSON, visualizing them on GeoJSON.io and so on. To some extent, this demonstrates that even though the data is tiled, this layout actually supports a wide range of use cases.

14.7.3. Visualization

Ecere software can currently visualize the CDB X/GeoPackage elevation and imagery directly in the Ecere 3D visualization tool (albeit with a few glitches at the moment at terrain tile edges).

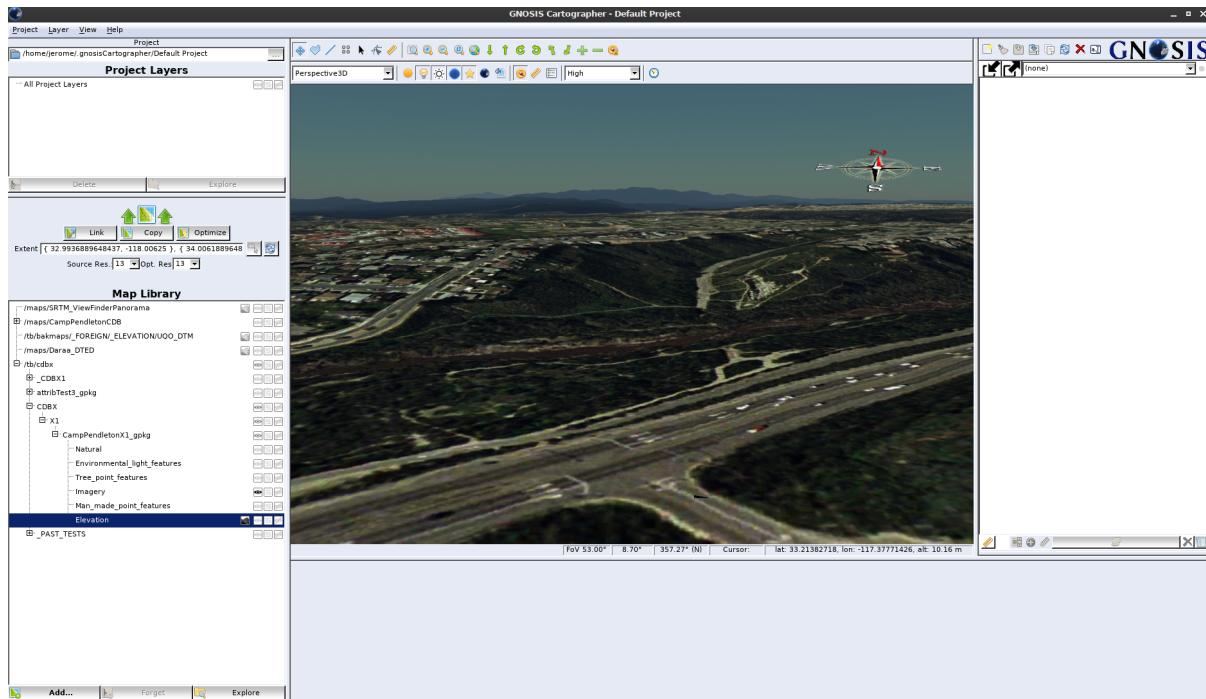


Figure Mtd_Ph3 - 17. Ecere Camp Pendleton GeoPackage in CDB X.

14.7.4. Live Cesium JS / 3D Tiles demonstation

Navigate to <https://sandcastle.cesium.com/> and copy/paste the following four (4) lines of Java Script code:

```
var worldTerrain = Cesium.createWorldTerrain({requestWaterMask: true,
requestVertexNormals: true, });
var viewer = new Cesium.Viewer("cesiumContainer", { terrainProvider: worldTerrain });
var scene = viewer.scene;
var tileset = scene.primitives.add(new Cesium.Cesium3DTileset({ url:
"https://maps.ecere.com/ogcapi/collections/CampPendletonCDB:Buildings/3DTiles/tileset.
json" }));
```

Then click "Run", and zoom in onto Camp Pendleton. Camp Pendleton is at the height of the southern part of the northernmost of the 2 islands just west of southern California. As the image is zoomed, 3D buildings should come into view.

This tiled 3D distribution (for OGC API collection <http://maps.ecere.com/ogcapi/collections/CampPendletonCDB:Buildings>) is currently being generated on the fly from the Ecere GNOSIS Data Store / E3D models. NOTE: No textures have been added yet.

Once the 3D models are added to the CDBX GeoPackage, it should be possible to stream as 3D Tiles <CESIUM??> straight from the CDBX/GeoPackage as well, rather than the original CampPendleton CDB repository.

14.7.5. Next steps (this section will change as new content is provided)

- Support for splitting in multiple GeoPackages with LoD grouping
- A table for storing glTF models, for referenced 3D models (either only for geo-typical trees, or

even the geo-specific buildings as well), and using Related tables extensions to relate the models table to other tables

- Single glTF models covering a whole tile for geo-specific models
- Export similar GeoPackages for San Diego CDB
- Attribution per model within the single tile model. We support this directly in E3D, and I wonder whether glTF2 supports this. I know the main thing that a batched 3D model 3D Tile adds in addition to a glTF is a Features Table which does precisely this, so I am not sure whether glTF 2 has this capability built in (i.e. allowing to use a .glb directly rather than a .b3dm).
- Support for visualizing the dataset including 3D models in our GNOSIS Cartographer client (work required to support CDB consisting of multiple of GeoPackages as a single data source)
- Support for GNOSIS Map Server streaming 3D models from CDB X/GeoPackage

14.8. FlightSafety GeoPackage Tiling Experiments

Setup: The data used for these experiments are primarily freely available, and include the following * Blue Marble (NASA) that was georeferenced using GDAL - <https://visibleearth.nasa.gov/collection/1484/blue-marble> * The high resolution inset is from USGS downloads of Central Park in New York City

Tiling Scheme: The tiling scheme uses the [GNOSIS Global Grid](<https://maps.ecere.com/ogcapi/tileMatrixSets/GNOSISGlobalGrid>) (using TMS extension—<https://gitlab.com/imagemattersllc/ogc-vtp2/-/blob/master/extensions/14-tile-matrix-set.adoc>). We are using the same type of json file that Ecere is using in their experiment.

LOD Grouping The grouping is pre-set per experiment. The groups are calculated from the highest LOD, back to coarser LODs. For example, if there are 7 LODs (0-6) and a grouping of 4, then LODs 3 through 7 are in one GeoPackage, and LODs 0 through 2 are in another GeoPackage.

Directory and Naming Scheme Each top level tile is within a directory that encodes the LOD, the row (rows are counted from the top, so north to south), and the column (longitude west to east). For example, "L0_R1_C2". Each tile directory contains one GeoPackage file (for example "Imagery_L0_L2_R1_C2.gpkg") and all the tile directories that refine this area (such as "L3_R9_C22"). There were two intentions to this directory structure:

- Limit the number of files in a directory (to keep from running into OS limitations).
- Make it a bit easier to export a portion of the world by hand from one CDB X to another.

14.8.1. FlightSafety Experiment 1

Purpose of Experiment

This experiment is to show how the top levels of the tiling scheme work, to show the LOD groupings within multiple GeoPackage files, and to show the proposed directory and file naming. There are 8 top level tiles (2 rows and 4 columns), and all GeoPackages that refine one of these tiles is under that tile's directory structure.

Processing

This experiment uses the NASA Blue Marble imagery to approximate world-wide imagery at a high level. This provides 7 levels of detail of data (L0 to L6). Normally, the GeoPackage files should be larger for efficient use, but to show the LOD groupings, only 4 LODs are grouped together. Also, the imagery is stored as Jpeg, so that tools can view the imagery easier. We originally were creating Jpeg2000 files, but it was harder to check the results in a tool like "DB Browser for SQLite". The data size for this experiment is around 300 MB.

Data Location

Compressed 7-zip file at: https://drive.google.com/drive/folders/1zfdBnAHpf9McLTbaJotdKGR_tXQad4dy?usp=sharing

14.8.2. FlightSafety Experiment 2

Purpose of Experiment

This experiment is to further test the limits of the LOD grouping and directory organization. It will be the World CDB X from experiment 1 with a small higher resolution inset of imagery. Added was some 15m data at LOD level 12 covering New York City, and 2 ft imagery covering Central Park on Manhattan Island at LOD level 16.

Processing

Same processing as experiment 1, but with an LOD grouping of 6 (thought during our planning to be ideal balancing size and number of sub-directories ($(2^6)^2 = 4096$ maximum directories within one folder). The maximum LOD for this experiment is 16 (60cm). To find the highest resolution data, look at file CDBX_highres\Imagery\L0_R0_C1\L5_R17_C37\L11_R1120_C2412\Imagery_L11_L16_R1120_C2412.gpk. The data size for this experiment is almost 1.5 GB.

Data Location

Compressed 7-zip file at: <https://drive.google.com/drive/folders/1zKuu0oQy3K5oKestuPIvL-BRiSPn9oaO?usp=sharing>

14.8.3. Observations

- The file names and directory names are pretty hard to read and understand by looking at the files. But since the tiles are rarely on a "geocell" boundary, their might not be a good naming scheme.
- Creating the LOD groupings based on the highest LOD of data makes it difficult to add data of a higher resolution later on. It might also make it harder to create "Versions" of the data that have been updated.
- There are a lot of directories created with this tiling and naming scheme. In general, there is a 1-to-1 ratio of files to directories.

For the GeoPackage CDB Interoperability Experiment, FlightSafety strictly tested existing Shapefile

performance vs GeoPackage performance from a flight simulation runtime perspective. So how fast can we pull data out of a GeoPackage in the same manner that we process shapefiles. GeoPackage creation timing wasn't that interesting to us at the time. We were also testing the different approaches put forward on how to structure the GeoPackage to contain the data, whether that was one GeoPackage containing a single shapefile's features, one GeoPackage table containing all the features at a given level of detail, or one GeoPackage table containing all the features at all the levels of detail. Changing the CDB conceptual model or levels of detail was outside the scope of this project.

FlightSafety found that querying large numbers of features out of very large tables was dependent on how the GeoPackage was structured. The original conversion of the field columns queried was to a string, making these fields integers made it faster, but creating an index on those columns was an order of magnitude faster. The table here shows the three different GeoPackage organizations, how many features ended up in the table we are querying, and how fast we could pull out 16k of rows from that table. Each test pulled out the same set of 16k rows. The highlighted data shows how structuring the GeoPackage matters when it comes to performance.

A	B	C	D	E	F	G	H	I	J	K	L	M
					SQL Query on String Fields		SQL Query on Integer Fields		SQL Query on Indexed Integers			
	Test Number	Features Returned in Query	Total Features in Table	% Returned	Per Test	Normalized (ms)	Per Test	Normalized (ms)	Per Test	Normalized (ms)	Normalized to Opt 1	
1	1	16,384	16,384	100.00%	96.60	1.00	87.34	1.00				
2	2	16,384	3,375,935	0.49%	1125.67	11.65	1330.79	15.24	138.17	0.63		
3	3	16,384	6,865,325	0.24%	3025.29	31.32	2677.78	30.66	173.90	0.50		

0 All features contain the row and column and lod from the CDB tile they were within before the conversion to GeoPackage
1 Timing was averaged over multiple runs, and includes the I/O of opening and closing the GeoPackage
2
3 First Test Query: `SELECT * FROM table_name`
4 Second and Third Test Query: `SELECT * FROM table_name WHERE ROW = x AND COL = y AND LOD = l`
5
6 Index for the last timing column created with: `CREATE INDEX '101_GTFeature_S002_T001_idx' on '101_GTFeature_S002_T001' (LOD, ROW, COL)`

Figure Mtd_Ph3 - 18. Performance.

Annex A: Revision History

replace below entries as needed

Table 1. Revision History

Date	Editor	Release	Primary clauses modified	Descriptions
June 2020	C. Reed	.1	all	Clone template and initial version

Annex B: Bibliography

- [1] STAC: [Spatio Temporal Asset Catalogs](https://stacspec.org/). [https://stacspec.org/] (2017).
- [2] [Ground-Warfighter Geospatial Data Model \(GGDM\)](https://ggdm.erdc.dren.mil/) [https://ggdm.erdc.dren.mil/] (2020)