Delivering Military Advantage through multi-national geospatial interoperability

DGIWG 126

Edition: 1.1

Dated 2 May 2025

DGIWG GeoPackage Profile

Document type:	Standard
Document Reference:	STD-DP-19-005
Document subtype:	Implementation profile
Document date:	2 May 2025
Edition number:	1.1
Document Status:	DP
Supersedes:	1.0
Responsible Party:	Defence Geospatial Information Working Group, Geospatial Web Services Program (P5)
Audience:	This document is approved for public release and is available on the DGIWG website, https://www.dgiwg.org
Abstract:	This document is a profile of OGC 12-128r19, OGC GeoPackage Encoding Standard, Version 1.4, dated 02-06-2024. It defines specific Defence requirements, recommendations and guidelines for interoperability between producers and consumers of geospatial content in the GeoPackage file format for use by DGIWG member countries.
Copyright:	

Table of Contents

Tables

Figures

i. Executive Summary

This profile of the Open Geospatial Consortium (OGC) GeoPackage standard enables DGIWG member nations producing and sharing geospatial raster and feature data to discover, understand, and utilize GeoPackages in a consistent interoperable format.

A GeoPackage is an open, standards-based, platform-independent, portable, self-describing, compact format used as a self-contained, run-time geospatial database and for transferring geospatial information. The OGC GeoPackage standard describes a set of conventions for storing the following within an SQLite database:

- Vector features
- Tile matrix sets of imagery and raster maps at various scales
- Gridded Coverages (elevation and other gridded data)
- Extensions

To ensure this profile can be implemented, and in order to distinguish between requirements, recommendation, and future work directions, the existing constraints identified by the submitting organizations and vendors have been taken into account. Requirements include both extensions/restrictions of the OGC GeoPackage standard as well as system requirements in order to enable interoperability by appropriately configuring existing software. System requirements are intended to be applied in the design of systems requiring services compliant with this profile. Requirements are associated with conformance tests which provide guidelines for testing the compliance of implementations of this profile.

ii. Contributing nations and organizations

Nation	Organization
Austria	Institute of Military Geography
France	Institut National de l'Information Géographique et Forestière (IGN)
Germany	Bundeswehr Geoinformation Centre (BGIC)
Sweden	Swedish Defence Materiel Administration
United Kingdom	Ministry of Defence (MOD)
United States	National Geospatial-Intelligence Agency (NGA)

iii. Document points of contact

All questions regarding this document shall be directed to the editor (secretariat@dgiwg.org [mailto:secretariat@dgiwg.org]) or the contributor organizations.

iv. Revision history

Date	Release	Primary clauses modified	Description
09/22/2023	1.0	Approved. Panel updated post-ballot to publish.	Publication of 1.0
05/02/2025	1.1	Addition of Gridded data coverages. Approved. Panel updated post-ballot to publish.	Publication of 1.1

Edition 1.1 Note. 1.1 adds requirements and descriptive text for using the GeoPackage gridded data extension. Edition 1.1 is a profile of OGC GeoPackage version 1.4 [4], updated from version 1.3.1 in Edition 1.0. The conformance in DGIWG-126 edition 1.1 for the Core GeoPackage requirements and the additional conformance for gridded data requirements are both identified as version 1.1. Conformance to edition 1.0 is otherwise the same as edition 1.1. See Editor note in 3.1 Conformance Classes for conformance changes in edition 1.1.

v. Future Work

None defined.

1 Introduction

This Interoperability Standard creates a Defence Geospatial-Information Working Group (DGIWG) profile based on the OGC GeoPackage based on the OGC GeoPackage Encoding Standard 1.4 ([4] OGC 12-128r19).

In this profile document, when the term "GeoPackage" is used without "OGC" or "DGIWG" preceding, the term refers to characteristics of the GeoPackage file and the content within the GeoPackage file that is applicable to all GeoPackage files.

This Interoperability Standard is a Class 2 conformance profile as defined by ISO 19106, Geographic Information – Profiles (dated 2004-07-01), that includes a single standard with permitted DGIWG extensions and restrictions to the standard. An Interoperability Standard provides detailed direction on how to use the clauses, options, and parameters of the base standard(s). An OGC GeoPackage is an open, standards-based, platform-independent, portable, self-describing, compact format for transferring geospatial information. The DGIWG GeoPackage standard profile describes a set of conventions for storing the following within an SQLite database:

- Vector features
- Tile matrix sets of imagery and raster maps at various scales
- Gridded data coverages in tile matrix sets
- Extensions to core functionality

The GeoPackage standard provides a configurable container that can be used to support several military use cases. The use cases range from discrete data bundles of operational information to support disconnected users, storing and dissemination of national or theater-level data to support large processing sites and servers, and support to autonomous data collection and retrieval operations. For each of these use cases, the core GeoPackage vector and raster content can be enhanced by a number of extensions to facilitate interoperability through standardizing tables and indexing information to include schema, tile matrix sets, styles, metadata, gridded coverage data (e.g., elevation data), and related non-spatial information (e.g., photos, audio files, sketches). This DGIWG profile provides direction on common means to leverage basic OGC GeoPackage capabilities as well as implementing extensions while maintaining interoperability.

A GeoPackage is stored as an SQLite container. The OGC and DGIWG GeoPackage Encoding Standards govern the rules and requirements of the content stored in a GeoPackage container. The OGC standard defines the schema for a GeoPackage, including table definitions, integrity assertions, format limitations, and content constraints. The required and supported content of a GeoPackage is entirely defined in the OGC standard. The DGIWG GeoPackage profile is written to clarify and constrain the implementation of OGC standard used by the DGIWG member nations.

As GeoPackage is a database container, it supports direct use. This means that data in a GeoPackage can be accessed and updated in a “native” storage format without intermediate

format translations. GeoPackages that comply with the requirements in this DGIWG standard and do not implement vendor-specific extensions are more likely to be interoperable across all enterprise and personal computing environments. GeoPackages are particularly useful on mobile devices such as cell phones and tablets in communications environments where there is limited connectivity and bandwidth.

Software implementations that claim to be conformant with the DGIWG GeoPackage profile will support all mandatory operations, parameters, and elements of the base standard, as well as the specifically defined requirements in this document.

The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance to the underlying OGC GeoPackage specification are summarized in the OGC Compliance Testing Policies and Procedures on the OGC Compliance Testing website. For this DGIWG standard, the compliance test will use the same compliance test engine as OGC. DGIWG compliance tests are hosted on the OGC beta test site at https://cite.opengeospatial.org/te2/ .

2 Scope

This document defines specific DGIWG requirements, recommendations, and guidelines for implementations of the OGC GeoPackage Encoding Standard 1.4 [4] OGC 12-128r19. This interoperability standard defines DGIWG implementable provisions that profile the OGC GeoPackage standard to ensure that DGIWG GeoPackages, GeoPackage SQLite Extensions, and supporting utilities and services fulfill their intended purposes and are fit for use.

The DGIWG GeoPackage profile is intended to be used by government, commercial, open-source development, and Non-Governmental Organizations (NGOs) in the following ways:

- Data Providers are able to deliver DGIWG GeoPackage data files. These applications may be stand-alone application or embedded within larger applications or environments.
- Infrastructure Software Providers deliver software that implements the SQLite engine, GeoPackage SQLite Extensions, GeoPackage libraries, and/or stand-alone GeoPackage utilities or web services to encode/decode, validate, and manage GeoPackage files.
- Application Software Providers deliver GeoPackage “Apps” that provide users with the capabilities they require in general or mission-specific environments. They may employ software provided by Infrastructure Providers and bundle their Apps with GeoPackages from Data Providers.
- Defence Procurement Organizations that acquire DGIWG conformant GeoPackages, infrastructure, and applications.

System architects, software engineers, developers, and data modelers of the DGIWG member organizations and its component systems/applications seek to promote uniform geospatial data access, GeoPackage and GeoPackage SQLite Extension development and use. The goal is to increase interoperability, modularity, and agility of applications within coalition architectures to provide geospatial support. This Profile is an adjunct to the base OGC GeoPackage standard – not a stand-alone document.

Coalition operations and multinational cooperation require interoperable approaches to GeoPackage production and exchange, in order to make sharing of raster and vector data possible. As the OGC GeoPackage specification itself does not provide sufficient guidance to enable interoperability in a defence coalition environment, this DGIWG profile provides requirements and recommendations to improve the interoperable use of GeoPackage files in the following areas:

- Define common components to include Coordinate Reference Systems (CRS), metadata, and tilesets structure.
- Identify common GeoPackage capabilities for vector datasets and raster.
- Address use cases.
- Address optional use of specific extensions most appropriate to DGIWG applications.

3 Conformance

The GeoPackage profile is defined in terms of the Normative Requirements defined in section 6.1. These requirements compose the Conformance Classes in section 3.1. Associated with a requirement, each conformance class is the basis for the abstract test cases in Annex A, which are used to define tests that measure an implementation's conformance to the DGIWG profile of GeoPackage.

3.1 Conformance classes

This document establishes ten conformance classes in two sets:

Table 1: Conformance Classes

Class #	Class Name	Class Identifier	ATS
OGC GeoPackage			
1	geopackage	http://www.dgiwg.org/std/gpkg/1.1/conf/geopackage	ATS A.1
DGIWG GeoPackage Profile			
2	extensions	http://www.dgiwg.org/std/gpkg/1.0/conf/extensions	ATS A.2
3	crs	http://www.dgiwg.org/std/gpkg/1.0/conf/crs	ATS A.3
4	metadata	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata	ATS A.4
5	validity	http://www.dgiwg.org/std/gpkg/1.0/conf/validity	ATS A.5
6	tile	http://www.dgiwg.org/std/gpkg/1.0/conf/tile	ATS A.6
7	zoom	http://www.dgiwg.org/std/gpkg/1.0/conf/zoom	ATS A.7
8	bbox	http://www.dgiwg.org/std/gpkg/1.0/conf/bbox	ATS A.8
9	gridded-crs	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs	ATS A.9
10	gridded	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded	ATS A.10

The conformance classes and requirements for OGC GeoPackage are contained in the referenced OGC specification. The OGC GeoPackage class includes both conformance classes in the OGC specification: Core (base) and Optional (options).

A DGIWG profile of GeoPackage must conform to both the OGC and DGIWG conformance class listed in Table 1. Any optional requirements in a DGIWG compliant GeoPackage must conform to

the Options OGC conformance class. Additional options and restrictions of the OGC conformance classes are stated in the DGIWG GeoPackage profile conformance classes.

DGIWG GeoPackage (GPKG) Profile conformance classes select requirements for GeoPackage producers and consumers allowing distribution of geospatial data in a military environment. The requirements in the conformance classes are contained within the sections of this document. Annex A lists the conformance abstract tests referencing requirements which will be exercised on any GeoPackage file claiming to implement the DGIWG GeoPackage profile.,

Editor Note on Edition 1.1 If a GeoPackage contains Gridded Data, a GeoPackage conformant to Edition 1.0 must also be conformant to Edition 1.1. The conformance classes *gridded-crs* (ATS A.9) and *gridded* (ATS A.10) are unique to the GeoPackage Gridded Data extension. Some *gridded* requirements apply only to elevation and bathymetric gridded data. The two gridded conformance classes added in DGIWG-126 GeoPackage edition 1.1 have a conformance identifier containing **1.1**. The *geopackage* conformance class www.dgiwg.org/std/gpkg/**1.1**/conf/geopackage changed to **1.1** because OGC GeoPackage version 1.3.1 was updated to version 1.4 [4]. OGC GeoPackage version 1.4 relaxed the Extensions requirement (# 4) and revised the update triggers for the R-Tree Spatial Indexes extension F.3. The conformance classes for DGIWG-126 GeoPackage edition 1.0 are distinct from the gridded conformance classes in Edition 1.1. The changes in GeoPackage 1.4 allow DGIWG-126 edition 1.0 *geopackage* requirements to be conformant to Edition 1.1.

NOTE

"http://www.dgiwg.org/std/gpkg/1.0/conf/<class>" is an HTTP URI which works as an identifier for the DGIWG conformance class. It is not a URL, redirecting to a web page. 1.0 identifies the version of the DGIWG standard (not the version of GPKG it profiles).

4 Normative and Informative References

Normative (prescriptive) references in the DGIWG GeoPackage Interoperability Standard are identified in Table-2. Normative references include provisions that are directly referenced in the DGIWG GeoPackage profile.

Table 2: Normative References in DGIWG GeoPackage Interoperability Standard

#	Title	Reference	Version
[1]	DGIWG Metadata Foundation (DMF)	DGIWG - 114. Geospatial metadata profile for the military community, based on ISO 19115, ISO 19115-1 and ISO 19115-2.	Ed.2.0 12 July 2017
[2]	EPSG Geodetic Parameter Registry Version: 8.6.1	The EPSG Geodetic Parameter Dataset is a structured dataset of Coordinate Reference Systems and Coordinate Transformations, accessible through this online registry (www.epsg-registry.org) or, as a downloadable zip file, through OGP's EPSG home page at www.epsg.org. The geographic coverage of the data is worldwide, but it is stressed that the dataset does not and cannot record all possible geodetic parameters in use around the world. The EPSG Geodetic Parameter Dataset is maintained by the Geodesy Subcommittee of OGP's Geomatics Committee.	v8.6.1
[3]	OGC 06-103r4, OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture	Revised OGC Simple Features specification – common architecture.	v1.2.1, 28 May 2011
[4]	OGC 12-128r19, OGC GeoPackage Encoding Standard	This OGC Encoding Standard defines GeoPackages for exchange and GeoPackage SQLite Extensions for direct use of vector geospatial features and / or tile matrix sets of earth images and raster maps at various scales.	v1.4.0, 02-06-2024
[5]	OGC 17-066r2, OGC GeoPackage Extension for Tiled Gridded Coverage Data	The "GeoPackage Extension for Tiled Gridded Coverage Data" extension (previously titled Elevation Extension) defines how to encode and store tiled regular gridded data, such as a digital elevation model, in a GeoPackage.	v1.1, 02-May-2022

#	Title	Reference	Version
[6]	OGC 17-083r2,- OGC Two Dimensional Tile Matrix Set	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 18-005r5] Abstract Specification Topic 2: Spatial Referencing by Coordinates.	v1.0, 06 Oct 2019
[7]	OGC 18-000, OGC GeoPackage Related Tables Extension (RTE)	This document defines the Related Tables Extension (RTE) for the GeoPackage Encoding Standard. The RTE defines the rules and requirements for creating relationships in a GeoPackage data store between geospatial data tables and other tables that contain or reference related content such as attributes or media.	v1.0, 08 May 2019
[8]	OGC 12-063r5, WKT1 - Standard: Geographic information - Well-known text representation of coordinate reference systems	Provides an updated version of Well-known Text (WKT) representation of coordinate reference systems that follows the provisions of ISO 19111:2007 and ISO 19111-2:2009. It extends the earlier WKT to allow for the description of coordinate operations.	v1.0, 01 May 2015
[9]	OGC 18-010r7, WKT2 - Geographic information - Well-known text representation of coordinate reference systems	OpenGIS Standard provides an updated version of WKT representation of coordinate reference systems that follows the provisions of ISO 19111:2019. It extends the WKT in OGC document 12-063r5 [8] (ISO 19162). The WKT2 specification is also known as ISO 19162:2019.	v2.0.6, 13 Aug 2019
[10]	OpenGIS Project Document 01-009, OpenGIS Implementation Specification: Coordinate Transformation Services	The OpenGIS Coordinate Transformation Service Standard (CT) provides a standard way for software to specify and access coordinate transformation services for use on specified spatial data. This standard addresses a key requirement for overlaying views of geodata (“maps”) from diverse sources: the ability to perform coordinate transformation in such a way that all spatial data are defined relative to the same spatial reference system.	v1.0, 12 Jan 2001

#	Title	Reference	Version
[11]	OGC 21-057, OGC GeoPackage WKT for Coordinate Reference Systems Extension	Revision to the GeoPackage WKT for Coordinate Reference Systems Extension that previously was published as Annex F.10 of the GeoPackage Encoding Standard.	V1.1.0, 14 Oct 2022

The documents listed in Table-3 contain useful information to augment DGIWG understanding and application of the material in this interoperability standard in conjunction with the actual standard profiled.

Table 3: Informative References in DGIWG GeoPackage Interoperability Standard

Reference Identifier	Standards and Specifications Title
CJCSI 3900.01D, 14 May 2015	Position (Point and Area) Reference Procedures.
DES.ISM.XML, v13, 09 May 2014	XML Data Encoding Specification for Information Security Markings - This XML Data Encoding Specification for Information Security Markings (ISM.XML) defines detailed implementation guidance for using Extensible Markup Language (XML) to encode Information Security Markings (ISM) data. This Data Encoding Specification (DES) defines the XML elements and attributes, associated structures and relationships, mandatory and cardinality requirements, and permissible values for representing security marking concepts using XML.
DGIWG-250 v 1.2.1, 2 Oct 2020	Defence Gridded Elevation Data (DGED) Product Implementation Profile.
EPSG Guidance Note 373-07-1	Surveying and Positioning Guidance Note Number 7, part 1, clause 5.9 EPSG codes and names.
GeoPackage Community Extensions	Not official extensions to the GeoPackage Encoding Standard. These extensions may be developed as part of OGC Innovation Program activities or completely outside of OGC activities. Using a community extension (even if it is proprietary) may be preferable to developing your own extension. See www.geopackage.org/extensions.html.
IETF RFC 2119, S. Bradner, Harvard University, March 1997	Key Words for use in RFCs to Indicate Requirement Levels.
ISO/IEC 13249-3:2016, Ed. 5, 2016	Information technology — Database languages — SQL multimedia and application packages — Part 3: Spatial - The purpose of ISO/IEC 13249 is to define multimedia and application specific types and their associated routines using the user-defined features in ISO/IEC 9075.
ISO/TS 19101-1:2014	Geographic Information – Reference Model - Part 1: Fundamentals.
ISO/TS 19101-2:2018	Geographic Information – Reference Model - Part 2: Imagery.
ISO 19106:2004(E)	Geographic information – Profiles.
ISO 19111:2019	Geographic information – Referencing by coordinates.
ISO 19112:2019	Geographic information – Spatial referencing by geographic identifiers.
ISO 19162:2019	Geographic information – Well-known text representation of coordinate reference systems.

Reference Identifier	Standards and Specifications Title
NATO AGeoP-21 Ed A Ver 1, February 2016	Geodetic Datums, Projections, Grids and Grid References - Establish the U.S. Department of Defense World Geodetic System 1984 (WGS 84) as the standard geodetic system for geospatial information used by NATO. The document also defines how datums, coordinates, projections, grids and position reporting are to be applied to geospatial information and used by NATO
NGA.SIG.0012_2.0.0_UTMUPS v2.0, 25 March 2014	This document defines the UTM, UPS and MGRS systems of coordinates and their implementation.
NGA.SIG.0014_1.0_PROJRAS V1.0, 24 April 2015	Specifies the map projections that should be used when implementing OGC GeoPackage Encoding Standard, OGC 12-128r10.
NGA.SIG.0028_1.0.0_MAPPROJ v1.0, 13 Dec 2017	Map Projections for GEOINT Content, Products, and Applications
NGA.SIG.0033_1.0.1_ELEV 2019, 30 Septemeber 2019	Digital Elevation Content.
NGA.STND.0012_3.0, v3.0, 31 August 2016	The U.S. National System for Geospatial Intelligence (NSG) Metadata Foundation (NMF) defines the minimum mandatory geospatial metadata for datasets, series, and services in the NSG.
NGA.STND.0018_3.0, version 3.0, 20 January 2017	The U.S. National System for Geospatial Intelligence (NSG) Metadata Implementation Specification (NMIS) v3.0 specifies an Extensible Markup Language (XML) encoding of the NSG Metadata Foundation (NMF) v3.0 conformant to ISO 19115-3:2016.
NGA.STND.0019_2.0.1, 20 Aug 2020	NGA Standardization Document + Time-Space-Position Information (TSPI)
NGA.STND.0037_2.0.0_GRIDS, v2.0.0, 28 February 2014	Universal Grids and Grid Reference Systems - intended for use when developing systems that generate map grids and grid data for portrayal on products, and to ensure their correctness on such products; and for the development of doctrine for land navigation and location referencing. This standard is referenced in NATO documents and is necessary for interoperability with DoD international partners and allies.
NGA.STND.0079_1.0_3DGEO, v1.0, 04 January 2023	3D GEOINT Standard.
OGC 06-104r4, v1.2.1, 04 Aug 2010	OpenGIS Implementation Standard for Geographic information — Simple feature access — Part 2: SQL option - Revised OGC Simple Features specification – SQL encoding.

Reference Identifier	Standards and Specifications Title
OGC 10-100r3, v3.2.1	Geography Markup Language (GML) simple features profile (with technical note) - GML Simple Features Profile that specifies restricted subset of simple geometry types.
OGC 18-005r5, v5.0.01, 2021-07-02	OGC Abstract Specification Topic 2: Referencing by coordinates Corrigendum - defines the conceptual schema for the description of referencing by coordinates consistent with ISO 19111:2019.
OGC 13-082r2 v1.0, 19 Jan 2016	OGC Web Map Tile Service (WMTS) Simple Profile.
OGC 99-049, Rev.1.1, 5 May 1999	OpenGIS Simple Features Specification for SQL - Original OGC SF/SQL specification.

5 Terms, definitions, and abbreviations

5.1 Definitions

The purposes of this document, terms, and definitions found in the DGIWG GeoPackage Profile apply.

Table 4: Terms and Definitions

aspatial support	Support for data lacking spatial attributes (e.g., ground cover, temperature, …). In GeoPackage, the SQLite tables/views are without a geometry column, potentially associated with metadata.
community extension	GeoPackage extensions that are not officially approved extensions to the OGC GeoPackage Encoding Standard. These extensions may be developed as part of OGC Innovation Program activities or completely outside of OGC activities. Using a community extension (even if it is proprietary) may be preferable to developing your extension. Extensions that have widespread adoption may be considered for OGC approval.
compound coordinate reference system	A compound coordinate reference system is a non-repeating sequence of two or more coordinate reference systems none of which can itself be compound. (OGC 18-005r5). The compound A coordinate reference system for GeoPackage uses two other independent coordinate reference systems, one for the horizontal component and one for the vertical component, to describe a position (derived from INSPIRE Glossary).
conformal mapping	Conformal mapping, also called a conformal map, conformal transformation, angle-preserving transformation, or biholomorphic map, is a transformation that preserves local angles. An analytic function is conformal at any point where it has a nonzero derivative. Conversely, any conformal mapping of a complex variable that has continuous partial derivatives is analytic. Conformal mapping is extremely important in complex analysis, as well as in many areas of physics and engineering.
coordinate reference system	A coordinate system that is related to an object by a datum. [ISO 19111]
coordinate system	A set of mathematical rules for specifying how coordinates are to be assigned to points. [OGC 18-005r5]

coverage	A coverage is a function that describes characteristics of real-world phenomena that vary over space and/or time. Typical examples are temperature, elevation and precipitation. A coverage is typically represented as a data structure containing a set of such values, each associated with one of the elements in a spatial, temporal or spatiotemporal domain. Typical spatial domains are point sets (e.g. sensor locations), curve sets (e.g. contour lines), grids (e.g. orthoimages, elevation models), etc. A property whose value varies as a function of time may be represented as a temporal coverage or time-series. [OGC 17-066r2]
data validity constraints	Constraints on allowable data values in GeoPackage SQL tables to enable assessment and enforcement of data validity. Its requirements are conditional based on whether a GeoPackage contains features or tiles.
datum	A datum (in modern geodesy, a reference frame) specifies the relationship of a coordinate system to an object, thus ensuring that the abstract mathematical concept “coordinate system” can be applied to the practical problem of describing positions of features by means of coordinates. The object will generally, but not necessarily, be the Earth or a feature on the Earth such as a building. [OGC 18-005r5]
disadvantaged user	Persons using information technology in an environment that lacks access to or has a low level of network bandwidth and/or computing capacity.
gridded data	Data whose attribute values are associated with positions on a grid coordinate system [ISO:19115-2:2019].
informative metadata reference	A citation to any information resources used to compile the metadata and construct the metadata record. Each Metadata Source Reference takes the form of a citation.
Infrastructure Providers	The underlying operating system and common services that provide a platform to host the applications used to create and utilize the GeoPackage format.
interoperability	Capability to communicate, execute programs, or transfer data among various functional units in a manner that requires the user to have little or no knowledge of the unique characteristics of those units ISO 2382-1. "The ability for a system or components of a system to provide information portability and inter-application, cooperative process control. Interoperability, in the context of the OpenGIS Specification, is software components operating reciprocally (working with each other) to overcome tedious batch conversion tasks, import/export obstacles, and distributed resource access barriers imposed by heterogeneous processing environments and heterogeneous data.
map projection	A set of mathematical algorithms and associated parameters that establish a systematic, one-to-one correspondence between points on the surface of an ellipsoid and points on a plane while controlling the resulting geometric distortions. (NGA.SIG.0028)

navigation	A recommended use of geospatial data that indicates whether the map projection can be used for land, sea, and air navigation provided that its sources, content, scale, and usage adhere to the Navigation doctrine. [NGA.SIG.0028_1.0.0_MAPPROJ]
OGC compliance and interoperability testing and evaluation program	The OGC Conformance & Interoperability Testing & Evaluation Initiative (CITE) is an OGC Interoperability Initiative designed to test and evaluate OGC Interfaces and products that implement them. The CITE Initiative has three focus areas related to the establishment of a successful and robust OGC Conformance and Interoperability Test and Evaluation Program: Planning and Feasibility Study, Conformance Engine, Scripts and Guidelines, and CITE Portal and Reference Implementations.
OGC web services context	The OGC Web Services Context Document (OWS Context) was created to allow a set of configured information resources (service set) to be passed between applications primarily as a collection of services. OWS Context is developed to support in-line content as well. The goal is to support use cases such as the distribution of search results, the exchange of a set of resources such as OGC Web Feature Service (WFS), Web Map Service (WMS), Web Map Tile Service (WMTS), Web Coverage Service (WCS) and others in a 'common operating picture'.
raster map	Numeric representation of a cartographic map that may represent differing features of an area, such as a topographic map, navigation chart, or general reference product; these are either replicas of graphic products that are scanned, or a representation of an equivalent topographic vector product at an equivalent scale, according to a sampling resolution. [DGIWG 254]
raster tile pyramid	A tile matrix set with successive image layers of a pyramid at scales of uniform resolution such as reducing the pixel size between levels by powers of 2. [OGC 17-083r2 and r4 – 2D tile matrix set]
Related Tables	The Related Tables extension (OGC 18-000) defines the rules and requirements for creating relationships in a GeoPackage datastore between geospatial or attributes data tables and other tables that contain or reference related content such as attributes or media. Geospatial data tables (such as features or tiles tables) contain location information and/or geometries. Conceptually, this extension is similar to the OGC Table Joining Service Interface Standard. [OGC TJS-IS, 10-070r2]
relative positioning	A recommended use of geospatial data that indicates whether data (i.e. map projection) can be used for relative positioning, i.e. the determination of the coordinates of a point by applying range and bearing measurements relative to a point whose coordinates are known.

RTree spatial indexes	The RTree Spatial Indexes extension provides a means to encode an RTree index for geometry values in a GeoPackage. An RTree index provides a significant performance advantage for searches with basic envelope spatial criteria that return subsets of the rows in a feature table with a non-trivial number (thousands or more) of rows. [OGC GeoPackage 12-128r19]
semantic annotations	The process of attaching additional information to various concepts (e.g., people, things, places, organizations, etc.) in a given text or any other content. Unlike classic text annotations for the reader's reference, semantic annotations are used by machines to refer.
spatial reference system	A Spatial Reference System is a superset which includes a Coordinate Reference System (CRS) and spatial referencing by geographic identifier. But in ISO 19125-1 only CRS concepts are described.
spatiotemporal reference system	A Spatial Reference System with a temporal dimension, i.e. time.
SQLite	SQLite is a C-language library that implements a small, fast, self-contained, high-reliability, full-featured, SQL Relational database engine.
SQL triggers	In a SQL database, triggers are database objects, a special kind of stored procedure, which "reacts" to certain actions we make in the database. The main idea behind triggers is that they always perform an action in case some event happens.
tile matrix set	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.
vector	A representation of the spatial extent of geographic features using geometric elements (such as point, curve, and surface) in coordinate space.
Web map server	a standard protocol developed by the OGC in 1999 for serving georeferenced map images over the Internet. These images are typically produced by a map server from data provided by a GIS database.
Web services	Web services are self-contained, self-describing, modular applications that can be published, located, and invoked across the Web. Web services perform functions that can be anything from simple requests to complicated business processes. Once a Web service is deployed, other applications (and other Web services) can discover and invoke the deployed service.
WebP	An image format that uses either (i) the VP8 key frame encoding to compress image data in a lossy way, or (ii) the WebP lossless encoding (and possibly other encodings in the future). These encoding schemes should make it more efficient than currently used formats.

well-known text	Well-known text (WKT) offers a compact machine- and human-readable representation of geometric objects. WKT may also be used for succinctly describing the critical elements of coordinate reference system (CRS) definitions.

5.2 Abbreviations

Table 5: Abbreviations

3D	3 Dimensional
AGC	Army Geospatial Center
COP	Common Operational Picture
CRS	Coordinate Reference Systems
CT	Coordinate Transformation Service Standard
COMP	Compound CRS type
DES ISM XML	XML Data Encoding Specification for Information Security Markings
DGED	Defence Gridded Elevation Data
DGIWG	Defence Geospatial Information Working Group
DMF	DGIWG Metadata Foundation
DoD	Department of Defense (USA)
DSE	Data Services Environment
DTED	Digital Terrain Elevation Data
EGM	Earth Gravitational Model
EPSG	European Petroleum Survey Group
ETRS89	The European Terrestrial Reference System 1989
GDAL	Geospatial Data Abstraction Library
GEOD	Geodetic CRS type
GeoJSON	Geographic JavaScript Object Notation
GML	Geography Markup Language
Gpkg	GeoPackage
GPS	The Global Positioning System
IC	Intelligence Community
IEC	International Electrotechnical Commission
IETF RFC	Internet Engineering Task Force Requests for Comments
IR	Information Resources

ISO	International Organization for Standardization
LAEA	Lambert azimuthal equal-area projection
LCC	Lambert Conformal Conic
MGCP	Multi-national Geospatial Co-production Program
MGRS	The Military Grid Reference System
MIME	Multipurpose Internet Mail Extensions
MSL	Mean Sea Level
NAD83	The North American Datum of 1983
NATO	The North Atlantic Treaty Organization
NGA SIG	NGA Standardization Information Guidance
NGOs	Non-Governmental Organizations
NMF	NSG Metadata Foundation
NMIS	NSG Metadata Implementation Specification
NSG	The National System for Geospatial Intelligence (USA)
OGC	Open Geospatial Consortium
OGC CITE	OGC Compliance and Interoperability Testing and Evaluation Program
OGC SF	OGC Simple Features
OGL	Open Government Licence
OGP	Open Government Partnership
OWS	OGC Web Services
QGIS	Quantum Geographic Information System
PROJ	Projected CRS type
RTE	Related Tables Extension
SME	subject matter expert
SQL	Structured Query Language
SRS	Spatial Reference Systems
TC	Technical Committee
TMS	tiles matrix sets
TSPI	Time-Space-Position Information
UPS	Universal Polar Stereographic
URI	Uniform Resource Identifier
UTM	Universal Transverse Mercator
VERT	Vertical CRS type

WGS 84	World Geodetic System of 1984
WKSS	Well-Known Scale Sets
WKT	Well-Known Text
WKT1	Well-Known Text version 1 (OGC 12-063r5)
WKT2	Well-Known Text version 2 (OGC 18-010r7)
XML	Extensible Markup Language

6 DGIWG GeoPackage Profile

A GeoPackage that complies with the DGIWG GeoPackage profile will:

a) satisfy all requirements stipulated in the OGC GeoPackage Encoding Standard 1.4.

b) satisfy all requirements stipulated in this document.

This profile provides advice on the implementation of the GeoPackage so that tests can be provided to ensure objective compliance with the profile. The profile provides a “Normative Clause” to describe how each component must be implemented. The Normative Clause defines requirements where mandatory compliance is required for the attainment of conformance. However, the profile also includes optional Recommendations which may require additional tests not defined in this profile.

Note: All Requirements and Recommendations presented within this document are the result of information gathered from DGIWG Nations participating in the DGIWG Web Services Technical Panel, GeoPackage experiments and the profile development process. Implementers should be aware that certain requirements are unique to a national profile and recognize that organizations may need to utilize different configurations to meet specific operational or nationally directed requirements, especially concerning projections and metadata. DGIWG recommends that in these cases, general adherence with the guidance of this profile is maintained with only essential national modifications.

6.1 Normative Requirements

This chapter defines normative requirements to implement the DGIWG GeoPackage conformance class. Specific conformance classes are defined in section 3.0 and tested for compliance in Annex A.

The first two requirements ensure compliance to the GeoPackage Encoding Standard.

Requirement 1	/req/geopackage/base ***The DGIWG GeoPackage SHALL comply with all the OGC GeoPackage standard requirements in the Base conformance class according to the normative reference [4] GeoPackage Encoding Standard and the requirements defined therein in any GeoPackage that contains the extensions used in this GeoPackage.***

Requirement 2	/req/geopackage/options ***The DGIWG GeoPackage SHALL comply with all the OGC GeoPackage standard requirements in the Options conformance class according to the normative reference [4] GeoPackage Encoding Standard and the requirements defined therein.***

The Normative requirements requested by this conformance class are summarized in Table-6.

The following syntax is used to indicate the compliance requirement within the profile:

- Mandatory (M) – The requirement must be implemented
- Conditional (C) – Mandatory when “If” statement applies

For the DGIWG profile, a Mandatory requirement states a capability that must be present in GeoPackage data produced by DGIWG member countries and GeoPackages used by DGIWG members in a coalition environment. The Conditional requirements in the DGIWG GeoPackage profile apply if the implementation or use of GeoPackage meets a specific condition. Each requirement has a unique identifier listed in Table-6. The Identifier for each Requirement is abbreviated in this document. According to best practices in the DGIWG namespace structure and management, the full identifier for each requirement in the DGIWG GeoPackage profile version 1.0 is www.dgiwg.org/std/gpkg/1.0/req/<identifier> . Following OGC structured requirements, the requirements in this document are referenced by a hyperlink to the name or the Requirement Identifier. The requirement number is provided for convenience and may change in revisions to the profile.

Table 6: Normative Requirements

No.	Requirement Name	Identifier	Compliance
1	GeoPackage Base definition	/req/geopackage/base	M
2	GeoPackage Options definition	/req/geopackage/options	M
3	Mandatory Extensions	/req/extensions/mandatory	M
4	Optional Extensions	/req/extensions/optional	C
5	Extensions Not Allowed	/req/extensions/not-allowed	M
6	Conditional Extensions	/req/extensions/conditional	C
7	Raster CRS Allowed	/req/crs/raster-allowed	M
8	CRS Raster tile matrix set	/req/crs/raster-tile-matrix-set	M
9	Two-Dimensional Vector CRS	/req/crs/2d-vector	M
10	Three-Dimensional Vector CRS	/req/crs/3d-vector	M
11	Two-Dimensional Gridded CRS	/req/crs/gridded-crs-2d-allowed	M
12	Three-Dimensional Gridded CRS	/req/crs/gridded-crs-3d-allowed	M
13	Well Known Text for CRS	/req/crs/wkt	M
14	Compound CRS Usage	/req/crs/compound	C
15	Compound CRS Well Known Text	/req/crs/crs-compound-wkt	M
16	Gridded Compound CRS Well Known Text	/req/crs/gridded-crs-compound-wkt	C
17	Gridded CRS Epoch Well Known Text	/req/crs/gridded-crs-epoch-wkt	C
18	GeoPackage Metadata DMF	/req/metadata/dmf	M
19	GeoPackage Metadata document	/req/metadata/gpkg	M
20	Complete Row GeoPackage Metadata	/req/metadata/row	M
21	User Row GeoPackage Metadata	/req/metadata/user	C

No.	Requirement Name	Identifier	Compliance
22	GeoPackage Product Metadata	/req/metadata/product	C
23	GeoPackage Product Partial Metadata	/req/metadata/product-partial	C
24	GeoPackage Data Validity	/req/validity/data-validity	M
25	Tile Matrix Width Height	/req/tile/size-matrix	M
26	Tile Pyramid Data Width Height	/req/tile/size-data	M
27	Zoom level factor	/req/zoom/factor	M
28	Tile Matrix Set with Multiple Zoom Levels	/req/zoom/matrix-sets-multiple	C
29	Tile Matrix Set with one Zoom Level	/req/zoom/matrix-sets-one	C
30	Tile Matrix Set CRS Bounding box	/req/bbox/crs	M
31	Tile layer Metadata	/req/metadata/tile	C
32	Feature layer Metadata	/req/metadata/feature	C
33	Gridded Data Extension Core	/req/gridded/gridded-coverage-extension-core	C
34	Gridded Field Name Elevation	/req/gridded/field-name-elevation	C
35	Gridded Center Value Elevation	/req/gridded/center-value-elevation	C
36	Gridded Zoom Factor	/req/gridded/zoom-factor	C
37	Gridded User Row Metadata	/req/gridded/metadata-gridded-user	C

6.2 Recommendations

This chapter defines informative recommendations that guide the implementation of the DGIWG GeoPackage profile. Recommendations have an identifier but are not bound to conformance classes.

The Recommendations defined by this profile are summarized in Table-7.

Table 7: Recommendations

No.	Recommendation	Identifier
1	WGS84 Geographic Compound CRS Identifier	/recco/gridded-cmpnd-crs-wgs84
2	UTM Projected Compound CRS Identifier	/recco/gridded-cmpnd-crs-utm
3	UPS Projected Compound CRS Identifier	/recco/gridded-cmpnd-crs-ups
4	GeoPackage Product Metadata	/recco/metadata-product
5	Other Gridded Field Names	/recco/gridded-field-name-other

6.3 Extensions

GeoPackage Extensions add or modify the SQL tables within a GeoPackage. Some extensions are not defined in the GeoPackage core specification, but in documents separate from the OGC GeoPackage Encoding Standard. All extensions are registered in the gpkg_extension table.

The extensions from OGC GeoPackage 1.3, other OGC published extensions, and community extensions are considered for the DGIWG Profile. Table-8 indicates those approved OGC GeoPackage extension that are mandatory/required (M), optional/recommended (O), conditional (C), not allowed (N) or not applicable (NA) to the DGIWG Profile. In the OGC document, details on the use of the extensions are addressed in section 2.3 for tiles and features. The deprecated extensions are not listed.

Table 8: Normative Extensions

Extension	Name	Source	DGIWG Profile: O – Optional, M – Mandatory, N - Not Allowed, C - Conditional, NA - Not Applicable		
			Features	Tiles	Gridded
F.1 GeoPackage Nonlinear Geometry Types	gpkg_geom_	[4] OGC GPKG 1.4	N	NA	NA
F.3 RTree Spatial Indexes	gpkg_rtree_index	[4] OGC GPKG 1.4	M	NA	NA
F.6 Zoom other Intervals	gpkg_zoom_other	[4] OGC GPKG 1.4	NA	N	NA
F.7 File Encoding WebP	gpkg_webp	[4] OGC GPKG 1.4	NA	N	NA
F.8 Metadata	gpkg_metadata	[4] OGC GPKG 1.4	M	M	M
F.9 Schema	gpkg_schema	[4] OGC GPKG 1.4	O	O	O
F.10 WKT for CRS (1.0)	gpkg_crs_wkt	[4] OGC GPKG 1.4	M	M	N
WKT for CRS 1.1 (replace F.10)	gpkg_crs_wkt_1_1	[11] OGC 21-057	O	O	M

Extension	Name	Source	DGIWG Profile: O – Optional, M – Mandatory, N - Not Allowed, C - Conditional, NA - Not Applicable		
			Features	Tiles	Gridded
F.11 Tiles Gridded Coverage	gpkg_2d_gridded_ coverage	[5] OGC 17-066r2	NA	NA	M
F.12 Related Tables	related_tables	[7] OGC 18-000	O	O	O

Extensions F.2, F.4, and F.5 have been removed from the OGC GeoPackage specification and are not listed in Table-8. Conformance of GeoPackage extensions is enforced by the following four requirements associated with the compliance of GeoPackages containing features and tiles as listed in Table-8.

Requirement 3	/req/extensions/mandatory ***GeoPackages containing Feature and/or Tile data SHALL be implemented to contain ALL of the Mandatory (M) GeoPackage extensions in Table-8.***

Requirement 4	/req/extension/optional ***GeoPackages containing Feature and/or Tile data SHALL be implemented to use any number of the Optional (O) GeoPackage extensions in Table-8 applicable to the data type Features and/or Tiles.***

Requirement 5	/req/extensions/not-allowed ***GeoPackages containing Feature and/or Tile data SHALL be implemented to contain NONE of the Not Allowed (N) GeoPackage extensions in Table-8.***

Requirement 6	/req/extensions/conditional ***GeoPackages containing Feature and/or Tile data SHALL be implemented to use any of the Conditional (C) GeoPackage extensions in Table-8 where the conditional criteria of the extension is true.***

When an extension is Not Allowed (“N”) for feature data or tile matrix sets in Table-8 of this profile and DGIWG in the future has a need to use that extension, the DGIWG community has an option to either modify this profile to allow the extension or to create a DGIWG profile of the extension to specify the community use of a specific GeoPackage extension. Once DGIWG approves a distinct profile of a GeoPackage extension, compliance to the extension profile’s requirements takes precedence over compliance to requirements for the same extension as specified in this DGIWG profile, DGIWG 126.

This profile accommodates additional extensions, whether an OGC extension or developmental extension that lacks a conformance requirement. Community extensions are not normative until they have been approved by OGC. Future consideration may promote a Community extension to become an approved extension that is incorporated into the OGC GeoPackage specification or a separate specification. Implementations of non-conformant extensions should recognize that changes or revisions are likely as the extension matures in the standards development process. The GeoPackage Community Extensions are published at https://www.geopackage.org/extensions.html .

OGC Standard extensions and Community extensions not contained in Table-8 are allowable in a GeoPackage implementation of the DGIWG profile if an extension does not conflict with a Mandatory extension. Only the extensions listed in Table-8 are enforced with compliance per conformance classes for extensions in ATS A.3.

A modification to this specification is necessary for an extension that adds to or supersedes the extensions in Table-8 when the extension is considered a Mandatory, Optional, or Not Allowed extension. Where two versions of the extension exist, a preference for a more recent version is given if it supersedes the previous extension.

6.4 File Names

When the DGIWG profile is implemented to define a minimum set of metadata, the users and clients can query the metadata in the GeoPackage for more information about the contents. The metadata required in sections 7.3, 8.4, and 9.1 describes the contents of the GeoPackage and its layers with more fidelity than contained in the file name. If a metadata query is not suitable for the user's search for GeoPackage files, the GeoPackage file name can provide useful elements about the contents of the GeoPackage.

The GeoPackage producer may follow the guidance in this section to name GeoPackage files. In this case, GeoPackage file names should include the following elements:

- GeoPackage Producer
- Data Product
- Geographic Coverage Area
- Zoom Levels
- Version
- GeoPackage Creation Date

Additional elements may be added to the file names for mission or agency-specific use. Elements should be ordered from general to specific, with "_" used to delimit elements and "-" used to delimit words within an element. The following general pattern should be used:

{Producer}_{Data Product}_{Geo Coverage Area}_{Zoom Levels}_{Version}_{Date}

The examples below provide further guidance on file names for DGIWG GeoPackages:

Table 9: DGIWG GeoPackage example

GeoPackage Example	**"AGC_BUCK_Ft-Bliss_14-20_v1-0_29AUG2016.gpkg"**	**"OGL_BOUND_UK_10K_v1-0_09APR2020.gpkg"**
GeoPackage Producer	*Army Geospatial Center (AGC)*	*Open Government Licence (OGL)*
Data Product(s)	*Buckeye*	*Boundary-Line*
Geographic Coverage Area	*Fort Bliss, Texas, USA*	*UK*
Zoom Levels	*14-20*	*-*
Scale Levels	*-*	*1:10,000*
Version	*1.0*	*1.0*
Creation Date	*29 August 2016*	*09 April 2020*

The differences between both GeoPackage examples are mainly because they are both from

different countries. Also, some use cases require higher zoom levels to create a gpkg.

GeoPackage versions should be updated in a consistent manner. Minor version updates (“v1-1”) may be used to indicate that new features and/or tiles have been added to the existing raster tile pyramid or feature data set. Major version updates (“v2-0”) may be used to indicate that the tiles and/or features are substantially different than in the previous version.

7 Tiles, Features, and Gridded Data

7.1 Coordinate Reference Systems

This clause specifies allowable CRS and their WKT definitions that will be included in the DGIWG GeoPackages. In this document, CRS is the preferred and more accurate terminology. The term Spatial Reference Systems (SRS) is not used in this document except in pre-existing field names. These CRS are used in the gpkg_spatial_ref_sys table and referenced by the srs_id column values in gpkg_contents, gpkg_tile_matrix_set and gpkg_geometry_columns table.

The DGIWG GeoPackage profile uses the 2-dimensional (2D) tiles matrix sets (TMS) defined in Annex D of the "OGC Two Dimensional Tile Matrix Set" ([6] OGC 17-083r2) as required in section 8.2. The CRS defined and named in the tables below correspond to the eight tile matrix sets defined in Annex D of [6]. An additional projection for Lambert Conformal Conic (LCC) is listed for NATO member country interoperability.

The CRSs that are allowed for 2D DGIWG GeoPackage data are listed in Table-11 and Table-13. DGIWG GeoPackages use raster 3-dimensional (3D) matrix sets as described in Table-12 below. The term *Recommended Use* in the DGIWG profile of GeoPackage applies to GeoPackage data produced and applications consuming GeoPackage with the CRS. Geospatial clients are expected to support all the GeoPackage CRS for Relative Positioning and Navigation Recommended Use. Geospatial clients are not required to use GeoPackage data in a CRS other than those required in this section but are expected to still be operable or gracefully reject GeoPackage data in an unsupported CRS. The CRS listed in Table-10 states the most restrictive Recommended Use of the CRS for each the tile matrix sets defined by the 2D Tile Matrix Sets [6] (D.1 to D.8) and the LCC projection required by NATO AGeoP-21. The use of "country specific usage" only does not require a DGIWG implementation.

Table 10: Tile Matrix Set CRSs for DGIWG GeoPackage Profile

2D Tile Matrix OGC 17-083r2 section	DGIWG Recommended Use	Tile matrix set name	CRS - EPSG
D.1	Visualization and Situational Awareness	Web Mercator Quad TileMatrixSet	EPSG:3857
D.2	Relative positioning & Navigation	World CRS84 Quad TileMatrixSet	WGS84 with coordinate order per Annex E [1]
D.3	Relative positioning & Navigation	World Mercator WGS84 Quad TileMatrixSet	EPSG:3395

2D Tile Matrix OGC 17-083r2 section	DGIWG Recommended Use	Tile matrix set name	CRS - EPSG
D.4	Relative positioning, country-specific usage	Universal Transverse Mercator WGS84 Quad family TileMatrixSet	EPSG:32601 - 32660, 32701 - 32760
D.5	Relative positioning & Navigation	Arctic Universal Polar Stereographic WGS 84 Quad TileMatrixSet	EPSG:5041
D.6	Relative positioning & Navigation	Antarctic Universal Polar Stereographic WGS84 Quad TileMatrixSet	EPSG:5042
D.7	Not required, country-specific usage	European ETRS89 LAEA Quad TileMatrixSet	EPSG:3035
D.8	Not required, country-specific usage	Canadian NAD83 LCC TileMatrixSet	EPSG:3978 (NAD83)
N/A	Relative positioning & Navigation	Lambert Conformal Conic - per product TileMatrixSet	projections using EPSG:9801 or 9802 coordinate operation

Note D.1 EPSG:3857, Web Mercator, is a common standard used for web mapping applications. It is used by virtually all major online map providers, including Google Maps, Bing Maps, OpenStreetMap, Mapquest, Esri, Mapbox, and others. **If using WMTS map data from a Volunteered Geographic Information (VGI) or commodity data source which is in the Web Mercator projection, it is highly recommended that your service warns users that this data is suitable for Visualization use cases only. For defence use cases and operations that require precise locations and precise navigation (land, air, and sea), one of the 5 CRS with permitted use for relative positioning and navigation, such as the World Mercator projection EPSG:3395, is mandatory.**

Table-11 lists the CRSs that are recommended for raster tile pyramid DGIWG GeoPackages viewed in 2D Map software applications, and Table-12 lists the CRSs that are recommended for raster tile pyramid DGIWG GeoPackages viewed in 3D Globe software applications. Table-13 lists the CRSs that are used for vector feature DGIWG GeoPackages, including both 2D and 3D vector features.

The following Requirement /req/crs/raster-allowed does not apply to "Tiled Gridded Coverage Data" as described in "OGC 17-066r2, OGC GeoPackage Extension for Tiled Gridded Coverage Data" [5]. This type of data has to be referenced in the "gpkg_contents" table with the value

"2d_gridded_coverage" for the "data_type" column. If necessary, other CRS requirements can be defined for this type of data in a specific DGIWG profile.

CRS parameter values adhere to be those stated in Table-15 through Table-26 of Section 7.2 (WKT). Parameter values for CRS defined by EPSG are those specified by the EPSG Geodetic Parameter Registry when this document was published.

Applications that can read standard GeoPackages can read the specified CRS definitions in the gpkg_spatial_ref_sys table.

Requirement 7	/req/crs/raster-allowed ***The CRSs listed in Table-11 and Table-12 SHALL be the only CRSs used by raster tile pyramid tables in a GeoPackage.***

Note: The integer codes assigned to DGIWG CRSs in this document meet two conditions to avoid confusion with CRS codes used for data imported from external sources and converted to a CRS defined by DGIWG. First, they are unique among all srs_id values referenced by the base GeoPackage specification and those defined in this document. Second, they do not duplicate the codes for any CRSs registered in https://epsg.org, although some of them duplicate codes for coordinate operations and other items in that registry, where codes are unique within a type. It is recommended that these conditions continue to be met if these codes are changed by the appropriate DGIWG authority.

The CRS definitions in Table-15 through Table-26 in section 7.2 are used to specify the CRS for the raster tiles and vector feature user data tables data in a DGIWG GeoPackage.

Note: Annex C provides a link to the Coordinate Reference Systems (CRS) defined in these requirements and the corresponding identifiers in the EPSG.

7.1.1 Raster Data CRS

Table 11: Raster and Gridded Tile Pyramid Projected Coordinate Reference Systems recommended for use in 2D Map Applications

CRS Name	CRS AUTH ID	CRS Type	CRS Dimension	CRS Def
WGS 84 / World Mercator	EPSG:3395	PROJ	2	Table 17
WGS 84 / UPS North (E,N)	EPSG:5041	PROJ	2	Table 18

CRS Name	CRS AUTH ID	CRS Type	CRS Dimension	CRS Def
WGS 84 / UPS South (E,N)	EPSG:5042	PROJ	2	Table 19
WGS 84 / Web Pseudo-Mercator	EPSG:3857	PROJ	2	Table 20
WGS 84 / UTM zone 01-60N	EPSG:32601-32660	PROJ	2	Table 21
WGS 84 / UTM zone 01-60S	EPSG:32701-32760	PROJ	2	Table 21
WGS 84 / Lambert Conformal Conic using 1SP	per product	PROJ	2	Table 22
WGS 84 / Lambert Conformal Conic using 2SP	per product	PROJ	2	Table 23

Table 12: Raster and Gridded Tile Pyramid Coordinate Reference Systems recommended for use in 2D and 3D Globe Applications

CRS Name	CRS AUTH ID	CRS Type	CRS Dimension	CRS Def
WGS 84 Geographic 2D	EPSG:4326	GEOD	2	Table 16
WGS 84 Geographic 3D	EPSG:4979	GEOD	3	Table 15
WGS 84 / World Mercator	EPSG:3395	PROJ	2	Table 17
WGS 84 / UPS North (E,N)	EPSG:5041	PROJ	2	Table 18
WGS 84 / UPS South (E,N)	EPSG:5042	PROJ	2	Table 19

Requirement 8	/req/crs/raster-tile-matrix-set ***WGS 84 Geographic 2D (EPSG:4326), World Mercator (EPSG:3395), UPS North/South (EPSG:5041/5042), and UTM (EPSG:362--, EPSG:327--) raster tile pyramid GeoPackages SHALL follow the technical requirements provided in OGC 17-083r2 (OGC Two Dimensional Tile Matrix Set Standard) for implementation of Tile Matrix Sets.***

7.1.2 Vector Data CRS

Table 13: Vector feature Coordinate Reference Systems

CRS Name	CRS AUTH ID	CRS Type	CRS Dim	CRS Def
WGS 84 Geographic 2D	EPSG:4326	GEOD	2	Table 16
WGS 84 Geographic 3D	EPSG:4979	GEOD	3	Table 15
WGS84 4326 + EGM2008 height 3855	EPSG:9518	COMP	3	Table 25, Table 26
EGM2008 geoid height	EPSG:3855	VERT	1	Table 24

Requirement 9	/req/crs/2d-vector ***The 2D CRS in Table-13 SHALL be the only CRS used for 2D vector features in GeoPackage.***

Requirement 10	/req/crs/3d-vector ***The 3D CRS or compound CRS in Table-13 SHALL be the only CRS used for 3D vector features in GeoPackage.***

7.1.3 Gridded Data CRS

While the GeoPackage extension data type is "2d_gridded_coverage" or "gpkg_2d_gridded_coverage", GeoPackage is capable of containing both two dimensional (2D) and three dimensional (3D) gridded tiles. A 3D tile contains a grid of values in three-dimensional space that represents the height (elevation), depth (bathymetry), or composition of a 3D point on a grid. A 2D tile is a "flat" grid containing a measurement or a representation of the properties of grid points in the 2D space. 2D gridded data contains values for topographic (soil, moisture), cultural (population density, ethnicity, median income), atmospheric (temperature, humidity, barometric pressure) or hydrographic (salinity, temperature, water velocity) phenomena. For 2D and 3D gridded data, the distinct CRS allowed for each are specified in two requirements.

Requirement /req/crs/gridded-crs-2d-allowed is for 2D gridded data whose values are NOT related to a vertical CRS (including hill shade, color relief, and slopes). The 2D CRS cited in requirement gridded-crs-2d-allowed are appropriate for encoding tiled gridded data in GeoPackage when the scale and coverage area of the data are consistent with intended use of the CRS.

Requirement 11	/req/crs/gridded-crs-2d-allowed ***The CRSs listed in Table-11 and 2D CRS in Table-12 SHALL be the only 2D CRS used by gridded data tables in a GeoPackage.***

Requirement gridded-crs-3d-allowed is for 3D gridded data whose values ARE related to a vertical CRS, including elevation (height) or bathymetry (depth). A GeoPackage with gridded data includes a column in the gpkg_2d_gridded_coverage_ancillary table to indicate the type of gridded data.

Requirement 12	/req/crs/gridded-crs-3d-allowed ***The CRSs listed in Table-14 SHALL be the only 3D CRS used by gridded data tables in a GeoPackage.***

For 3D gridded data, three Compound CRS listed in Table 14 have a horizontal CRS, either geographic (WGS84) or projected (UTM, UPS), and each horitizal CRS is paired with a vertical CRS. The WGS 84 realizations are listed as an Ensemble in the WKT2 definition of CRS based upon WGS 84, such as EPSG:4326. Those realizations defined are listed in the informative reference NGA.STND.0079_1.0_3DGEO Annex E footnote [5]. A valid vertical CRS is an authoritative, publicly defined geodetic vertical datum, such as the Earth Gravitational Model (EGM). Examples of vertical CRS defined in [2] EPSG are EGM 2008 (EPSG:3855) and EGM96 (EPSG:5773). The vertical

CRS used in a gridded data product is determined by a product specification for the gridded data (i.e. DGIWG-250, DGED) and other standards adopted by the producer, such as NATO AGeoP-21, *Geodetic Datums, Projections, Grids and Grid Reference* (an informative reference in section 4). The Compound CRS well known text format is defined in the tables indicated in section 7.2.

Table 14: 3D Gridded Data Coordinate Reference Systems

CRS Name	CRS Type	CRS 3D ID	CRS Horizontal ID	CRS Vertical ID	Example CRS Def
WGS 84 Geographic 3D	Geographic 3D	EPSG:4979 or 3D realization [5]	N/A	N/A	Table 15
WGS84 4326 + EGM2008 height	Geographic Compound	EPSG:9518	EPSG:4326	EPSG:3855	Table 26
WGS84 4326 + Vertical	Geographic Compound	unique ID	EPSG:4326 or 2D realization [5]	A valid vertical CRS	Table 27
UTM + Vertical	Projected UTM Compound	unique ID	EPSG:32601-32660, 32701-32760	A valid vertical CRS	Table 28
UPS + Vertical	Projected UPS Compound	unique ID	EPSG:5041, 5042	A valid vertical CRS	Table 29

7.2 Well Known Text

This clause specifies allowable WKT definitions that are included in the DGIWG GeoPackages. CRS parameter values correlate to those in Table-15 through Table-26 below. Parameter values for CRS defined by EPSG are those specified by the EPSG Geodetic Parameter Registry when this document was published.

Requirement 13	/req/crs/wkt ***The CRS definitions in Table-15 through Table-24 in section 7.2 SHALL be used to specify the CRS used for tiles and vector features in a DGIWG compliant GeoPackage.***

WKT1 [8] specifies a WKT encoding for a 3D geodetic CRS, and a way to correctly specify the axis directions for polar stereographic projections, which are required for this profile. The WKT values in the gpkg_spatial_ref_sys columns are expressed in the definition column in the gpkg_spatial_ref_sys table, which is based on WKT in OGC 01-009. This DGIWG profile requires GeoPackage extension F.10 gpkg_crs_wkt to be used with WKT1 [8] expressed in column definition_12_063 for all supported CRS. Extension F.10 / Requirement 117 in GeoPackage 1.3.1, GeoPackage 1.4 [4], and WKT extension 1.1.0 [11] states that column definition_12_063 takes precedence if both definition values are defined.

The OGC GeoPackage extension *WKT for Coordinate Reference Systems Extension* (21-057) version 1.1.0 [11] allows WKT in the definition_063 column to conform to WKT1 (OGC 12-063r5 [8]) and WKT2 (OGC 18-010r7 [9]). The WKT 1.1.0 extension is required for the 2d_gridded_coverage extension in order to use WKT2 for 3D Compound CRS and the optional Epoch date. DGIWG-126 Edition 1.0 GeoPackage profile does not require OGC 21-057 or WKT2, but implementations have the option of using WKT2 if WKT extension 1.1.0 is included in the GeoPackage.

For each CRS in a GeoPackage, the AXIS definition includes the ORDER, LENGTHUNIT, or ANGLEUNIT terms, with the exception of Well-Known Binary (WKB) encoding of geometry as stated in [4] GeoPackage section 2.1.3.

7.2.1 Geographic WKT

Table 15: WGS 84 Geographic 3D CRS Definition Annex E footnote [2]

gpkg_spatial_ref_sys Column Name : value
srs_name : http://www.opengis.net/def/crs/EPSG/0/4979
srs_id : 4979 Annex E footnote [3]
organization : EPSG

gpkg_spatial_ref_sys Column Name : value
organization_coordsys_id : 4979
description : Used by the GPS satellite navigation system and for NATO military geodetic surveying.
definition :

```
GEODCRS["WGS 84",
    DATUM["World Geodetic System 1984",
        ELLIPSOID["WGS 84",6378137,298.257223563,
            LENGTHUNIT["metre",1.0]]],
    CS[ellipsoidal,3],
    AXIS["Geodetic latitude (Lat)",north,
        ANGLEUNIT["degree",0.0174532925199433]],
    AXIS["Geodetic longitude (Long)",east,
        ANGLEUNIT["degree",0.0174532925199433]],
    AXIS["Ellipsoidal height (h)",up,
        LENGTHUNIT["metre",1.0]],
    ID["EPSG",4979]]
```

OGC ISO 19125-1:2004 and OGC 01-009 specify a WKT encoding for the WGS 84 Geographic 2D CRS which ensures interoperability with existing GeoPackage implementations.

Table 16: WGS 84 Geographic 2D CRS Definition Annex E footnote [4]

gpkg_spatial_ref_sys Column Name : value
srs_name : http://www.opengis.net/def/crs/EPSG/0/4326 or http://www.opengis.net/def/crs/OGC/1.3/CRS84
srs_id : 4326
organization : EPSG
organization_coordsys_id : 4326
description : Horizontal component of 3D system. Used by the GPS satellite navigation system and for NATO military geodetic surveying.
definition :

```
WKT1:
GEOCCS["WGS 84",
    DATUM["WGS_1984",
        SPHEROID["WGS84",6378137,298.257223563]],
    PRIMEM["Greenwich",0],
    UNIT["degree",0.0174532925199433]]

WKT2:
GEOGCRS["WGS 84",
    ENSEMBLE["World Geodetic System 1984 ensemble",
        MEMBER["World Geodetic System 1984 (Transit)"],
        MEMBER["World Geodetic System 1984 (G730)"],
        MEMBER["World Geodetic System 1984 (G873)"],
        MEMBER["World Geodetic System 1984 (G1150)"],
        MEMBER["World Geodetic System 1984 (G1674)"],
        MEMBER["World Geodetic System 1984 (G1762)"],
        MEMBER["World Geodetic System 1984 (G2139)"],
        ELLIPSOID["WGS 84",6378137,298.257223563,
            LENGTHUNIT["metre",1]],
        ENSEMBLEACCURACY[2.0]],
    PRIMEM["Greenwich",0,
        ANGLEUNIT["degree",0.0174532925199433]],
    CS[ellipsoidal,2],
        AXIS["geodetic latitude (Lat)",north,
            ORDER[1],
            ANGLEUNIT["degree",0.0174532925199433]],
        AXIS["geodetic longitude (Lon)",east,
            ORDER[2],
            ANGLEUNIT["degree",0.0174532925199433]],
    USAGE[
        SCOPE["Horizontal component of 3D system."],
        AREA["World."],
        BBOX[-90,-180,90,180]],
    ID["EPSG",4326]]
```

7.2.2 Projected WKT

Table 17: WGS 84 / World Mercator Projected CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : http://www.opengis.net/def/crs/EPSG/0/3395
srs_id : 3395

gpkg_spatial_ref_sys Column Name : value
organization : EPSG
organization_coordsys_id : 3395
description : Mercator view of world excluding polar areas for very small scale mapping
definition :

```
PROJCRS["WGS 84 / World Mercator",
   BASEGEODCRS["WGS 84",
       DATUM["World Geodetic System 1984",
           ELLIPSOID["WGS 84",6378137,298.257223563]]],
   CONVERSION["Mercator",
       METHOD["Mercator (variant A)",
           ID["EPSG","9804"]],
       PARAMETER["Latitude of natural origin",0,
           ANGLEUNIT["degree",0.0174532925199433]],
       PARAMETER["Longitude of natural origin",0,
           ANGLEUNIT["degree",0.0174532925199433]],
       PARAMETER["Scale factor at natural origin",1,
           SCALEUNIT["unity",1.0]],
       PARAMETER["False easting",0,
           LENGTHUNIT["metre",1.0]],
       PARAMETER["False northing",0,
           LENGTHUNIT["metre",1.0]],
       ID["EPSG","19833"]],
   CS[Cartesian,2],
   AXIS["Easting (E)",east,
       ORDER[1]],
   AXIS["Northing (N)",north,
       ORDER[2]],
   LENGTHUNIT["metre",1.0],
   ID["EPSG","3395"]]
```

Table 18: WGS 84 / UPS North (E,N) Projected CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : http://www.opengis.net/def/crs/EPSG/0/5041
srs_id : 5041
organization : EPSG
organization_coordsys_id : 5041
description : Military mapping by NATO north of 60° N

gpkg_spatial_ref_sys Column Name : value
definition :

```
PROJCRS["WGS 84 / UPS North (E,N)",
    BASEGEODCRS["WGS 84",
        DATUM["World Geodetic System 1984",
            ELLIPSOID["WGS 84",6378137,298.257223563,
                LENGTHUNIT["metre",1.0]]]],
    CONVERSION["Universal Polar Stereographic North",
        METHOD["Polar Stereographic (variant A)",
            ID["EPSG","9810"]],
        PARAMETER["Latitude of natural origin",90,
            ANGLEUNIT["degree",0.0174532925199433]],
        PARAMETER["Longitude of natural origin",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        PARAMETER["Scale factor at natural origin",0.994,
            SCALEUNIT["unity",1.0]],
        PARAMETER["False easting",2000000,
            LENGTHUNIT["metre",1.0]],
        PARAMETER["False northing",2000000,
            LENGTHUNIT["metre",1.0]],
        ID["EPSG","16061"]],
    CS[Cartesian,2],
    AXIS["Easting (E)",south,
        MERIDIAN[90,
            ANGLEUNIT["degree",0.0174532925199433]],
        ORDER[1]],
    AXIS["Northing (N)",south,
        MERIDIAN[180,
            ANGLEUNIT["degree",0.0174532925199433]],
        ORDER[2]],
    LENGTHUNIT["metre",1.0],
    ID["EPSG","5041"]]
```

Table 19: WGS 84 / UPS South (E,N) Projected CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : http://www.opengis.net/def/crs/EPSG/0/5042
srs_id : 5042
organization : EPSG
organization_coordsys_id : 5042

gpkg_spatial_ref_sys Column Name : value
description : Military mapping by NATO south of 60° S
definition :

```
PROJCRS["WGS 84 / UPS South (E,N)",
    BASEGEODCRS["WGS 84",
        DATUM["World Geodetic System 1984",
            ELLIPSOID["WGS 84",6378137,298.257223563,
                LENGTHUNIT["metre",1.0]]]],
    CONVERSION["Universal Polar Stereographic South",
        METHOD["Polar Stereographic (variant A)",
            ID["EPSG","9810"]],
        PARAMETER["Latitude of natural origin",-90,
            ANGLEUNIT["degree",0.0174532925199433]],
        PARAMETER["Longitude of natural origin",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        PARAMETER["Scale factor at natural origin",0.994,
            SCALEUNIT["unity",1.0]],
        PARAMETER["False easting",2000000,
            LENGTHUNIT["metre",1.0]],
        PARAMETER["False northing",2000000,
            LENGTHUNIT["metre",1.0]],
        ID["EPSG","16161"]],
    CS[Cartesian,2],
    AXIS["Easting (E)",north,
        MERIDIAN[90,
            ANGLEUNIT["degree",0.0174532925199433]],
        ORDER[1]],
    AXIS["Northing (N)",north,
        MERIDIAN[0,
            ANGLEUNIT["degree",0.0174532925199433]],
        ORDER[2]],
    LENGTHUNIT["metre",1.0],
    ID["EPSG","5042"]]
```

Table 20: Web Mercator Quad Tile Matrix Set

gpkg_spatial_ref_sys Column Name : value
srs_name : http://www.opengis.net/def/crs/EPSG/0/3857
srs_id : 3857
organization : EPSG

gpkg_spatial_ref_sys Column Name : value
organization_coordsys_id : 3857
description : Uses spherical development of ellipsoidal coordinates. This should only be used for visualization purposes.
definition :

```
PROJCS["WGS 84 / Pseudo-Mercator",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
            SPHEROID["WGS 84",6378137,298.257223563,
                ID["EPSG","7030"]],
            ID["EPSG","6326"]],
        PRIMEM["Greenwich",0,
            ID["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            ID["EPSG","9122"]],
        ID["EPSG","4326"]],
    PROJECTION["Mercator_1SP"],
    PARAMETER["central_meridian",0],
    PARAMETER["scale_factor",1],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0],
    UNIT["metre",1,
        ID["EPSG","9001"]],
    AXIS["X",EAST],
    AXIS["Y",NORTH],
    ID["EPSG","3857"]]
```

Table 21: Universal Transverse Mercator WGS84 Quad Family TileMatrixSet

gpkg_spatial_ref_sys Column Name : value
srs_name : [http://www.opengis.net/def/crs/EPSG/0/32601 - http://www.opengis.net/def/crs/EPSG/0/32660]
srs_id : 32601 - 32660, 32701-32760
organization : EPSG
organization_coordsys_id : 32601 - 32660, 32701-32760
description : Each UTM Zone is a universal grid with an instance of the transverse mercator projection, onshore and offshore. Extent details - each UTM zone spans 6° of longitude between 174°E and 180°E, latitude in northern hemisphere between equator and 84°N, southern hemisphere between equator and 80°S.

gpkg_spatial_ref_sys Column Name : value
definition :

```
PROJCS["WGS 84 / UTM zone <zone><hemisphere>",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
            SPHEROID["WGS84",6378137,298.257223563,
                ID["EPSG","7030"]],
            ID["EPSG","6326"]],
        PRIMEM["Greenwich",0,
            ID["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            ID["EPSG","9122"]],
        ID["EPSG","4326"]],
    PROJECTION["Transverse_Mercator"],
    PARAMETER["latitude_of_origin",0],
    PARAMETER["central_meridian",<central_meridian>],
    PARAMETER["scale_factor",0.9996],
    PARAMETER["false_easting",500000],
    PARAMETER["false_northing",<false northing>],
    UNIT["metre",1,
        ID["EPSG","9001"]],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH],
    ID["EPSG","<epsg reference code>"]]

<zone> := >= 1 AND <= 60
<hemisphere> := N OR S
<central_meridian> := >= -177 AND <= 177
<false northing> := 0 for N, 10000000 for S
```

In Table-22 and Table-23, the values within <name> are variables to be replaced by the instantiation of a Lambert Conic Conformal (LCC) projection used in the GeoPackage. The allowable values for LCC projections are constrained in this profile as defined in section 7.4 Table-37.

Table 22: Lambert Conic Conformal using 1SP

gpkg_spatial_ref_sys Column Name : value
srs_name : <registered name> Lambert Conformal Conic using 1SP
srs_id : <registered LCC 1SP SRS>
organization : EPSG
organization_coordsys_id : <registered LCC 1SP CRS>
description :For a one standard parallel Lambert the natural origin of the projected coordinate system is the intersection of the standard parallel with the longitude of origin (central meridian).
definition :

```
Lambert_Conic_Conformal_using_1SP

PROJCS["<name>",
    GEODCRS | GEOGCS["<CRS Reference>",
        DATUM["<datum reference>",
            SPHEROID["<spheroid>",<semi-major axis>,<inverse flattening>]],
        PRIMEM["<prime meridian name>",<irm_longitude>],
        UNIT["degree",0.0174532925199433]],
    PROJECTION["Lambert_Conformal_Conic_1SP"],
    PARAMETER["latitude_of_origin",<latitude_of_origin>],
    PARAMETER["central_meridian",<central_meridian>],
    PARAMETER["scale_factor",<scale_factor>],
    PARAMETER["false_easting",<false_easting>],
    PARAMETER["false_northing",<false_northing>],
    UNIT["Meter",1],
    AXIS["X",EAST],
    AXIS["Y",NORTH],
    AUTHORITY["EPSG","<srs_id>"]]
```

Example Value :

```
PROJCS["Unnamed Lambert_Conformal_Conic using 1SP",
    GEOGCS["NAD83",
        DATUM["North_American_Datum_1983",
            SPHEROID["GRS 1980",6378137,298.257222101],
            AUTHORITY["EPSG","6269"]],
        PRIMEM["Greenwich",0],
        UNIT["degree",0.0174532925199433]],
    PROJECTION["Lambert_Conformal_Conic_1SP"],
    PARAMETER["latitude_of_origin",49],
    PARAMETER["central_meridian",-95],
    PARAMETER["scale_factor",1],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0],
    UNIT["Meter",1],
    AXIS["X",EAST],
    AXIS["Y",NORTH],
    AUTHORITY["EPSG","0000"]]
```

Table 23: Lambert Conic Conformal using 2SP

gpkg_spatial_ref_sys Column Name : value
srs_name : <registered name> Lambert Conformal Conic using 2SP
srs_id : <registered LCC 2SP SRS>
organization : EPSG
organization_coordsys_id : <registered LCC 2SP SRS>
description : Two standard parallels will usually be made according to the latitudinal extent of the area which it is wished to map, the parallels usually being chosen so that they each lie a proportion inboard of the north and south margins of the mapped area.
definition :

```
Lambert_Conic_Conformal_using_2SP

PROJCS["<name>",
    GEODCRS | GEOGCS["<CRS Reference>",
        DATUM["<datum reference>",
            SPHEROID["<spheroid>",<semi-major axis>,<inverse flattening>]],
        PRIMEM["<prime meridian name>",<irm_longitude>],
        UNIT["degree",0.01745329251994328]],
    PROJECTION["Lambert_Conformal_Conic_2SP"],
    PARAMETER["standard_parallel_1",<standard_parallel_1>],
    PARAMETER["standard_parallel_2",<standard_parallel_2>],
    PARAMETER["latitude_of_origin",<latitude_of_origin>],
    PARAMETER["central_meridian",<central_meridian>],
    PARAMETER["false_easting",<false_easting>],
    PARAMETER["false_northing",<false_northing>],
    AXIS[X,EAST],
    AXIS[Y,NORTH],
    AUTHORITY["EPSG","<srs_id>"]]
```

Example Value :

```
PROJCS["NAD83 / Canada Atlas Lambert",
    GEOGCS["NAD83",
        DATUM["North_American_Datum_1983",
            SPHEROID["GRS 1980",6378137,298.257222101,
                AUTHORITY["EPSG","7019"]],
            AUTHORITY["EPSG","6269"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.01745329251994328,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4269"]],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    PROJECTION["Lambert_Conformal_Conic_2SP"],
    PARAMETER["standard_parallel_1",49],
    PARAMETER["standard_parallel_2",77],
    PARAMETER["latitude_of_origin",49],
    PARAMETER["central_meridian",-95],
    PARAMETER["false_easting",0],
    PARAMETER["false_northing",0],
    AXIS["Easting",EAST],
    AXIS["Northing",NORTH],
    AUTHORITY["EPSG","3978"]]
```

7.2.3 Compound CRS WKT

Requirement 14	/req/crs/compound ***If a GeoPackage contains feature data where the gpkg_geometry_column does not allow Z geometry values, a Compound CRS SHALL not be used as a CRS definition.***

Three dimensional CRS can be defined using a Compound CRS following Table-25, Table-26, and Table-30. The Compound CRS elements are combined in this example Compound CRS WKT definition.

```
COMPOUNDCRS["WGS84 Height EGM08",
    GEODCRS["WGS 84",
        DATUM["World Geodetic System 1984",
            ELLIPSOID["WGS 84",6378137,298.257223563,
                LENGTHUNIT["metre",1.0]]],
        CS[ellipsoidal,2],
        AXIS["Geodetic latitude (Lat)",north],
        AXIS["Geodetic longitude (Long)",east],
        ANGLEUNIT["degree",0.0174532925199433],
        ID["EPSG",4326]],
    VERTCRS["EGM2008 geoid height",
        VDATUM["EGM2008 geoid",
            ANCHOR["WGS 84 ellipsoid"]],
        CS[vertical,1],
        AXIS["Gravity-related height (H)",up],
        LENGTHUNIT["metre",1.0],
        ID["EPSG",3855]],
    ID["EPSG",9518]]
```

Table 24: EGM2008 geoid height vertical CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : EGM2008 height Vertical
srs_id : 3855
organization : EPSG
organization_coordsys_id : 3855
description : Good approximation of Orthometric height above the EGM2008 model of the geoid. Replaces EGM96 geoid (CRS code 5773).
definition :

```
VERTCRS["EGM2008 geoid height",
    VDATUM["EGM2008 geoid",
        ANCHOR["WGS 84 ellipsoid"]],
    CS[vertical,1],
    AXIS["Gravity-related height (H)",up],
    LENGTHUNIT["metre",1.0],
    ID["EPSG",3855]]
```

Table 25: Compound CRS Definition Template

gpkg_spatial_ref_sys Column Name : value
srs_name : <Compound CRS Name>
srs_id : <Comp CRS Code>
organization : EPSG
organization_coordsys_id : <Comp CRS Code>
description : Geodetic position based on the World Geodetic System 1984 (WGS 84), extended by height position based on the Earth Gravity Model 2008 (EGM08).
definition :

```
COMPOUNDCRS[<Compound CRS Name>,
    <Horizontal CRS WKT>,
    <Vertical CRS WKT>,
ID["EPSG",<Comp CRS Code>]
]
```

Requirement 15	/req/crs/compound-wkt ***For 3D vector data, the compound CRS template variables <Comp CRS Code>, <Compound CRS Name>, <Horizontal CRS WKT>, and <Vertical CRS WKT> in Table-25 above SHALL be replaced with parameter values referenced in Table-26.***

The compound CRS in Table-26 uses the WGS84 for X,Y coordinates and the Vertical CRS EGM2008 (3855 in the EPSG definition) for the Z coordinate.

Table 26: Compound CRS Template Parameter Values

<Comp CRS Code>	**<Compound CRS Name>**	**<Horizontal CRS Code>**	**<Horizontal CRS WKT> Table**	**<Vertical CRS Code>**	**<Vertical CRS WKT> Table**
EPSG:9518	WGS 84 + EGM2008 height Compound	4326	Table-16	3855	Table-24

7.2.4 Gridded Data WKT

3D Gridded Data uses the Compound CRS for geographic and projected tile matrix sets. The horizontal datum of the 3D Compound CRS is either Geograpchic using WGS 84 or Projected in a UTM Zone. The vertical datum is a specific WGS geoid. For uses requiring precise location, the geographic coordinates are defined by a dynamic frame CRS and epoch date, following the WGS 84 realizations as recommended in NGA.STND.0079_1.0_3DGEO. The WKT for 3D Gridded data has a 3D CRS with WKT parameters that enable higher fidelity using Compound CRS. Following the OGC GeoPackage [4] and OGC WKT extension standards [11], the Compound CRS in GeoPackage uses a unique srs_id for each gridded coverage. The srs_id is set by the producer of a Geopackage that contains gridded coverages. A range of srs_id values is recommended by the DGIWG organization for the different compound SRS using the same group of horizontal CRS. These identifiers are necessary in DGIWG GeoPackages because the srs_id and organization_coordsys_id are associated with each gridded coverage as shown in tables 27, 28, and 29.

Gridded Compound CRS WKT

If 3D Gridded Data does not use a COMPOUNDCRS, such as EPSG:4979, the WKT must still be conformant to requirement /req/crs/wkt tested in ATS 3.5. The COMPOUNDCRS WKT for 3D gridded data is conformant per requirement /req/crs/gridded-crs-compound-wkt tested in ATS 9.3.

Requirement 16	/req/crs/gridded-crs-compound-wkt ***For 3D Gridded Data using a compound CRS in Table-14, the Compound CRS WKT SHALL use WKT of COMPOUNDCRS with valid WKT for both horizontal and vertical coordinates, referenced by a unique COMPOUNDCRS ID in the srs_id column and in WKT definition_12_063 or definition columns of the gpkg_spatial_ref_sys table as shown in tables 27, 28, 29.***

An example of a valid compound CRS for gridded data in WKT2 using a WGS 84 realization for the horizontal datum combined with EGM2008 for vertical datum are defined in Table 27. A WKT1 compound CRS definition is also valid for WGS 84 geographic coordinates if the FRAMEEPOCH element is not needed for requirement /req/crs/gridded-crs-epoch-wkt.

Table 27: Compound WGS 84 (G1762) and EGM2008 geoid height vertical CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : https://ns.dgiwg.org/def/crs/100100
srs_id : unique identifier, such as 100100

gpkg_spatial_ref_sys Column Name : value
organization : DGIWG
organization_coordsys_id : unique identifier, such as 100100
description : DGIWG Compound CRS with WGS 84. Example using WGS 84 Realization G1762 (EPSG:9057) horizontal datum and EGM2008 (EPSG:3855) vertical height above the Orthometric model of the geoid.
definition_12_063 :

```
COMPOUNDCRS["WGS 84 (G1762) + EGM2008 height",
 GEOGCRS["WGS 84 (G1762)",DYNAMIC[FRAMEEPOCH[2005]],
   DATUM["World Geodetic System 1984 (G1762)",
    ELLIPSOID["WGS 84", 6378137,298.257223563,LENGTHUNIT["metre",1]]],
   PRIMEM["Greenwich",0,
     ANGLEUNIT["degree",0.0174532925199433]],CS[ellipsoidal,2],
     AXIS["geodetic latitude (Lat)",north,ORDER[1],
     ANGLEUNIT["degree",0.0174532925199433]],
     AXIS["geodetic longitude (Lon)",east,ORDER[2],
     ANGLEUNIT["degree",0.0174532925199433]],
     USAGE[SCOPE["Geodesy. Navigation and positioning using GPS satellite
system."],
     AREA["World."],
   BOX[-90,-180,90,180]],
 ID["EPSG",9057] ],
VERTCRS
        ["EGM2008 height",
  VDATUM["EGM2008 geoid"],CS[vertical,1],
   AXIS["gravity-related height (H)",up,
   LENGTHUNIT["metre",1]],
   USAGE[SCOPE["Geodesy."],
   AREA["World."],
  BBOX[-90,-180,90,180]],
  ID["EPSG",3855]],
ID["DGIWG","100100"]]
```

An example Compound CRS for gridded data in WKT2 with projection in a single UTM Zone for the horizontal datum and a EGM96 vertical datum is defined as shown in Table 28. The GeoPackage may use any specific UTM zone with a central_meridian and false northing in North and South UTM Zones as defined by EPSG:32601-32660 and EPSG:32701-32770, repsectively. Using WKT1 is also valid for this Compound CRS.

Table 28: Compound UTM (North Zone 1) and EGM96 height vertical CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : https://ns.dgiwg.org/def/crs/100200
srs_id : unique identifier, such as 100200
organization : DGIWG
organization_coordsys_id : unique identifier, such as 100200
description : DGIWG Compound CRS with UTM Projection. Example using UTM North Zone 1 (EPSG:32601) horizontal datum and EGM96 (EPSG:5773) vertical height above the Orthometric model of the geoid.
definition_12_063 (example) :

```
COMPOUNDCRS["UTM ZONE 1N + EGM96 height",
  PROJCRS["WGS 84 / UTM zone 1N",
    BASEGEOGCRS["WGS 84",
        ENSEMBLE["World Geodetic System 1984 ensemble",
            MEMBER["World Geodetic System 1984 (Transit)"],
            MEMBER["World Geodetic System 1984 (G730)"],
            MEMBER["World Geodetic System 1984 (G873)"],
            MEMBER["World Geodetic System 1984 (G1150)"],
            MEMBER["World Geodetic System 1984 (G1674)"],
            MEMBER["World Geodetic System 1984 (G1762)"],
            MEMBER["World Geodetic System 1984 (G2139)"],
            ELLIPSOID["WGS 84",6378137,298.257223563,
                LENGTHUNIT["metre",1]],
            ENSEMBLEACCURACY[2.0]],
        PRIMEM["Greenwich",0,
            ANGLEUNIT["degree",0.0174532925199433]],
        ID["EPSG",4326]],
    CONVERSION["UTM zone 1N",
        METHOD["Transverse Mercator",
            ID["EPSG",9807]],
        PARAMETER["Latitude of natural origin",0,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8801]],
        PARAMETER["Longitude of natural origin",-177,
            ANGLEUNIT["degree",0.0174532925199433],
            ID["EPSG",8802]],
        PARAMETER["Scale factor at natural origin",0.9996,
            SCALEUNIT["unity",1],
            ID["EPSG",8805]],
        PARAMETER["False easting",500000,
            LENGTHUNIT["metre",1],
            ID["EPSG",8806]],
```

```
        PARAMETER["False northing",0,
            LENGTHUNIT["metre",1],
            ID["EPSG",8807]]],
    CS[Cartesian,2],
        AXIS["(E)",east,
            ORDER[1],
            LENGTHUNIT["metre",1]],
        AXIS["(N)",north,
            ORDER[2],
            LENGTHUNIT["metre",1]],
    USAGE[
        SCOPE["Navigation and medium accuracy spatial referencing."],
        AREA["Between 180°W and 174°W, northern hemisphere between equator
and 84°N, onshore and offshore. Russian Federation; United States (USA) -
Alaska (AK)."],
        BBOX[0,-180,84,-174]],
    ID["EPSG",32601]],
  VERTCRS["EGM96 height",
    VDATUM["EGM96 geoid"],
    CS[vertical,1],
        AXIS["gravity-related height (H)",up,
            LENGTHUNIT["metre",1]],
    USAGE[
        SCOPE["Geodesy."],
        AREA["World."],
        BBOX[-90,-180,90,180]],
    ID["EPSG",5773]],
 ID["DGIWG","100200"]]
```

An example Compound CRS for gridded data in WKT1 with projection in the Universal Polar Stereographic (UPS) South for the horizontal datum and a EGM96 vertical datum is defined as shown in Table 29. Using WKT2 is also valid for this Compound CRS.

Table 29: Compound UPS (South) and EGM96 height vertical CRS Definition

gpkg_spatial_ref_sys Column Name : value
srs_name : https://ns.dgiwg.org/def/crs/100300
srs_id : unique identifier, such as 100300
organization : DGIWG
organization_coordsys_id : unique identifier, such as 100300

gpkg_spatial_ref_sys Column Name : value
description : DGIWG Compound CRS with UPS Projection. Example using UPS South (EPSG:5042) horizontal datum and EGM96 (EPSG:5773) vertical height above the Orthometric model of the geoid.
definition_12_063 (example) :

```
COMPOUNDCRS["UPS SOUTH + EGM96 height",
  PROJCS["WGS 84 / UPS South (E,N)",
    GEOGCS["WGS 84",
        DATUM["WGS_1984",
            SPHEROID["WGS 84",6378137,298.257223563,
                AUTHORITY["EPSG","7030"]],
            AUTHORITY["EPSG","6326"]],
        PRIMEM["Greenwich",0,
            AUTHORITY["EPSG","8901"]],
        UNIT["degree",0.0174532925199433,
            AUTHORITY["EPSG","9122"]],
        AUTHORITY["EPSG","4326"]],
    PROJECTION["Polar_Stereographic"],
    PARAMETER["latitude_of_origin",-90],
    PARAMETER["central_meridian",0],
    PARAMETER["scale_factor",0.994],
    PARAMETER["false_easting",2000000],
    PARAMETER["false_northing",2000000],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AUTHORITY["EPSG","5042"]],
  VERT_CS["EGM96 height",
    VERT_DATUM["EGM96 geoid",2005,
        AUTHORITY["EPSG","5171"]],
    UNIT["metre",1,
        AUTHORITY["EPSG","9001"]],
    AXIS["Gravity-related height",UP],
    AUTHORITY["EPSG","5773"]],
ID["DGIWG","100300"]]
```

The WKT defines the actual composition of the Compound CRS. To allow for multiple combinations of horizontal CRS and veritical CRS, as well as multiple epoch dates (described below) within the same GeoPackage, a unique srs_id value for gridded data are required for each CRS the gpkg_spatial_ref_sys table. The 3D Compound CRS for the DGIWG organization are defined at https://www.dgiwg.org/def/crs for each of the 3 types of compound CRS with the srs_id values of 100100, 100200, and 100300. As defined by the DGIWG organization, a range of srs_id values are recommended, but not required, based upon the type of horizontal CRS, in

these recommendations.

Recommendation 1	/recco/gridded-cmpnd-crs-wgs84 ***For each gridded coverage in GeoPackage that has a 3D Compound CRS of WGS84 geographic coordinates, in order to reference DGIWG's identification of the CRS, use a unique value within the range of 100100-100199 for the srs_id column, the organization_coordsys_id column, and ID# in the* ID["DGIWG","ID#"] *field of the Compound CRS WKT in the definition and definition_063 columns of the gpkg_spatial_ref_sys table.***

Recommendation 2	/recco/gridded-cmpnd-crs-utm ***For each gridded coverage in GeoPackage that has a 3D Compound CRS of Universal Transverse Mercator (UTM) projected coordinates, in order to reference DGIWG's identification of the CRS, use a unique value within the range of 100200-100299 for the srs_id column, the organization_coordsys_id column, and ID# in the* ID["DGIWG","ID#"] *field of the Compound CRS WKT in the definition and definition_063 columns of the gpkg_spatial_ref_sys table.***

Recommendation 3	/recco/gridded-cmpnd-crs-ups ***For each gridded coverage in GeoPackage that has a 3D Compound CRS of Universal Polar Stereographic (UPS) projected coordinates, in order to reference DGIWG's identification of the CRS, use a unique value within the range of 100300-100399 for the srs_id column, the organization_coordsys_id column, and the ID# in the* ID["DGIWG","ID#"] *field of the Compound CRS WKT in the definition and definition_063 columns of the gpkg_spatial_ref_sys table.***

Epoch Date

DGIWG-126 version 1.1 enables implementations to use the *Coordinate Epoch* to calculate the horizontal location of gridded data while accounting for movement of the earth's surface. The WKT extension [11] (OGC 21-057) required for gridded data in section 6.3 allows a coordinate epoch date to be included in gpkg_spatial_ref_sys table. Requirement /req/gridded-crs-epoch-wkt applies when the gridded data has an Epoch date and the FRAMEEPOCH is set in the compound CRS WKT for gridded data. In the sample 3D Compound CRS in Table 27, the DYNAMIC[FRAMEEPOCH[2005]] defines the Epoch date "2005" for the horizontal datum "WGS 84 (G1762) - EPSG:9057," which will accompany the EPOCH column value for each layer referenced

by EPSG:9057 in the gpkg_spatial_ref_sys table.

The Epoch requirement is compliant only if the DYNAMIC FRAMEEPOCH element is included in the horizonatal CRS WKT, which excludes the projected UTM CRS and UPS CRS used in Table 28 and Table 29. Besides gridded data, though raster and vector CRS allow the DYNAMIC and FRAMEEPOCH element in WKT2, no requirements regarding epoch are specified for raster and vector data in this GeoPackage profile, DGIWG-126 version 1.1, subject to future revision.

Requirement 17	/req/crs/gridded-crs-epoch-wkt ***When applicable to a compound CRS in Table-14, if the coordinate epoch field in gpkg_spatial_ref_sys as defined in [11] (OGC 21-057) is set to a valid date, the WKT for the Gridded Data CRS shall include the DYNAMIC FRAMEEPOCH with a valid epoch date of the applicable CRS.***

Examples of the coordinate *epoch date* values in the gpkg_spatial_ref_sys table for WGS84 realizations are listed in Table 30 below and described in Annex E footnote [5] and Annex E footnote [6]. The Epoch value should use four significant digits for precision of a specific date.

Table 30: WGS 84 Realizations and Coordinate Epoch Dates

EPSG 2D	EPSG 3D	WGS8 84 realization	CRS origin Epoch value	Example Coordinate Date	# days	Example EPOCH value
4326	4979	WGS 84	1984.00	01/01/1990	1	1990.0027
9053	7657	WGS 84 (G730)	1997.00	12/31/2020	365	2021.00
9054	7659	WGS 84 (G873)	1997.00	10/02/2012	275	2012.7514
9055	7661	WGS 84 (G1150)	2001.00	10/02/2023	274	2023.7507
9056	7663	WGS 84 (G1674)	2005.00	02/05/2024	36	2024.0984
9057	7665	WGS 84 (G1762)	2005.00	03/25/2016	86	2016.2350
9755	9754	WGS 84 (G2139)	2016.00	03/25/2017	85	2017.2329

For the GeoPackage CRS to reference a WGS84 realization, the GeoPackage producer uses one of

the applicable realizations of WGS84 and EPSG codes in Table 14, such as WGS84 2D G873 / EPSG:9054 and G1150 / EPSG:9055. The FRAMEEPOCH is defined in the WKT for the Compound CRS.

7.3 Metadata

This clause describes the storage of metadata using the metadata extension F.8 to embed XML in GeoPackages. The DGIWG GeoPackage profile specifies the use of particular metadata values in certain conditions per the DGIWG Metadata Foundation (DMF) and associated metadata files. Optionally, metadata from a national profile (e.g. U.S. NSG Metadata Foundation (NMF)) can be included in a GeoPackage in addition to DMF.

The structure and allowed content values of DGIWG metadata for a GeoPackage are specified by the DMF. Examples of the creation use, and exchange of metadata in GeoPackage are described in these use cases: D-2 Disadvantaged, mobile, and autonomous users, D-3 Data Exchange between processing sites and D-6 Workflow for generating a GeoPackage.

A DGIWG GeoPackage is required to contain a DMF metadata instance for the *Entire GeoPackage* as defined in the tables described in this section, and it may optionally contain metadata instances of additional national metadata profiles.

The metadata in a GeoPackage is defined in two related tables, gpkg_metadata and gpkg_metadata_reference, which define metadata for the entire GeoPackage, a series of GeoPackage data, and optionally for subsets of the GeoPackage contents, referred to in this specification as a *Partial GeoPackage*. The requirements and related tables that follow in this section define the DGIWG implementation of metadata using the structure defined in the OGC GeoPackage standard.

Requirement 18	/req/metadata/dmf ***The metadata for the entire GeoPackage and, if present, metadata for a part of the GeoPackage vector layers or tile matrices SHALL include one or more complete metadata instance of DGIWG Metadata Foundation (DMF) 2.0 or later version.***

Annex B provides an informative metadata reference with mapping of DMF metadata elements to NMF core metadata, and examples of metadata content for each format.

Requirement 19	/req/metadata/gpkg ***A GeoPackage SHALL contain one or more complete metadata XML instance document that describes the entire GeoPackage in a gpkg_metadata table row with the contents shown in Table-31 below.***

Table 31: gpkg_metadata Table Contents - Required for Entire GeoPackage

Column Name	Column Description	Column Value
id	Metadata primary key	unique integer value
md_scope	Case sensitive name of the data scope to which this metadata applies	'series' or 'dataset'
md_standard_uri	URI reference to the metadata structure definition authority	REQUIRED URI location of the DGIWG standard for use in Table-32 and Table-34
mime_type	MIME encoding of metadata	text/xml
metadata	metadata	REQUIRED DMF metadata instance document

If national metadata is contained for the entire GeoPackage file or a series of GeoPackage files, the national metadata will be included according to the contents of Table-32. The national metadata is optional; but if populated, the contents are not NULL.

Table 32: gpkg_metadata Table Contents - Optional Metadata for Entire GeoPackage

Column Name	Column Description	Column Value
id	Metadata primary key	unique integer value
md_scope	Case sensitive name of the data scope to which this metadata applies	'series' or 'dataset'
md_standard_uri	URI reference to the metadata structure definition authority	URI location of the national metadata profile (ex. US NMF/NMIS)

Column Name	Column Description	Column Value
mime_type	MIME encoding of metadata	text/xml
metadata	metadata	Optional National profile metadata instance document

Requirement 20	/req/metadata/row ***The gpkg_metadata table row with the values shown in Table-31 SHALL be associated with the complete GeoPackage by a gpkg_metadata_reference table row with the contents shown in Table-32.***

Table 33: gpkg_metadata_reference Table Contents - Entire GeoPackage

Column Name	Column Description	Column Value
reference_scope	Lowercase metadata reference scope	'geopackage'
table_name	NULL for reference_scope of 'geopackage'.	NULL
column_name	Name of the column to which this metadata reference applies; NULL for reference_scope of 'geopackage'	NULL
row_id_value	NULL for reference_scope of 'geopackage'	NULL
timestamp	timestamp value in ISO 8601format as defined by the strftime function '%Y-%m-%dT%H:%M:%fZ' format string applied to the current time	strftime('%Y-%m-%dT%H:%M:%fZ','now')
md_file_id	gpkg_metadata table id column value for the metadata to which this gpkg_metadata_reference applies	unique integer value, id field value for the gpkg_metadata contents in Table-31 or Table-32

Column Name	Column Description	Column Value
md_parent_id	NULL if md_file_id forms the root of a metadata hierarchy	file id field value for the previous gpkg_metadata_reference contents, or NULL

A GeoPackage MAY contain additional metadata XML instance documents or fragments for a smaller subset of the GeoPackage that describe particular feature layers, TileMatrixSet layers, or other entities contained within the GeoPackage. For implementation of *Partial GeoPackage* metadata, the gpkg_metadata_reference table can identify other user-defined tables containing information about a subset of the GeoPackage contents for featureType, feature, attributeType, attribute, model (tileSet), and tile. More description of tile and feature metadata is written in sections 8.4, Tile Metadata, and 9.1, Feature Metadata.

An example of the tables used to define metadata for vector feature types in the GeoPackage is illustrated in three figures using the md_scope value *featureType* defined in Table-36. The GeoPackage geometry and attributes for each feature are contained in the user-defined tables roads and bridges. Users of the profile can store *featureType* metadata using the nominal definition of metadata tables depicted in Figure-1. The gpkg_metadata_reference identifies the rows in gpkg_metadata for the feature types in the geometry tables, roads_geom and bridges_geom. Alternatively, a user-defined table of feature types could be added as shown in Figure-2. In the second example, the GeoPackage populates the same rows of the gpkg_metadata table for two feature types. However, the instance in Figure-3 extends the nominal use of gpkg_metadata_reference to populate table_name and column_name columns for the user-defined table Feature_types, which contains feature type names. Beyond these examples, additional metadata for the roads and bridges tables could be populated in gpkg_metadata and gpkg_metadata_reference with a md_scope value of *feature*, *attribute_type*, or *attribute* (not depicted in these examples).

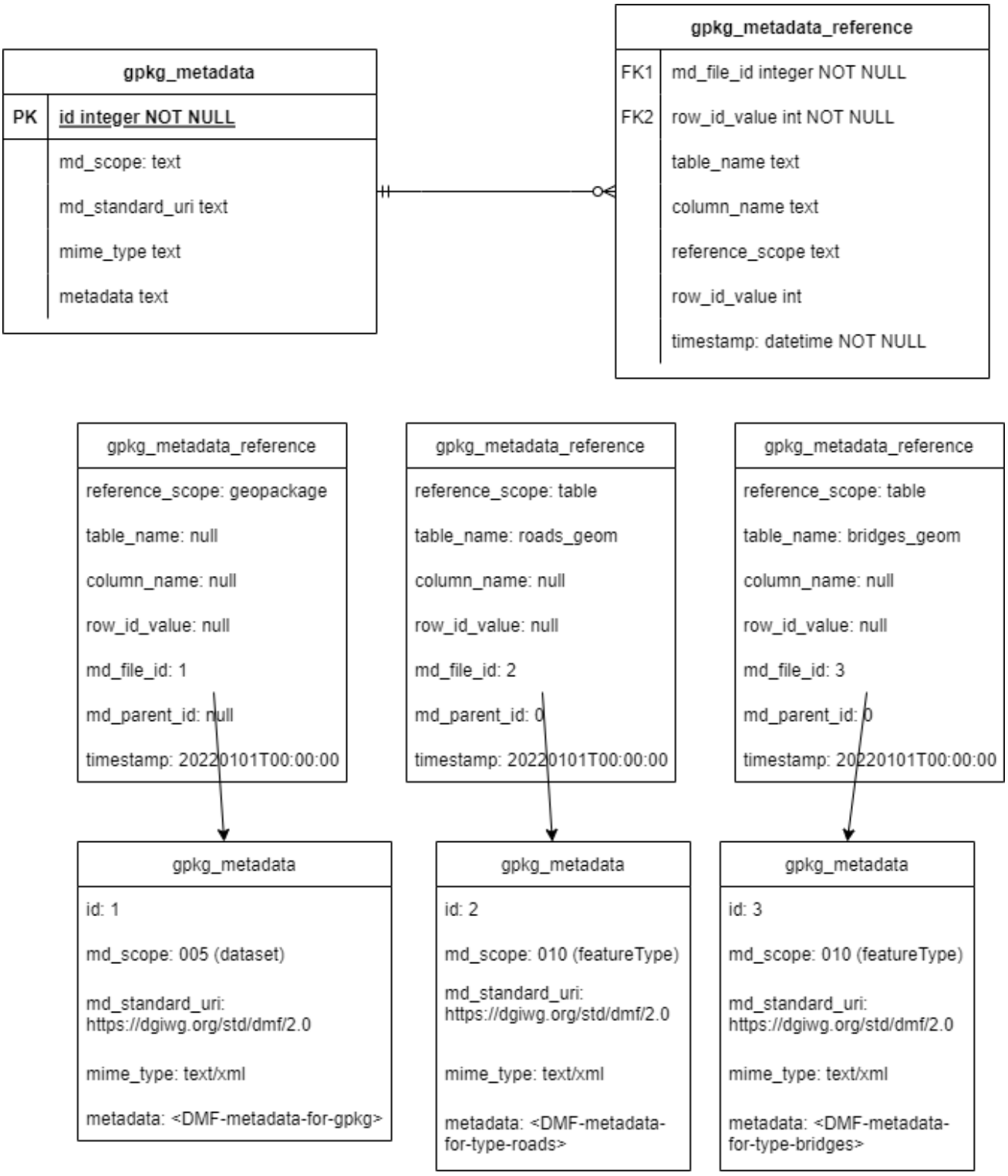

Figure 1: GeoPackage Feature Type Metadata Nominal Example

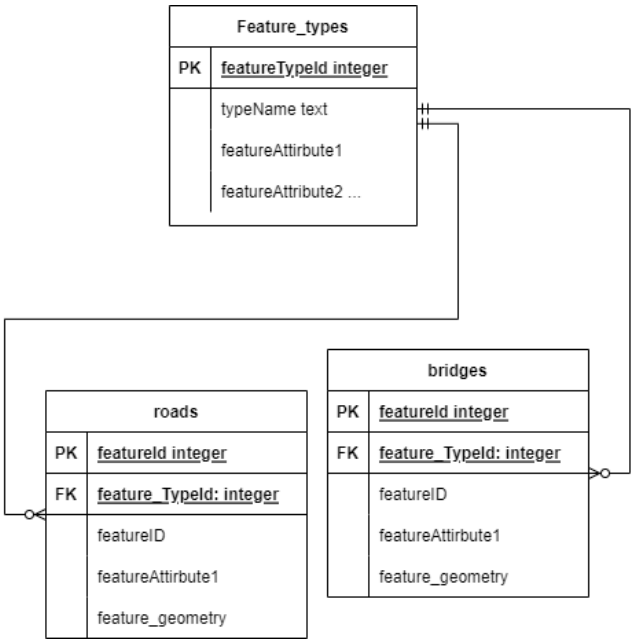

Figure 2: GeoPackage Feature Type Metadata Extended Example User Tables

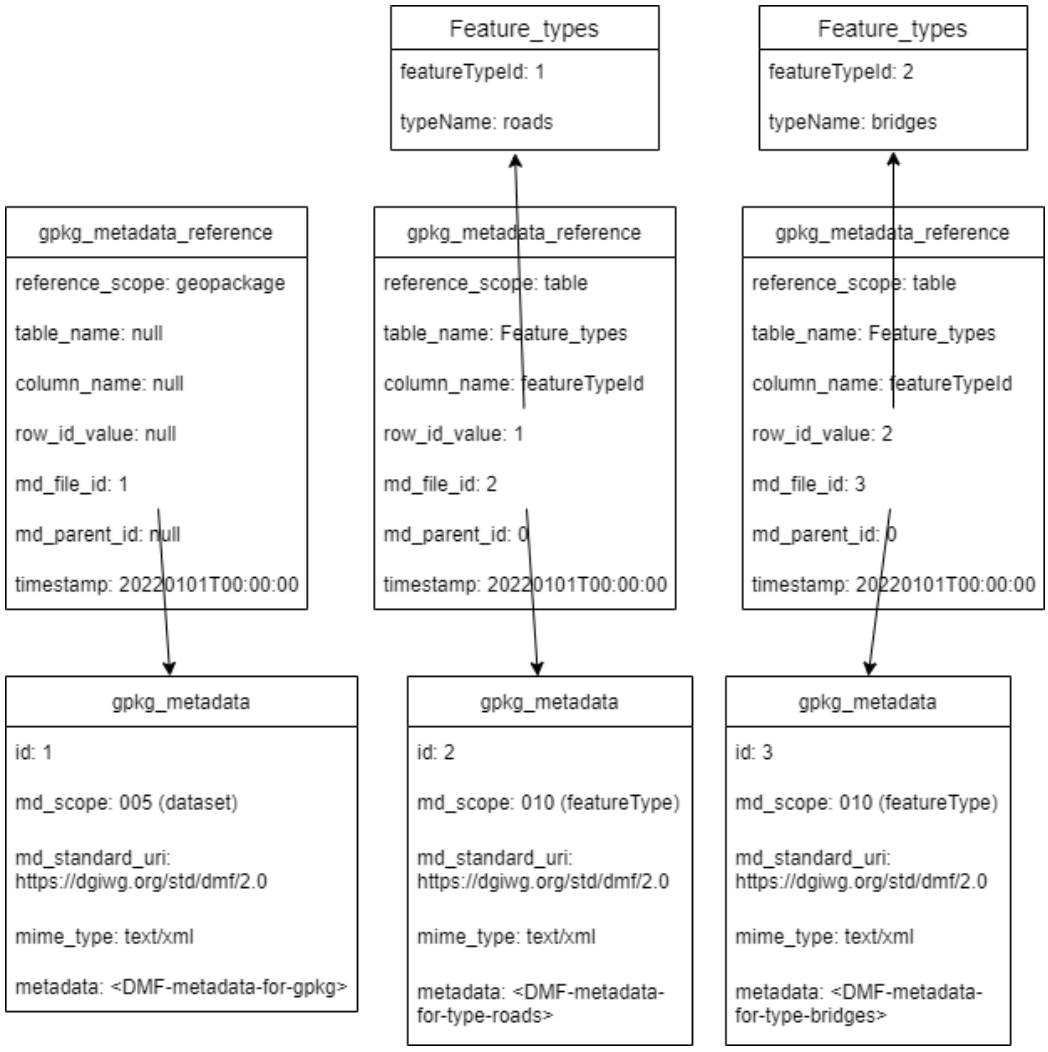

Figure 3: GeoPackage Feature Type Metadata Extended Example Instance

As illustrated in these figures, gpkg_metadata and gpkg_metadata_reference can optionally be utilized to contain the metadata for any *Partial GeoPackage* content. The metadata contained for reference_scope values of table, row, column, or row/col in Table-36 is specified for a GeoPackage in the gpkg_metadata contents in Table-34 and the corresponding gpkg_metadata_reference in Table-35.

Table 34: gpkg_metadata Table Contents - Partial GeoPackage

Column Name	Column Description	Column Value
id	Metadata primary key	unique integer value
md_scope	Case sensitive name of the data scope to which this metadata applies	See Table-36
md_standard_uri	URI reference to the metadata structure definition authority	REQUIRED URI location of the applicable DMF version OPTIONAL URI location of national metadata profile (ex. US NMIS)
mime_type	MIME encoding of metadata	text/xml
metadata	metadata	REQUIRED DMF metadata instance document OPTIONAL National metadata profile instance document

Requirement 21	/req/metadata/user ***If a GeoPackage contains metadata for vector layers or tile matrices, any gpkg_metadata table row with values AND applies to multiple GeoPackage table, row and/or column content items shown in Table-34 SHALL be linked to the specific row and/or column in that table by the md_file_id in gpkg_metadata_reference AND those content items by one or more gpkg_metadata_reference table rows with the content specified in Table-35.***

The connection between gpkg_metadata and gpkg_metadata_reference tables for GeoPackage user metadata in Requirement /req/metadata/user is the same type of connection for the whole GeoPackage in Requirement /req/metadata/row. This relationship is verified in ATS-4.3 and ATS-4.4.

Table 35: gpkg_metadata_reference Table Contents - Product Partial GeoPackage

Column Name	Column Description	Column Value
reference_scope	Lowercase metadata reference scope; one of ‘table’,’column’, ‘row’, ‘row/col’	See Table-36 below
table_name	Name of the table to which this metadata reference applies, not applicable unless optionally used for reference_scope of ‘table', ‘row’, 'column' or 'row/col' to describe a featureType, feature, tileSet, tile, attributeType, or attribute; else NULL.	As described, set to NULL if not applicable

Column Name	Column Description	Column Value
column_name	Name of the column to which this metadata reference applies; not applicable unless optionally used for reference_scope of ‘table’, 'row/col' or ‘row’, or the name of a column in the table_name table for reference_scope of ‘column’ or ‘row/col’ to describe a feature, tile, attributeType, or attribute; or optionally a column name in the table_name table for reference_scope of ‘table’ and gpkg_metadata.md_scope value of 'featureType', 'model' (tileSet), or 'attributeType'	As described, set to NULL if not applicable
row_id_value	NULL for reference_scope of ‘table’ or ‘column’ with NULL "table_name", or the row id of a row record in the table_name table for reference_scope of ‘table’, ‘column’, 'row' or ‘row/col’	As described, set to NULL if not applicable
timestamp	timestamp value in ISO 8601format as defined by the strftime function ‘%Y-%m-%dT%H:%M:%fZ’ format string applied to the current time	strftime(‘%Y-%m-%dT%H:%M:%fZ’,’now’)
md_file_id	gpkg_metadata table id column value for the metadata to which this partial gpkg_metadata_reference applies	unique integer value, id field value for the partial gpkg_metadata contents in Table-34

Column Name	Column Description	Column Value
md_parent_id	gpkg_metadata table id column value for the hierarchical parent gpkg_metadata for the gpkg_metadata to which this gpkg_metadata_reference applies, or NULL if md_file_id forms the root of a metadata hierarchy	file id field value for the previous partial gpkg_metadata_reference contents, or NULL

The md_scope and corresponding reference scope values for the DGIWG implementation of metadata are defined in Table-36. The scope code values for md_scope come from the GeoPackage specification [4], Table 21, Metadata Scopes. The enumeration for a *tileSet* is added in this DGIWG profile using the md_scope value for *model* because the tileSet code is not defined in the Metadata Scopes of the GeoPackage standard. Additional metadata_scope values for the 2d_gridded_coverage extension is defined in section 10.2, Table-40.

Table 36: GeoPackage and metadata scopes

gpkg_metadata. md_scope (scope code)	gpkg_metadata_reference. reference_scope
series (006)	table
dataset (005)	geopackage
featureType (010)	table
feature (009)	row
model (015) for a tileSet	table
tile (016)	row/col
attributeType (002)	column
attribute (001)	row/col

A GeoPackage may constitute a single product or a collection of products in one GeoPackage. The metadata content is capable of describing products within a GeoPackage. A hierarchy of product metadata is defined by the use of the md_parent_id in the GeoPackage metadata reference table (gpkg_metadata_reference). The following Recommendation and two requirements (/req/metadata/product, /req/metadata/product-partial) address use of metadata for one or more products in a GeoPackage.

Recommendation 4	/recco/metadata-product ***If there is different kind of data (e.g. raster tiles and vector features, or aggregation of data from various origins) in the Geopackage, there should be a dataset metadata for each kind of data using the scope "table", in addition to the metadata for the whole Geopackage.***

Requirement 22	/req/metadata/product ***If an entire GeoPackage constitutes a data product, then the component elements above that describe it SHALL be in the metadata XML that applies to the entire GeoPackage as shown in Table-31.***

Requirement 23	/req/metadata/product-partial ***If particular tables in a GeoPackage constitute separate data products, those parts of Geopackage SHALL be described by a metadata element as defined in Table-34, and the component elements that describe those data products SHALL reference the appropriate subpart(s) of Geopackage as shown in Table-35.***

7.4 Data Validity Constraints

This clause specifies constraints on allowable data values in GeoPackage SQL tables to enable assessment and enforcement of data validity. Its requirements are conditional based on whether a GeoPackage contains features or tiles. It applies to tables specified in the table name column of Table-37 below.

GeoPackage Infrastructure and Applications MAY

- Maintain data validity after every SQL command that changes GeoPackage data when this is a critical requirement (e.g. for data extent or metadata). Doing so (e.g. with SQL triggers) can cause significant performance degradation.
- Maintain data validity after a sequence of SQL commands, or after some application-specific unit of work, or upon application startup and/or termination, when this is not a critical requirement. Doing so (e.g. with a background process or a “batch” job upon application termination) can provide significant performance enhancement.

Requirement 24	/req/validity/data-validity ***Data validity SHALL be assessed against data value constraints specified in Table-37 below using a test suite. Data validity MAY be enforced by SQL triggers.***

Table 37: Data Validity Constraints

ID#	Table Name	Column Name	Value Constraints
1	“gpkg_spatial_ref_sys”	“organization”	“EPSG”
2	“gpkg_spatial_ref_sys”	“description”	NOT NULL, not an empty string, not all whitespace, not “unknown” (any case), not “tbd” (any case). For CRS specified in section 7.2 above, “description” column value from Table-15 through Table-25

ID#	Table Name	Column Name	Value Constraints
3	“gpkg_spatial_ref_sys”	“definition”, "definition_12_063"	If PROJECTION is Lambert_Conformal_Conic_1SP, then DATUM = WGS84, or European_Terrestrial_Reference_System_1989, or North_American_Datum_1983. SPHEROID = WGS84, GRS 1980 PRIMEM = Greenwich, <irm_longitude> value between 0.0 and 359.0. PARAMETER – values according to a valid Lambert Conic Conformal projection for 1 standard parallel in Table-22 in section 7.2. PROJECTION AUTHORITY = Valid EPSG code for this LCC 1SP projection.
4	“gpkg_spatial_ref_sys”	“definition”, "definition_12_063"	If PROJECTION is Lambert_Conformal_Conic_2SP, then DATUM = WGS84, or European_Terrestrial_Reference_System_1989, or North_American_Datum_1983. SPHEROID = WGS84, GRS 1980 PRIMEM = Greenwich, <irm_longitude> value between 0.0 and 359.0. PARAMETER – values according to a valid Lambert Conic Conformal projection for 2 standard parallels in Table-23 in section 7.2. PROJECTION AUTHORITY = Valid EPSG code for this LCC 2SP projection.
5	“gpkg_spatial_ref_sys”	“definition_12_063”	Compatible with WKT1
6	“gpkg_contents”	“data_type”	“features” or “tiles” or an implementer-defined value for other data tables in an Extended GeoPackage.

ID#	Table Name	Column Name	Value Constraints
7	“gpkg_contents”	“min_x”	NULL if no information available, Otherwise: If gpkg_contents.data_type = “features”, value SHALL be >= the minimum ST_MinX({geom_col}) value from gpkg_contents.table_name. If gpkg_contents data_type = “tiles”, value SHALL be >= gpkg_tile_matrix_set.min_x column value for gpkg_contents.table_name.
8	“gpkg_contents”	“min_y”	NULL if no information available. Otherwise: If gpkg_contents.data_type = “features”, value SHALL be >= the minimum ST_MinY({geom_col}) value from gpkg_contents.table_name. If gpkg_contents data_type = “tiles”, value SHALL be >= gpkg_tile_matrix_set.min_y column value for gpkg_contents.table_name.
9	“gpkg_contents”	“max_x”	NULL if no information available. Otherwise: If gpkg_contents.data_type = “features”, value SHALL be <= the maximum ST_MaxX({geom_col}) value from gpkg_contents.table_name. If gpkg_contents data_type = “tiles”, value SHALL be <= gpkg_tile_matrix_set.max_x column value for gpkg_contents.table_name.
10	“gpkg_contents”	“max_y”	NULL if no information available. Otherwise: If gpkg_contents.data_type = “features”, value SHALL be <= the maximum ST_MaxY({geom_col}) value from gpkg_contents.table_name. If gpkg_contents data_type = “tiles”, value SHALL be <= gpkg_tile_matrix_set.max_y column value for gpkg_contents.table_name.

ID#	Table Name	Column Name	Value Constraints
11	“gpkg_geometry_columns”	“z”	not equal to 2: z values optional is prohibited
12	“gpkg_geometry_columns”	“z”	0: z values prohibited is required for all 2-D CRS, i.e. where definition column of gpkg_spatial_ref_sys with same srs_id value contains WKT definition starting with “GEODCRS” or “PROJCRS” per WKT1
13	“gpkg_geometry_columns”	“z”	1: z values mandatory is required for all 3-D CRS, i.e. where definition column of gpkg_spatial_ref_sys with same srs_id value contains WKT definition starting with “GEODCRS” or “COMPOUNDCRS” with either “GEODCRS” or “PROJCRS” and “VERTCRS” definitions per WKT1
14	Every gpkg_geomtry_columns table_name column value vector feature user data table name feat_table	Corresponding gpkg_geometry_columns column_name column value geom_col name	If corresponding gpkg_geometry_columns.z = 0, **SELECT** COUNT () ***FROM** feat_table **WHERE** ST_Is3D(geom_col) SHALL return 0. If z = 1, **SELECT** COUNT () ***FROM** feat_table **WHERE NOT** ST_Is3D(geom_col) SHALL return 0.
15	“gpkg_data_columns”	“constraint_name”	Column value constraint name may be mixed case (not all lowercase as in GeoPackage Annex F.9).
16	“gpkg_tile_matrix”	“zoom_level”	0 <= zoom_level <= max_level.
17	“gpkg_tile_matrix”	“tile_width”	256
18	“gpkg_tile_matrix”	“tile_height”	256
19	“gpkg_tile_matrix”	“pixel_x_size”	SHALL vary by a factor of 2 between all adjacent zoom levels for the same table_name.
20	“gpkg_tile_matrix”	“pixel_y_size”	SHALL vary by a factor of 2 between all adjacent zoom levels for the same table_name.
21	Every gpkg_contents table_name column value where data_type = “tiles”	“zoom_level”	0 < = zoom_level <= max_level.
22	“gpkg_metadata”	“id”	Unique value SHALL be used only for DGIWG DMF or other national (e.g. NSG NMF) Metadata versions for entire GeoPackage.

ID#	Table Name	Column Name	Value Constraints
23	“gpkg_metadata”	“md_scope”	equal to scope code for “series” or “dataset” for id = primary key of root of metadata hierarchy; else equal to scope code for "featureType", "feature", "model", "tile", "attributeType", or "attribute".
24	“gpkg_metadata”	“md_standard_uri”	valid URI for DGIWG DMF equal to "https://dgiwg.org/std/dmf/2.0", or higher DMF version, or an optional national (e.g. NSG NMF) Metadata uri for id = primary key of root of metadata hierarchy.
25	“gpkg_metadata”	“mime_type”	“text/xml”
26	“gpkg_metadata”	“metadata”	DGIWG DMF or national metadata (e.g. NSG NMF/NMIS) metadata instance document.
27	“gpkg_metadata_reference”	“reference_scope”	“geopackage” for md_file_id = primary key of root of metadata hierarchy; else equal "table", "row", "row/col", or "column".
28	“gpkg_metadata_reference”	“table_name”	NULL for md_file_id = primary key of root of metadata hierarchy; NOT NULL if reference_scope = "table", "row", "column", or "row/col"; else, NULL.
29	“gpkg_metadata_reference”	“column_name”	NULL for md_file_id = primary key of root of metadata hierarchy; NOT NULL for the column_name in the "table_name" used in gpkg_metadata_reference if reference_scope = "column" or "row/col"; else, NULL.
30	“gpkg_metadata_reference”	“row_id_value”	NOT NULL for the row_id in the "table_name" used in gpkg_metadata_reference for reference_scope = "table", "column", "row" or "row/col"; else, NULL.
31	“gpkg_metadata_reference”	“timestamp”	timestamp value in ISO 8601 format.
32	“gpkg_metadata_reference”	“md_file_id”	unique integer value required equal to the id field value for the gpkg_metadata to which it applies.
33	“gpkg_metadata_reference”	“md_parent_id”	NULL for md_file_id = primary key of root of metadata hierarchy; else "0" if not used in metadata hierarchy; else gpkg_metadata id value for parent to which this gpkg_metadata_reference applies in metadata hierarchy.

8 Tiles

8.1 Tile Size

This clause mandates 256x256 tiles in any tile pyramid user data tables defined in accordance with OGC GeoPackage Clause 2.2.8 that contain DGIWG data.

As noted in [4] GeoPackage section 2.2.7, Tile Matrix footnote [K23], when tiles on the edge of the bounding box at a particular zoom level are within but not equal to the bounding box, then the non-image area of matrix edge tiles must be padded with no-data values, preferably transparent ones.

Requirement 25	/req/tile/size-matrix ***The gpkg_tile_matrix table SHALL contain tile_width and tile_height column values of 256 for every table_name tile pyramid data table.***

Requirement 26	/req/tile/size-data ***Every tile_data tile in every table_name tile pyramid data table shall have a width and height of 256 pixels.***

8.2 Zoom Levels and Tile Matrix Sets

This clause specifies that pixel sizes vary by a factor of 2 between all adjacent zoom levels, adopting the commonly used "zoom times two" convention, and precluding the use of "zoom other intervals" conventions. It applies to any tile pyramid user data tables defined in accordance with Clause 2.2.8 [4] that contain DGIWG data.

Requirement 27	/req/zoom/factor ***The gpkg_tile_matrix table SHALL contain pixel_x_size and pixel_y_size column values that differ by a factor of 2 between all adjacent zoom levels for each tile pyramid data table per OGC GeoPackage Clause 2.2.3. It SHALL NOT contain pixel sizes that vary by irregular intervals or by regular intervals other than a factor of 2 between adjacent zoom levels per OGC GeoPackage Clause 2.2.3.***

Requirement /req/crs/raster-tile-matrix-set restricts the Tile Matrix Sets (formerly called Well-Known Scale Sets (WKSS)) used in GeoPackage raster tiles as defined in Annex D of [6] OGC 17-083r2 for the Tile Matrix Sets referenced in this specification. An exception to the <req-zoom-matrix-sets-multiple, /req/zoom/matrix-sets-multiple>> requirement is allowed for tile matrix sets that represent a single scale product, such as raster charts using the Lambert Conformal Conic projection, expressed in requirement /req/zoom/matrix-sets-one.

For requirement /req/zoom/matrix-sets-multiple, because requirement /req/zoom/factor requires all zoom levels to be adjacent, a subset of zoom levels from the OGC 2D TMS [6] must all be adjacent (i.e. consecutive). For example, a single tile matrix set with a subset of zoom levels for EPSG:3395 (World Mercator) could not contain levels 1-4 and 8-12 because levels 4 and 8 are not adjacent levels. However, these two ranges of zoom levels could be in two distinct tile matrix sets: one set for zoom levels 1-4 and another set for zoom levels 8-12. Requirement /req/zoom/matrix-sets-multiple is not applicable to tile_matrix sets with a CRS that is not in [6] OGC 17-083r2.

Requirement 28	/req/zoom/matrix-sets-multiple ***If the GeoPackage tile matrix has more than one zoom level for a CRS in [6] OGC 17-083r2, the gpkg_tile_matrix table SHALL contain zoom levels that are in all or a subset of consecutive zoom levels in the tile matrix sets for the respective CRS defined in OGC 17-083r2 Annex D, TileMatrixSets.***

Requirement 29	/req/zoom/matrix-sets-one ***For a GeoPackage tile matrix set with only one zoom level, the gpkg_tile_matrix table SHALL contain a single tile matrix set for the scale of the product (e.g., 1:12,500, 1:50,000, 1:100,000).***

8.3 Bounding Boxes

This clause specifies how bounding boxes should be used to support the global tile indexing scheme that is specified in [6] OGC 2D Tile Matrix Set, 17-083r2, the 2D Tile Matrix Set.

Requirement 30	/req/bbox/crs ***The (min_x, min_y, max_x, max_y) values in the gpkg_tile_matrix_set table SHALL be within or equal to the maximum bounds of the CRS specified for the tile pyramid data table.***

As stated in [4] GeoPackage section 1.1.3.1.2 Data, after requirement 15, the bounding box (min_x, min_y, max_x, max_y) provides an *informative* bounding box of the content. In usage of the GeoPackage, the bounding box of the gpkg_tile_matrix_set provides the latitude and longitude extent that enables applications to determine the geographic position of each tile in the tile pyramid data table.

8.4 Tile Layer Metadata

This clause requires the storage of metadata as XML in GeoPackages and specifies the use of particular metadata values in certain conditions per the DMF and optionally national metadata such as NMF.

The metadata in this section describes the individual layers of tiled imagery or other coverages contained in the GeoPackage. As stated in section 7.3 Requirement /req/metadata/product-partial, "If particular tables in a GeoPackage constitute separate data products in Table-34, then the component elements that describe those data products SHALL be in the metadata XML as shown in Table-35 … ." The columns in Table-34 are repeated as a row with a unique id value and md_scope of "model" for each tileSet, and md_scope of "tile" for each tile. This section gives an example of metadata that applies to Tile layers and tiles of raster and other coverage data contained in a GeoPackage.

While the Tile Metadata uses the same table structure as the comprehensive GeoPackage metadata, it does not contain a replica of the whole GeoPackage product metadata defined in section 7.3 Table-31. The fields listed in Table-38 are suggested in a tabular format (which would be represented in XML using DMF). The tileSet metadata use is optional and should be chosen as best suited to the contents of a particular tile Matrix Set layer or set of layers in the GeoPackage data product. Information about the tileSet layer that is contained within the GeoPackage (such as SRS) does not necessarily need to be in the metadata, though it can be repeated for clarity.

Table 38: gpkg_metadata.metadata Contents - Example TileSet Metadata

Field	Example
Abstract of the layer	Advanced Spaceborne Thermal Emission and Reflection Radiometer Version 2 Global Digital Elevation Model Color w/ Elevations, 1 arc-second Resolution
Keyword list	elevation, ASTER
CLASSIFICATION	UNCLASSIFIED
RELEASABILITY	REL NATO
Layer Update Date and Time	2019-05-15T01:43:45
SRS	EPSG:4326
Dimension name, units	elevation, meters
Dimension name, Extent	elevation, -600.0/9568.0/0
MAX scale	1:2000000
MIN scale	1:18000000
Legend	representation of the color legend

Requirement 31	/req/metadata/tile ***If a GeoPackage with raster tile data contains metadata, the GeoPackage shall use the table metadata to describe a tileSet or tile using the format defined in Table-34 and Table-35 with metadata the GeoPackage producer provides the contents of the tile data.***

9 Features

9.1 Feature Layer Metadata

This clause describes the storage of metadata as XML in GeoPackages and specifies the use of particular metadata values in certain conditions per the DMF and optionally national metadata such as NMF.

The metadata in this section describes the individual layers of Feature data in the SQLite database contained in the GeoPackage. As stated in section 7.3 Requirement /req/metadata/product-partial, "If particular tables in a GeoPackage constitute separate data products in Table-34, then the component elements that describe those data products SHALL be in the metadata XML as shown in Table-35 … ." This section gives an example of metadata that applies to Feature layers of vector data contained in a GeoPackage. Metadata for features and attributes of vector data can be defined using the same metadata construct.

While the Feature Metadata uses the same table structure as the comprehensive GeoPackage metadata, it does not contain a replica of the whole GeoPackage product metadata defined in section 7.3 Table-31. The fields listed in Table 39 are suggested in a tabular format (which would be represented in XML using DMF) based upon a fictitious example derived from the open-source Multi-national Geospatial Co-production Program (MGCP) World data. The specific Feature Type Metadata used is optional and should be chosen as best suited to the contents of a particular Feature layer or set of layers in the GeoPackage data product. Information about the Feature layer that is contained within the GeoPackage (such as CRS) does not necessarily need to be in the metadata, though it can be repeated for clarity.

Table 39: gpkg_metadata.metadata Contents - Example Feature Type Metadata

Field	Example
Abstract of the layer	Show a tract containing a concentration of buildings and/or other structures in Samoa, Kiribati, and Tonga.
Keyword list	geoscientificInformation
Category	Population
Data type	Vector polygon
CLASSIFICATION	UNCLASSIFIED
RELEASABILITY	REL NATO
Layer Update Date and Time	2015-03-25T01:43:45
Layer Temporal Extent (Date and Time range)	2015-03-25T00:00:00,2020-03-25T23:59:59
MAX scale intended use	1:10000
MIN scale intended use	1:100000

Field	Example
Portrayal standard	DGIWG 109, Portrayal Standard for MGCP

Requirement 32	/req/metadata/feature ***If a GeoPackage with Feature data contains metadata, the GeoPackage shall use the table metadata to describe a vector layer, feature, or feature attribute using the format in Table-34 and Table-35 with metadata the GeoPackage producer provides for the contents of the vector data.***

10 Gridded Coverages

This section contains requirements for DGIWG conformance of GeoPackage in capabilities unique to Gridded Data that are not applicable or contained in requirements for the common GeoPackage capabilities for tiles, metadata, and validity.

Coverages are digital geospatial information representing space / time-varying phenomena (from OGC WCS 2.1). The requirements in this section apply to using the “GeoPackage Extension for Tiled Gridded Coverage Data” [5], which defines how to encode and store tiled regular gridded data, such as a digital elevation model, in a GeoPackage. The tiles in a gridded coverage contain values, such as elevation, temperature or pressure, which are encoded in either PNG or TIFF format (from geopackage.org).

The base requirement for Gridded Coverages tests the Core requirements of the Gridded Coverage Extension [5] OGC 17-066r2 .

Requirement 33	/req/gridded/gridded-coverage-extension-core ***The DGIWG GeoPackage that includes the Gridded Data Extension SHALL comply with all the mandatory requirements in the Tiled Gridded Coverage Core Conformance Class in according to the normative reference [5] "GeoPackage Extension for Tiled Gridded Coverage Data", Annex A.1.***

10.1 Gridded Tiles in GeoPackage

The following requirements and recommendation apply to how the 2d gridded coverage and 3d gridded coverage data is represented within a DGIWG GeoPacakge.

Field Name and Quantity Definition

The gpkg_2d_gridded_coverage_ancilliary data table is encoded for specific types of gridded data. The Gridded data extension has a default field_name value of "Height" as stated in [5], OGC GeoPackage Gridded Extension section 7.1.1, Coverage Ancillary. DGIWG’s elevation gridded data requires the field_name value of "Height" in the gpkg_2d_gridded_coverage_ancilliary table as a measure of heights above or below a reference elevation, such as mean sea level, which corresponds to 3D gridded tiles of elevation or bathymetry. GeoPackage may contain other types of 2D and 3D gridded data, which is defined within the tables and metadata of the GeoPackage gridded data extension. The distinction between 2D and 3D gridded tiles is explained in opening paragraph of section 7.1.3, for which the implementatation is specified in this section.

To interpret the field_name column, the Gridded Data Extension of GeoPackage table named gpkg_2d_gridded_coverage_ancillary contains the column *quantity_definition*, which is "required

if the values contained in a grid coverage are anything other than height (elevation). This is a text string that describes the field (type)." ([5] OGC GeoPackage Gridded Extension section 7.1.1, Coverage Ancillary) The quantity_definition field is used in the DGIWG Profile to indicate whether the 2d_gridded_coverage contains elevation and bathymetry data, or some other type of gridded coverage. The default value for quantity_definition is also "Height". The other columns in gpkg_2d_gridded_coverage_ancillary table define more gridded coverage attributes, such as unit of measure (uom), which are necessary for the use of the gridded coverage.

Requirement 34	/req/gridded/field-name-elevation ***For 2d gridded coverage elevation (to include bathymetry) in DGIWG GeoPackage, for each tile set, if the word "elevation" or "height" (case insensitive) is included in the value of the quantity_definition column of the gpkg_2d_gridded_coverage_ancillary table, then in the same row the elevation gridded data SHALL have a field_name value of "Height" (case insensitive).***

Besides "Height", other types of gridded data should have a field_name value appropriate to the type of data in the coverage used consistently by the DGIWG producers and consumers of the gridded coverages. For example, gridded coverages with a distinct quantity_definition may include field_name values of Temperature, Air-pressure, Humidity, Soil-type, Slope, Population-density, Wave-height, and Salinity. To distinguish between elevation and other gridded_data in DGIWG GeoPackages, requirement /req/gridded/field-name-elevation applies to elevation / bathymetry and /recco/gridded-field-name-other applies other types of gridded coverages.

Recommendation 5	/recco/gridded-field-name-other ***For each gridded data set NOT containing elevation, the gridded data sets should use a common list of recommended field_name values and quantity_definition values for atmospheric, topographic, cultural, hydrographic, and other coverage data.***

Other requirements and guidance applicable to gridded coverages continue in the remainder of this section for DGIWG use.

Grid as center

Gridded data with field_name of Height shall use the default value of grid-value-is-center for grid-cell-encoding column.

Requirement 35	/req/gridded/center-value-elevation ***For 2d gridded coverage elevation (to include bathymetry) in DGIWG GeoPackage, for each tile set, if the word "elevation" or "height"(case insensitive) is included in the value of the quantity_definition column of the gpkg_2d_gridded_coverage_ancillary table, then in the same row the elevation gridded data SHALL use the grid_cell_encoding value of "grid-value-is-center".***

Zoom Factor

Requirement 36	/req/gridded/zoom-factor ***The gpkg_tile_matrix table for 2d_gridded_coverage SHALL contain pixel_x_size and pixel_y_size column values that differ by a factor of 2 between all adjacent zoom levels for each tile pyramid data table per OGC GeoPackage Clause 2.2.3. It SHALL NOT contain pixel sizes that vary by irregular intervals or by regular intervals other than a factor of 2 between adjacent zoom levels per OGC GeoPackage Clause 2.2.3.***

The tile matrix size, zoom and bbox requirements in section 8 listed below *do not* apply to gridded data:

- /req/tile/size-matrix
- /req/tile/size-data
- /req/zoom/matrix-sets-multiple
- /req/zoom/matrix-sets-one
- /req/bbox/crs

10.2 Gridded Metadata in GeoPackage

Gridded data in GeoPackage contains metadata consistent with these requirements in section 7.3:

- DMF Metadata /req/metadata/dmf

- Mininum metadata record /req/metadata/gpkg

In addition to required DMF, the 2d_gridded_coverage in GeoPackage may contain optional national metadata (e.g. NMF) as expressed in Requirement /req/metadata/row and *Table-32* and defined in this optional user-defined metadata requirement for 2d_gridded_coverage.

Requirement 37	/req/gridded/metadata-gridded-user ***If a GeoPackage contains metadata for gridded coverages, any gpkg_metadata table row with metadata values which apply to multiple GeoPackage table, row and/or column content items listed in Table-34 and having md_scope in Table-40 SHALL be linked to the specific row and/or column in that table by the md_file_id in gpkg_metadata_reference AND those content items by one or more gpkg_metadata_reference table rows with the content specified in Table-35.***

Gridded data may contain additional metadata for tiles and other values in gpkg_2d_gridded_coverage_ancillary table. The GeoPackage metadata scope defined in Table-31 in section 7.3 is intended for vector and raster tile data. Table 40 defines additional scope values applicable to Gridded data referenced by metadata for requirement /req/gridded/metadata-gridded-user. This optional metadata requirement covers gridded data tileSets, tiles, attribute types, and attributes. The attribute metadata can apply to any column in the gpkg_2d_gridded_coverage_ancillary table, such as field_name, datatype, precision, uom, or offset.

Table 40: GeoPackage gridded metadata scope values

gpkg_metadata. md_scope (scope code)	**gpkg_metadata_reference. reference_scope**
model (015) for a 2d_gridded_coverage tileSet	table
tile (016)	row/col
attributeType (002) 2d_gridded_coverage attribute type	column
attribute (001) 2d_gridded_coverage attribute value	row/col

These two metadata requirements in section 7.3 can optionally be applied to gridded data products and portions of products, if they exist in the GeoPackage.

- A gridded data product /req/metadata/product
- A portion of a gridded data product /req/metadata/product-partial

Annex A Abstract Test Suite for Conformance

A.1 Conformance Class OGC GeoPackage

Table A-1.1 ATS 1.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/geopackage/base
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the OGC Core Requirements in GPKG version 1.4 (http://www.geopackage.org, /base/core/<ogc-test-name>)
Requirements:	/req/geopackage/base : OGC Core Requirements in GPKG section 1.1
Conformance Class:	Base
Test type:	Capability
Test method:	Validate DGIWG GeoPackage against OGC tests for OGC Core Requirements using [4] GPKG Annex A.1.1.

Table A-1.2 ATS 1.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/geopackage/options
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the OGC Optional Requirements in GPKG version 1.4 (http://www.geopackage.org, /opt/features/<ogc-test-name>)
Requirements:	/req/geopackage/options : OGC Options Requirements in GPKG sections 2.1-2.4
Conformance Class:	Options
Test type:	Capability
Test method:	Validate DGIWG GeoPackage against OGC tests for OGC Optional Requirements using [4] GPKG Annex A.2.1 - A.2.4 for DGIWG mandatory, conditional, and optional GPKG extensions according to Table-8 in this GeoPackage.

A.2 Conformance Class Extensions

Table A-2.1 ATS 2.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/mandatory
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Extension Mandated GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/mandatory)
Requirements:	/req/extensions/mandatory
Conformance Class:	extensions
Test type:	Capability
Test method:	Validate DGIWG GeoPackage contains mandatory extensions for http://www.dgiwg.org/std/gpkg/1.0/conf/extension/mandatory.

```
ATS returns Pass or Fail compliance for each extension for Features and for
Tiles.

If no extensions are mandatory in the GeoPackage profile, this ATS is null.

For each extension name _e_ in Table 8,
  If Table 8 Features column value for _e_ = "M" and any row of
gpkg_contents.data_type = "features", then
  Search rows of table gpkg_extension for _e_:
  If gpkg_extensions.extension_name contains _e_, then Pass for _e_ Vector;
End For
If no extension matched _e_, Fail for extension name _e_ Vector.

For each extension name _e_ in Table 8,
  If Table 8 Tiles column value for _e_ = "M" and any row of
gpkg_contents.data_type = "tiles", then
  Search rows of table gpkg_extension for _e_:
  If gpkg_extensions.extension_name contains _e_, then Pass for _e_ Tiles;
End For
If no extension matched _e_, Fail for extension name _e_ Tiles.
```

Table A-2.2 ATS 2.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/optional
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Extension Optional GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/optional)
Requirements:	/req/extensions/optional

Conformance Class:	extensions
Test type:	Capability
Test method:	Validate extensions in DGIWG GeoPackage are optional for http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/optional when the extension is applicable to the GeoPackage content. If no extensions are optional in the profile, this ATS is null.

```
ATS returns Pass or Fail compliance for each extension for Features and for
Tiles.
Returns NULL if no extensions are optional.

For each extension name _e_ in Table 8,
  If Table 8 Features value for _e_ = "O" and any row of
gpkg_contents.data_type = "features", Then
    Pass for extension name _e_ Features.
  End If
End For

For each extension name _e_ in Table 8,
  If Table 8 Tiles value for _e_ = "O" and any row of gpkg_contents.data_type
= "tiles", Then
    Pass for extension name _e_ Tiles.
  End If
End For

Note - If the extension is NA and present in GeoPackage, then the test fails
because this extension is not optional.

For each extension name _e_ in Table 8,
  If Table 8 Features value for _e_ = "NA" and any row of
gpkg_contents.data_type = "features", Then
    Search rows of table gpkg_extension for _e_ with a features extension.
    For each extension _e2_ in gpkg_extension
      If gpkg_extensions(_e2_).extension_name contains _e_,
      Then
        Fail if extension applies to features and features table exists in
geopackage.
        If gpkg_extensions(_e2_).table_name = NULL OR
gpkg_extensions(_e2_).table_name matches any gpkg_contents.table_name where
gpkg_contents.data_type = "features"
        Then Fail for _e_ Features;
      End If
```

```
    End For
  End If

  If Table 8 Tiles value for _e_ = "NA" and any row of
gpkg_contents.data_type = "tiles", Then
    Search rows of table gpkg_extension for _e_ with a tiles extension
    For each extension _e2_ in gpkg_extension
      If gpkg_extensions(_e2_).extension_name contains _e_,
      Then
        Determine if extension applies to tiles and tiles for this extension
exist in geopackage.
        If gpkg_extensions(_e2_).table_name = NULL OR
gpkg_extensions(_e2_).table_name matches any gpkg_contents.table_name where
gpkg_contents.data_type = "tiles"
        Then Fail for extension name _e_ Tiles;
        End If
    End For
  End If
End For
```

Table A-2.3 ATS 2.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/not-allowed
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Extension Not Allowed GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/not-allowed)
Requirements:	/req/extensions/not-allowed
Conformance Class:	extensions
Test type:	Capability
Test method:	Validate that no extensions in DGIWG GeoPackage that are not allowed are implemented according to http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/not-allowed. If no extensions are disallowed in the profile, this ATS is null.

```
ATS returns Pass or Fail compliance for each extension for Features and for
Tiles.

For each extension name _e_ in Table 8
  Check features extensions
  If Table 8 Features value for _e_ = "N" and any row of
gpkg_contents.data_type = "features"
  Then
    Search rows of table gpkg_extension for _e_:
    If gpkg_extensions.extension_name contains _e_, then Fail for extension
name _e_ Features;
  Else
    If no extension matched _e_, then Pass for extension name _e_ Features.
  End If

  Check tiles extensions
  If Table 8 Tiles value for _e_ = "N" and any row of gpkg_contents.data_type
= "tiles"
  Then
    If gpkg_extensions.extension_name contains _e_, then Fail for extension
name _e_ Tiles.
  Else
    If no extension matched _e_, then Pass for extension name _e_ Tiles.
  End If

End for
```

Table A-2.4 ATS 2.4

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/conditional
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Extension Conditional GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/conditional)
Requirements:	/req/extensions/conditional
Conformance Class:	extensions
Test type:	Capability
Test method:	Validate extensions in DGIWG GeoPackage exist that meet conditional criteria for http://www.dgiwg.org/std/gpkg/1.0/conf/extensions/conditional. If no conditional extensions are required in the profile, this ATS is null.

```
ATS returns Pass or Fail compliance for each conditional extension for
Features and for Tiles; null if no extensions are conditional.

For each extension name _e_ in Table 8,
if Table 8 Features value for _e_ = "C" and any row of
gpkg_contents.data_type = "features", then
If gpkg_extensions.extension_name contains _e_, then Pass for extension name
_e_ Features; Else Null result.

For each extension name _e_ in Table 8,
if Table 8 Tiles value for _e_ = "C" and any row of gpkg_contents.data_type =
"tiles", then
If gpkg_extensions.extension_name contains _e_, then Pass for extension name
_e_ Tiles; Else Null result.
```

A.3 Conformance Class Coordinate Reference Systems

Table A-3.1 ATS 3.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/raster-allowed
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the CRS GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/raster-allowed)
Requirements:	/req/crs/raster-allowed
Conformance Class:	crs
Test type:	Capability
Test method:	Validate the Coordinate Reference Systems (CRS) in each Tile Matrix Set used in DGIWG GeoPackage complies with http://www.dgiwg.org/std/gpkg/1.0/conf/crs/raster-allowed.

```
ATS returns Pass or Fail compliance for each tile matrix set.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  If the srs_id of gpkg_tile_matrix_set for _t_ equals the corresponding
gpkg_spatial_ref_sys.srs_id,
  Set the default tile-matrix-set-CRS(_t_) to Pass.
  Then for this _srs_id_
    If the gpkg_spatial_ref_sys.srs_name of _srs_ does not match one of the
CRS in Table 11 or Table 12 that corresponds to the srs_name defined in
section 7.2 tables 15 through 21
    Then Fail for tile-matrix-set-CRS(_t_)
    End If
  End If
End For
```

Table A-3.2 ATS 3.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/raster-tile-matrix-set
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the CRS GPKG for raster data (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/raster-tile-matrix-set)
Requirements:	/req/crs/raster-tile-matrix-set
Conformance Class:	crs
Test type:	Capability
Test method:	Validate each tile matrix set contained in DGIWG GeoPackage follow the implementation for the respective CRS defined in [6] OGC 17-083r2 conformance class http://www.opengis.net/spec/tilematrixset/1.0/conf/crs-raster-tile-matrix-set.

```
ATS returns Pass or Fail compliance for each tile matrix set.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles"
  If the srs_id of gpkg_tile_matrix_set _t_ equals the corresponding
gpkg_spatial_ref_sys.srs_id
  Then
    Set the default tile-matrix-set-conform(_t_) to Null.
    For this _srs_id_,
    If the gpkg_spatial_ref_sys.srs_name of _srs_ matches a srs_name in Table
15 through Table 21
    Then
      If the gpkg_tile_matrix_set of _t_ conforms to the XML or JSON_LD
encoding as verified by ATS A.1 - A.16 of [6] OGC 2D TMS 17-083r2
        AND the gpkg_tile_matrix_set for *each* zoom_level of _t_ matches the
well known scale set of the corresponding TileMatrixSet.ScaleDenominator of
the CRS with the same zoom level in Annex C.1, C.2, C.3, C.4, D.1, D.2, D.3,
or D.4 of [6] OGC 2D TMS 17-083r2
      Then Pass for tile-matrix-set-conform(_t_)
      Else Fail for tile-matrix-set-conform(_t_)
      End If
    Else
      Remains Null (This ATS does not apply)
    End If
  End If
End For
```

Table A-3.3 ATS 3.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/2d-vector
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the CRS GPKG for 2D vector data (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/2d-vector)
Requirements:	/req/crs/2d-vector
Conformance Class:	crs
Test type:	Capability
Test method:	Validate the 2D vector CRS used in each feature set in DGIWG GeoPackage complies with http://www.dgiwg.org/std/gpkg/1.0/conf/crs/2d-vector.

```
ATS returns Pass or Fail compliance for each 2D feature set; NULL if no
features are in GeoPackage.

For each feature set _f_ in gpkg_contents where the gpkg_contents.data_type
of _f_ = "features"
  Set the default Geometry-Columns-2D-CRS(_f_) to Pass.
  If the srs_id of gpkg_geometry_columns for _f_ equals the corresponding
gpkg_spatial_ref_sys.srs_id AND gpkg_geometry_column.z = 0
  Then
    For this _srs_id_,
    If the gpkg_spatial_ref_sys.srs_name of _srs_id_ does not match one of
the CRS with "CRS Dim" = 2  in Table 13
    Then Set Geometry-Columns-2D-CRS(_f_) to Fail for feature set _f_
    End If
  End If

  Set the default Geometry-Binary-CRS(_f_) to Pass.

  If the empty geometry flag for _f_ is not set AND the srs_id of
GeoPackageBinaryHeader for _f_ equals the corresponding
gpkg_spatial_ref_sys.srs_id
  Then
    For this _srs_id_,
    If the gpkg_spatial_ref_sys.srs_name of _srs_id_ does not match one of
the CRS in Table 13 that corresponds to the srs_name defined in section 7.2
tables 15 through 21
    Then Set Geometry-Binary-CRS(_f_) to Fail for feature set _f
    End If
  End If

  If Geometry-Binary-CRS(_f_) AND Geometry-Columns-2D-CRS(_f_) are both Pass
  Then Pass for feature set _f_
  Else Fail for feature set _f_
  End If
```

Table A-3.4 ATS 3.4

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/3d-vector
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the CRS GPKG for 3D vector data (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/3d-vector)
Requirements:	/req/crs/3d-vector

Conformance Class:	crs
Test type:	Capability
Test method:	Validate the 3D vector CRS used in DGIWG GeoPackage comply with http://www.dgiwg.org/std/gpkg/1.0/conf/crs/3d-vector.

```
ATS returns Pass or Fail compliance for each 3D feature set; NULL if no
features are in GeoPackage.

For each feature set _f_ in gpkg_contents where the gpkg_contents.data_type
of _f_ = "features"
  Set the default Geometry-Columns-3D-CRS(_f_) to Pass.
  If the srs_id of gpkg_geometry_columns for _f_ equals the corresponding
gpkg_spatial_ref_sys.srs_i AND gpkg_geometry_column.z != 0
  Then
    For this _srs_id_,
    If the gpkg_spatial_ref_sys.srs_name of _srs_id_ does not match one of
the CRS with "CRS Dim" = 3 or "CRS DIM" = 1 in Table 13
    Then Set Geometry-Columns-3D-CRS(_f_) to Fail for feature set _f_
    End If
  End If

  Set the default Geometry-Binary-CRS(_f_) to Pass.

  If the empty geometry flag for _f_ is not set AND the srs_id of
GeoPackageBinaryHeader for _f_ equals the corresponding
gpkg_spatial_ref_sys.srs_id
  Then
    For this _srs_id_,
    If the gpkg_spatial_ref_sys.srs_name of _srs_id_ does not match one of
the CRS in Table 13
    Then Set Geometry-Binary-CRS(_f) to Fail for feature set _f_
    End If
  End If

  If Geometry-Binary-CRS(_f_) AND Geometry-Columns-3D-CRS(_f_) are Pass,
  Then Pass for feature set _f_
  Else Fail for feature set _f_
  End If
End For
```

Table A-3.5 ATS 3.5

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/wkt
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the CRS / SRS WKT GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/wkt)
Requirements:	/req/crs/wkt
Conformance Class:	crs
Test type:	Capability
Test method:	Validate the CRS / SRS definition used in DGIWG GeoPackage contains WKT that is compliant with optional and mandatory elements appropriate for the CRS / SRS as defined by WKT1 [8] to comply with DGIWG GeoPackage http://www.dgiwg.org/std/gpkg/1.0/conf/crs/wkt.

```
ATS returns Pass or Fail compliance for each CRS in gpkg_spatial_ref_sys
The WKT in definition_12_063 can be WKT1 or WKT2.

For each crs _srs_id_ in gpkg_spatial_ref_sys
  Set default WKT-CRS for _srs_id_ to Pass.
  _wkt_def_ = gpkg_spatial_ref_sys.definition(srs_id)

  If _wkt_def_ does not contain "COMPOUNDCRS"
    Validate the _wkt_def_ is consistent with the WKT definition in Table 15
through Table 24 for the corresponding _srs_id_
    If _wkt_def_ is NOT valid for _srs_id_
    Then Set WKT-CRS for _srs_id_ to Fail
    Else
      _wkt_def_ = gpkg_spatial_ref_sys.definition_12_063(_srs_id_)
      If _wkt_def_ is not NULL
      Then
        Validate the _wkt_def_ is consistent with the WKT definition in Table
15 through Table 24 for the corresponding _srs_id_
        If _wkt_def_ is NOT valid for _srs_id_
        Then Set WKT-CRS for _srs_id_ to Fail
        End If
      End If
    End If
  End If
End For
```

Table A-3.6 ATS 3.6

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/compound
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Compound CRS GPKG Z (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/compound)
Requirements:	/req/crs/compound
Conformance Class:	crs
Test type:	Capability
Test method:	Validate every compound CRS used in DGIWG GeoPackage contains Z geometry values to comply with DGIWG GeoPackage http://www.dgiwg.org/std/gpkg/1.0/conf/crs/compound. If Z doesn't exist (2D features), then the CRS cannot have a COMPOUNDCRS; if Z exists, COMPOUNDCRS is optional (e.g. could use EPSG:4979 instead of compound CRS).

```
ATS returns Pass or Fail compliance for each feature set

For each feature set _f_ in gpkg_contents where the gpkg_contents.data_type
of _f_ = "features"
  Set the default Geometry-Columns-Compound-CRS(_f_) to Pass.
  If the srs_id of gpkg_geometry_columns for _f_ equals the corresponding
gpkg_spatial_ref_sys.srs_id
  Then
    If gpkg_geometry_columns(_srs_id_).z = 0 (Z prohibited)
    Then
      For this _srs_id_,
      If the gpkg_spatial_ref_sys.definition(_srs_id_) or
definition_12_063(_srs_id_) for _srs_id_ contains "COMPOUNDCRS"
      Then set Geometry-Columns-Compound-CRS(_f_) to Fail for feature set
_f_.
    End If
  End If
End For.
```

Table A-3.7 ATS 3.7

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/crs/compound-wkt
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Compound CRS WKT GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/crs/compound-wkt)
Requirements:	/req/crs/compound-wkt
Conformance Class:	crs
Test type:	Capability
Test method:	Validate every compound CRS used in DGIWG GeoPackage contains WKT compliant according to Table-30 with optional and mandatory elements appropriate for the CRS / SRS as defined by WKT1 [8] to comply with DGIWG GeoPackage http://www.dgiwg.org/std/gpkg/1.0/conf/crs-compound-wkt.

```
ATS returns Pass or Fail compliance for each CRS in gpkg_spatial_ref_sys

For each crs _srs_id_ in gpkg_spatial_ref_sys
  Set default COMPOUND-WKT-CRS for _srs_id_ to Pass.
  _wkt_def_ = gpkg_spatial_ref_sys.definition(srs_id)
  If _wkt_def_ contains "COMPOUNDCRS"
  Then
    Validate the _wkt_def_ contains the compound fields consistent with the
WKT definition for COMPOUNDCRS in Table 29 and Table 30
    If _wkt_def_ is NOT valid for _srs_id_
    Then Set COMPOUND-WKT-CRS for _srs_id_ to Fail
    Else
      _wkt_def_ = gpkg_spatial_ref_sys.definition_12_063(srs_id)
      If _wkt_def_ is not NULL
      Then
        If _wkt_def_ contains "COMPOUNDCRS"
        Then
          Validate the _wkt_def_ contains the compound fields consistent with
the WKT definition for COMPOUNDCRS in Table 29 and Table 30
          If _wkt_def_ is NOT valid for _srs_id_
          Then Set COMPOUND-WKT-CRS for _srs_id_ to Fail
          End If
        End If
      End If
    End If
  End If
End For.
```

A.4 Conformance Class Metadata

Table A-4.1 ATS 4.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/dmf
Test purpose:	Verify that a DGIWG GeoPackage is conformant to contain metadata for the whole GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-dmf)
Requirements:	/req/metadata/dmf
Conformance Class:	metadata
Test type:	Capability
Test method:	Validate at least one whole data product metadata contains metadata in gpkg_metadata.metadata for each row of Table-31 that has a gpkg_metadata.md_scope of "dataset" and the required gpkg_metadata.metadata exists for the complete DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-dmf.

AND contains DMF for any other part of the GeoPackage.

```
ATS produces a validation result for the entire gpkg_metadata.

Set default Metadata-whole-dmf to Fail.
Set default Metadata-part-dmf to Fail.
partial-metadata = false

If ATS 1.2 for GeoPackage Metadata Extension F.8 Pass
Then continue to perform more specific validation of the gpkg_metadata
  For each row in gpkg_metadata with id _m_
    If gpkg_metadata.md_scope(_m_) is equal to "series" or "dataset"
      If gpkg_metadata(_m_).md_standard_id is a valid DMF URL
      Then
        Validate gpkg_metadata(_m_).metadata XML content with DMF ATS.
        dmf_valid = result of DMF ATS for a DMF Metadata Set [1] DGIWG
Metadata Foundation ATS A.4.

        If  dmf_valid is true
        Then Metadata-whole-dmf = Pass.
        End If
      End If
    Else
      Check if a part of the GeoPackage also contains DMF metadata
      If another metadata set exists, at least one must be in DMF.
```

```
      partial-metadata = true
      If gpkg_metadata(_m_).md_standard_id is a valid DMF URL
      Then
        Validate gpkg_metadata(_m_).metadata XML content with DMF ATS.
        dmf_valid = result of DMF ATS for a DMF Metadata Set [1] DGIWG
Metadata Foundation ATS A.4.

        If dmf_valid is true
        Then Metadata-part-dmf = Pass.
        End If

      End If
    End If
  End For
End If

If any part of the GeoPackage contains metadata, then both whole and part
must be valid DMF.

If partial_metadata is True
Then
  Return Metadata-whole-dmf AND Metadata-part-dmf
Else
  Return Metadata-whole-dmf
End If
```

Table A-4.2 ATS 4.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/gpkg
Test purpose:	Verify that a DGIWG GeoPackage is conformant to contain metadata for the whole GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-gpkg)
Requirements:	/req/metadata/gpkg
Conformance Class:	metadata
Test type:	Capability
Test method:	Validate at least one whole data product metadata contains metadata in gpkg_metadata.metadata for each row of Table-31 that has a gpkg_metadata.md_scope of "dataset" and the required gpkg_metadata.metadata exists for the complete DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-gpkg.

```
ATS produces a validation result for the entire gpkg_metadata.

Set default Metadata-whole to Fail.

  For each row in gpkg_metadata with id _m_
    If gpkg_metadata.md_scope(_m_) is equal to "series" or "dataset"
    Then
      If gpkg_metadata.metadata(_m_) is not NULL AND
        gpkg_metadata(_m_).md_standard_id is not NULL and
        gpkg_metadata(_m_).mime_type = "text/xml"
      Then
        Metadata-whole = Pass.
      End If
    End If
  End For

Return Metadata-whole
```

Table A-4.3 ATS 4.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/row
Test purpose:	Verify that a DGIWG GeoPackage metadata_reference table row is conformant to the metadata table structure for the whole GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-row)
Requirements:	/req/metadata/row
Conformance Class:	metadata
Test type:	Capability
Test method:	Validate each row of metadata in Table-31 have at least one corresponding row in metadata_reference in Table-33 for the whole GPKG in DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-row.

```
ATS produces a validation result for each row in gpkg_metadata for a DGIWG
GeoPackage, including user defined metadata and its association to the
metadata_reference table.

For each row in gpkg_metadata with id _m_
  Set default gpkg-reference-row(_m_) to Pass
 If gpkg_metadata(_m_).md_scope = "dataset" (geopackage) or "series"
(product)
 Then
  Select table_name, column_name, row_id_value, timestamp, md_parent_id from
gpkg_metadata_reference
  where gpkg_metadata_reference.md_file_id = _m_ AND
gpkg_metadata_reference.reference_scope = "table" OR "geopackage"

  Ensure all user defined values in a gpkg_metadata are linked to the
associated metadata_reference
  If result set is empty
  Then
    Set gpkg-reference-row(_m_) to Fail
  Else
    For each result in result set, verify expected values
      If table_name != NULL OR row_id_value != NULL OR time_stamp is invalid
timestamp
      Then
        Set gpkg-reference-row(_m_) to Fail
      End If
    End For
  Else
    Note: metadata for other md_scope values are not tested in this
requirement.
    Set gpkg-reference-row(_m_) to NULL
  End If
End For
```

Table A-4.4 ATS 4.4

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/user
Test purpose:	Verify that a DGIWG GeoPackage with user defined metadata associates a row in a partial set of GeoPackage content in the metadata_reference table to have a metadata table in a GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-user)
Requirements:	/req/metadata/user

Conformance Class:	metadata
Test type:	Capability
Test method:	If optional partial metadata exists in the gpkg_metadata, validate metadata associated with each user-defined row of partial tile and feature metadata as defined in Table-34 for gpkg_metadata.md_scope of "featureType", "feature", "model" (tileSet). and "tile" to have a gpkg_metadata_reference.reference_scope as defined in Table-35 that adhere to the md_scope and reference_scope defined in Table-36 for the DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata-user. Note: this ATS applies to tiles and features; 2d_gridded_coverage metadata tested in ATS A-10.5.

```
ATS produces a validation result for each tile_matrix_set, tile_matrix,
geometry_columns and feature_geometry in gpkg_contents; Pass if metadata
tables are valid, Fail if metadata tables are invalid, NULL if no metadata
exists (it is optional)

** TILES **

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles"
  Set default Metadata-user-tile-set(_t_) to Null

  For each gpkg_metadata_reference where gpkg_metadata_referenc.table_name =
tile_matrix_set.table_name _t_  and gpkg_metadata_reference.row_id_value is
NULL

    This tile_matrix_set has 1 or more instances of metadata, search all tile
metadata_reference instances to have correct reference_scope and
corresponding gpkg_metadata.md_scope of "model"

    If the gpkg_metadata_reference.reference_scope = "table" for table_name
_t_ and row_id = NULL
    Then
      Does the gpkg_metadata exist for the gpkg_metadata_reference describing
this tile_matrix_set ?

      If exists gpkg_metadata_reference for table_name = _t_ and md_file_id
equal to 1 or more rows equal to gpkg_metadata.id where
gpkg_metadata.md_scope = "model"
      Then Set Metadata-user-tile-set(_t_) to Pass
```

```
      Else Set Metadata-user-tile-set(_t_) to Fail  (no metadata exists)
      End If
    Else
      Set Metadata-user-tile-set(_t_) to Fail (wrong reference scope for
tile_matrix_set)
    End If
  End For

  Next Search the tile_matrix pyramid tiles for tile metadata

  For each gpkg_metadata_reference where gpkg_metadata_reference.table_name =
tile_matrix.table_name _t_ and gpkg_metadata_reference.row_id value is not
NULL

    Set all default Metadata-user-tile(_t_,1 to max_zoom) to Null

    This tile has metadata, search all tile metadata_reference instances to
have correct reference_scope and corresponding gpkg_metadata.md_scope of
"tile" and the matching md_file_id value

    For each pyramid tile gpkg_tile_matrix _z_ where
gpkg_tile_matrix.zoom_level = metadata_reference.row-id AND
gpkg_metadata_reference.table_name = _t_

      If the gpkg_metadata_reference.reference_scope  = "row/col" for
table_name _t_ and row_id = _z_
      Then
        Does the gpkg_metadata exist for the gpkg_metadata_reference
describing this tile ?
        If exists gpkg_metadata_reference for _t_ and _z_ with md_file_id
equal to 1 or more rows equal to gpkg_metadata.id where
gpkg_metadata.md_scope = "tile"
        Then Set Metadata-user-tile(_t_,_z_) to Pass
        Else Set Metadata-user-tile(_t_,_z_) to Fail  (no metadata exists for
tile)
        End If
      Else
        Set Metadata-user-tile(_t_,_z_) to Fail (wrong reference scope for
tile)
      End If
    End For
  End For
End For

** FEATURES **
```

```
For each feature set _f_ in gpkg_contents where the gpkg_contents.data_type
of _f_ = "features"
  Set default Metadata-user-feature-set(_f_) to Null

  For each feature_geometry _f_ in gpkg_geometry_columns where
gpkg_geometry_columns.table_name = gpkg_metadata_reference.table_name and
gpkg_metadata_reference.row_id_value is NULL

    For each gpkg_metadata_reference where gpkg_metadata_referenc.table_name
= gpkg_geometry_columns.table_name _t_  and
gpkg_metadata_reference.row_id_value is NULL

      Search all feature sets with metadata to have correct reference_scope
and gpkg_metadata

      If gpkg_metadata_reference.reference_scope = "table" for table_name =
_f_ and row_id = NULL
      Then
        If exists gpkg_metadata_reference for table _f_ with md_file_id equal
to 1 or more rows equal to gpkg_metadata.id where gpkg_metadata.md_scope =
"featureType"
        Then Set Metadata-user-feature-set(_f_) to Pass
        Else Set Metadata-user-feature-set(_f_) to Fail  (no metadata exists)
        End If
      Else
        Set Metadata-user-feature-set(_f_) to Fail (wrong reference_scope)
      End If
    End For
  End For

  Next Search the feature_geometry for feature metadata

  For each gpkg_metadata_reference where gpkg_metadata_reference.table_name =
geometry_columns.table_name _f_ and gpkg_metadata_reference.row_id value is
not NULL

    Set all default Metadata-user-features(_f_,1 to max) to Null

    This tile has metadata, search all tile metadata_reference instances to
have correct reference_scope and corresponding gpkg_metadata.md_scope of
"tile" and the matching md_file_id value

    For each feature id _fid_ in geometry_columns.table_name _f_ where
geometry_columns.table_name.id = metadata_reference.row-id AND
```

```
gpkg_metadata_reference.table_name = _f_

      If the gpkg_metadata_reference.reference_scope = "row"  and row_id =
_fid_
      Then
        Does the gpkg_metadata exist for the gpkg_metadata_reference
describing this feature ?
        If exists gpkg_metadata_reference for _f_ and _fid_ with md_file_id
equal to 1 or more rows equal to gpkg_metadata.id where
gpkg_metadata.md_scope = "feature"
        Then Set Metadata-user-tile(_f_,_fid_) to Pass
        Else Set Metadata-user-tile(_f_,_fid_) to Fail  (no metadata exists
for feature)
        End If
      Else
        Set Metadata-user-tile(_f_,_z_) to Fail (wrong reference scope for
feature)
      End If
    End For
  End For
End For
```

Table A-4.5 ATS 4.5

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/product
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Metadata Product GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/product)
Requirements:	/req/metadata/product
Conformance Class:	metadata
Test type:	Capability
Test method:	If a gpkg-metadata contains metadata for a series of products, validate gpkg product metadata contains metadata in gpkg_metadata.metadata with each row of Table-31 that has a gpkg_metadata.md_scope of "series", then the required gpkg_metadata.metadata exists for the gpkg product for the DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/product.

```
ATS produces a validation result for the entire gpkg_metadata if product
metadata is present and DMF compliant; NULL if no product metadata present.,

Set default Metadata-product-dmf to NULL.
```

```
Set default Metadata-product-other to NULL.

  For each row in gpkg_metadata with id _m_
    If gpkg_metadata.md_scope(_m_) is equal to "series"
      If gpkg_metadata(_m_).md_standard_id is a valid DMF URL
      Then
        Validate gpkg_metadata(_m_).metadata XML content with DMF ATS.
        dmf_valid = result of DMF ATS for a DMF Metadata Set [1] DGIWG
Metadata Foundation ATS A.4.

        If  dmf_valid is true
        Then Set Metadata-product-dmf to Pass.
        Else
          Set Metadata-product-dmf to Fail.
        End If
      Else
        Check if national metadata is populated for product.
        If gpkg_metadata.metadata(_m_) is not NULL AND
          gpkg_metadata(_m_).md_standard_id is not NULL and
          gpkg_metadata(_m_).mime_type = "text/xml"
        Then
          Set Metadata-whole-other to Pass.
        Else
          Set Metadata-whole-other to Fail.
        End If
      End If
    End If
  End For
End If

If any part of the GeoPackage contains metadata, then both whole and part
must be valid DMF.

If Metadata-product-dmf is not NULL
Then
  If Metadata-whole-other is not null
  Then
    Return Metadata-product-dmf AND Metadata-product-other
  Else
    Return Metadata-product-dmf.
Else
  Return NULL
End If
```

Table A-4.6 ATS 4.6

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/product-partial
Test purpose:	Verify that a DGIWG GeoPackage with a partial set of metadata is conformant to the Metadata Partial GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/partial)
Requirements:	/req/metadata/product-partial
Conformance Class:	metadata
Test type:	Capability
Test method:	If optional metadata for a product (md_scope of 'series') in Table-34 exists in gpkg_metadata.metadata, verify that each row of metadata associated with the gpkg_metatdata_reference.md_parent_id in Table-35 has gpkg_metadata_reference.reference_scope value of "table" OR "row/col" OR "row" OR "column" for a DGIWG GeoPackage to be compliant http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/product-partial. Note: This ATS is applicable for partial product metadata about gpkg_contents with a data_type = tiles, features and 2d_gridded_coverage.

```
ATS produces a validation result for each row of the gpkg_metadata_reference.
Valid if partial metadata for a product is present; invalid if non-partial
metadata for a product exists; NULL if the gpkg_metadata_reference is not for
a product,

For each _r_ in gpkg_metatdata_reference
  Set default Metadata-product-partial(_r_) to NULL.
  parent_id = gpkg_metatdata_reference(_r_).md_parent_id
  If parent_id is not NULL
  Then
    This metadata reference is part of a hierarchy of metadata.

    md_file_id = gpkg_metatdata_reference(_r_).md_file_id
    If md_file_id != NULL
    Then
      If md_file_id != parent_id
      Then
        Since this gpkg_metadata_reference row is not the root of a metadata
hierarchy, determine if the metadata row of the parent_id is a product.

        If gpkg_metadata(parent_id).md_scope = "series"
        Then
          If the parent of the metadata is a series (product), then this
metadata record shall be a dataset or partial metadata content to be valid.

          If gpkg_metatdata_reference(_r_).reference_scope = "table" OR
"row/col" OR "row" OR "column"
          Then
            Set Metadata-product-partial(_r_) to Pass.
          Else
            Set Metadata-product-partial(_r_) to Fail.
          End If
        Else
          The md_parent_id cannot be equal to the md_file_id (See Annex F.8
Req 102), so this reference row fails.
          Set Metadata-product-partial(_r_) to Fail.
        End If
      End If
    Else (md_file_id is NULL, which is invalid)
      Set Metadata-product-partial(_r_) to Fail.
    End If
  End If
End For
```

Table A-4.7 ATS 4.7

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/tile
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Tile Metadata GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/tile)
Requirements:	/req/metadata/tile
Conformance Class:	metadata
Test type:	Capability
Test method:	Validate the tile layer metadata in DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/tile.

```
ATS produces a validation result for each row of tile matrix set or tile
metadata in the gpkg_metadata table; null for those metadata that are not
tile matrix sets or tiles.

For each row in gpkg_metadata with id _m_
  Set default Metadata-tile(_m_) to NULL.
  partial-tile-metadata = false
  If gpkg_metadata.scope(_m_) is equal to "model" or "tile"
  Then
    partial-tile-metadata = true
    If gpkg_metadata(_m_).md_standard_id is a valid URL AND
gpkg_metadata(_m_).metadata is XML content AND gpkg_metadata(_m_).mime_type =
"text/xml"
    Then
      Check the gpkg_metadata_reference table row exists for this
gpkg_metadata row.

      For each reference_scope, table_name, column_name, row_id from
gpkg_metadata_reference.md_file_id = _m_

        If results are not empty result set
        Then
          To validate requirement for raster tiles, distinguish between
data_type = "tiles" and "features" or data_type = "2d_gridded_data"

          For each dataset _t_ in gpkg_contents

            If gpkg_metadata_reference.table_name =
gpkg_contents(_t_).table_name
            Then
```

```
              If gpkg_contents(_t_).data_type = "tiles"
              Then
               If reference_scope = "table" (for md_scope ="model")
               Then
                  If table_name exists in gpkg_tile_matrix_set
                  Then
                    Set Metadata-tile(_m_) to Pass
                  Else
                    Set Metadata-tile(_m_) to Fail (invalid no tile matrix
table for tiles)
                  End If
               Else
                If reference_scope = "row/col" (for md_scope of "tile")
                  If table_name exists in gpkg_tile_matrix
                  Then
                    Set Metadata-tile(_m_) to Pass
                  Else
                    Set Metadata-tile(_m_) to Fail (invalid reference_scope
table for tiles)
                  End If
                Else
                  Set Metadata-tile(_m_) to Fail (invalid reference_scope for
tiles)
                End If
              Else
                Set Metadata-tile(_m_) to Null (does not apply, contents are
not tiles)
              End If
             Else
              Set Metadata-feature(_m_) to Fail (invalid table name for tile
metadata_reference)
             End If
           End For
        Else
          (no applicable metadata in gpkg_metadata_reference)
        End If
      End For
    End If
  End If

  If partial-tile-metadata is True
  Then
    Verify the "attributeType" or "attribute" in gpkg_metadata_reference.

    Repeat the results of the GPKG Options ATS for Requirement 119 and the
```

```
ATS for the Metadata Extension Annex F.8 Requirements 96-99. These ATS
validate the column_name and row_id of tile attributes in
gpkg_metadata_reference correspond to the tables. columns and row values in
gpkg_contents, gpkg_tile_matrix_set, gpkg_tile_matrix and pyramid tiles for
the tile_matrix_set.
  End If
End For
```

Table A-4.8 ATS 4.8

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/feature
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Vector Metadata GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/feature)
Requirements:	/req/metadata/feature
Conformance Class:	metadata
Test type:	Capability
Test method:	Validate Vector (feature) layer metadata in DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/metadata/feature.

```
ATS produces a validation result for each row of feature metadata in the
gpkg_metadata table; null for those metadata that are not feature data sets
or features.

For each row in gpkg_metadata with id _m_
  Set default Metadata-feature(_m_) to NULL.
  partial-feature-metadata = false
  If gpkg_metadata.md_scope(_m_) is equal to "featureType" or "feature" or
"attributeType" or "attribute"
  Then
    partial-feature-metadata = true
    If gpkg_metadata(_m_).md_standard_id is a valid URL AND
gpkg_metadata(_m_).metadata is XML content AND gpkg_metadata(_m_).mime_type =
"text/xml"
    Then
      Check the gpkg_metadata_reference table row exists for this
gpkg_metadata row.

      For each reference_scope, table_name, column_name, row_id from
gpkg_metadata_reference.md_file_id = _m_

        If results are not empty result set
```

```
        Then
          To validate requirement for features, distinguish between data_type
= "features" and "tiles" or "2d_gridded_data"

          For each dataset _f_ in gpkg_contents

            If gpkg_metadata_reference.table_name =
gpkg_contents(_f_).table_name
            Then
              If gpkg_contents(_f_).data_type = "features"
              Then
                If reference_scope = "table" or "row" (for md_scope
="featureType")
                Then
                  If table_name exists in gpkg_contents where
gpkg_contents.data_type = 'features' AND table_name exists in
gpkg_geometry_columns
                  Then
                    Set Metadata-feature(_m_) to Pass
                  Else
                    Set Metadata-feature(_m_) to Fail  (invalid no geometry
content for feature metadata)
                  End If
              Else If reference_scope = "row" (for md_scope = "feature")
                If table_name exists in gpkg_contents where
gpkg_contents.data_type = 'features' AND table_name exists in
gpkg_geometry_columns
                Then
                  Check that the feature exists in the feature table
                  If table_name.row_id equals gpkg_metadata_reference.row_id
where gpkg_metadata_reference.md_file_id = _m_
                  Then
                    Set Metadata-feature(_m_) to Pass
                  Else
                    Set Metadata-feature(_m_) to Fail (invalid no feature
content for metadata)
                  End If
              Else
                Set Metadata-tile(_m_) to Null (does not apply, contents are
not features)
              End If
            Else
              Set Metadata-feature(_m_) to Fail (invalid table name for
feature metadata_reference)
            End If
```

```
          End For
        Else
          Do nothing (no applicable metadata in gpkg_metadata_reference)
        End If
      End For
    End If
  End If

  If partial-feature-metadata is True
  Then
    Verify the "attributeType" or "attribute" in gpkg_metadata_reference.

    Repeat the results of the GPKG Options ATS for Requirement 119 and the
ATS for the Metadata Extension Annex F.8 Requirements 96-99. These ATS
validate the column_name and row_id of tile attributes in
gpkg_metadata_reference correspond to the tables. columns and row values in
gpkg_contents, gpkg_geometry_columns, gpkg_geometry and feature table for the
feature data set.
  End If
End For
```

Editor Note: the ATS for Requirement gridded-metadata-user is located in ATS A-10.5 for conformance class conf/gridded, in additon to other conformance in conf/metadata.

A.5 Conformance Class Validity

Table A-5.1 ATS 5.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/ref-sys
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Validity constraints for specific fields in GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/ref-sys)
Requirements:	/req/validity/data-validity, rows 1-5
Conformance Class:	validity
Test type:	Capability
Test method:	Validate the DGIWG GeoPackage table and column values to be compliant with the value constraints for CRS in Table-37 rows 1-5 for http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/ref-sys.

```
ATS produces a validation result of each crs in gpkg_spatial_ref_sys.

For each _srs_id_ in gpkg_spatial_ref_sys, test the following:
  If gpkg_spatial_ref_sys(_srs_id_) columns .organization, .description,
.definition, .definition_12-063 are consistent with the rules in Table 37
rows 1-5,
  Then Pass for gpkg_spatial_ref_sys(_srs_id_) srs_name;
  Else Fail for gpkg_spatial_ref_sys(_srs_id_) srs_name.
End For
```

Table A-5.2 ATS 5.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data_validity/contents
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Validity constraints for specific fields in GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/contents)
Requirements:	/req/validity/data-validity, rows 6-10
Conformance Class:	validity
Test type:	Capability
Test method:	Validate the DGIWG GeoPackage table and column values to be compliant with the value constraints for content validity in Table-37 rows 6-10 for http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/contents.

```
ATS produces a validation result of each data set table in gpkg_contents.

For each dataset _d_ in gpkg_contents, test the following:
  If gpkg_contents(_d_) columns .data_type, min_x, min_y, max_x, max_y are
consistent with the rules in Table 37 rows 6-10,
  Then Pass for gpkg_contents(_d_).table_name
  Else Fail for gpkg_contents(_d_).table_name
End For
```

Table A-5.3 ATS 5.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/geometry
Test purpose:	Verify that a DGIWG GeoPackage table and column values to be compliant with the value constraints for the geometry values in Table-37 rows 11-14 for (http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/geometry)
Requirements:	/req/validity-data-validity, rows 11-14
Conformance Class:	validity
Test type:	Capability
Test Method:	ATS produces a validation result of each feature set in gpkg_geometry_columns.

```
ATS returns Pass or Fail compliance for each feature set table in
geometry_columns; NULL if no features are in GeoPackage.

For each feature set _f_ in gpkg_geometry_columns, test the following:
  If gpkg_geometry_columns(_f_) .z, .table_name (feat_table) are consistent
with the rules in Table 37 rows 11-14
  Then Pass for gpkg_geometry_columns(_f_).table_name
  Else Fail for gpkg_geometry_columns(_f_).table_name
  End If
End For

The test for z = 0 and z = 1 are similar to ATS 3.6.
```

Table A-5.4 ATS 5.4

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/validit/data-validity/constraint
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Validity constraints for specific fields in GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/constraint)
Requirements:	/req/validity/data-validity row 15
Conformance Class:	validity
Test type:	Capability
Test method:	Validate the DGIWG GeoPackage table and column values to be compliant with the value constraints for the data column constraint name in Table-37 row 15 for\http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/constraint.

```
ATS produces a validation result of each row in gpkg_data_columns and each
constraint for a column.

For each column _col_ in gpkg_data_columns,
    If constraint_name for _col_ is not NULL and has mixed case or all
lowercase values,
    Then Pass for gpkg_data_columns(_col_).constraint_name
    Else Fail for gpkg_data_columns(_col_).constraint_name
End For

For each constraint _con_ in gpkg_data gpkg_data_columns_constraints
  If constraint_name for _con_ is  not NULL and has mixed case or all
lowercase values,
  Then Pass for gpkg_data_columns_constraints(_con_).constraint_name
  Else Fail for gpkg_data_columns_constraints(_con_).constraint_name
End For

Note: this requirement's ATS overrides the Requirement 106 in Extension F.9,
Schema, ATS /extensions/schema/data_columns/constraint_name.
```

Table A-5.5 ATS 5.5

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/tile-matrix
Test purpose:	Verify that a DGIWG GeoPackage (http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/tile-matrix)
Requirements:	/req/validity/data-validity 16-21
Conformance Class:	validity
Test type:	Capability
Test method:	Validate the DGIWG GeoPackage table and column values to be compliant with the value constraints for tile matrix definition in Table-37 rows 16-21 for http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/tile-matrix. The compliance test uses results of ATS A-3.2 to validate tile matrix sets in [6] OGC 17-083r2.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  If ATS 3.2 is Pass for _t_
  Then Pass this ATS for tile_matrix_set _t_
  Else verify the validity constraints Table 37 rows 16-21 are met for a
tile_matrix_set that is not defined in [6] OGC 17-083r2
     Verify zoom_level constraint <= max zoom in table.
     verify tile_width & tile_height = 256  (same as requirements
/req/tile/size-matrix and /req/tile/size-data, ATS 7.1)
     Verify pixel_x_size, pixel_y_size to be factor of 2 (same as requirement
/req/zoom/factor, ATS 8.1)
     Verify pyramid-table_name zoom_level constraint <= max zoom in table.
  End If
End For
```

Table A-5.6 ATS 5.6

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/metadata
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Validity constraints for specific fields in GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/metadata)
Requirements:	/req/validity-data-validity rows 22-30
Conformance Class:	validity
Test type:	Capability
Test method:	Validate the DGIWG GeoPackage table and column values to be compliant with the value constraints for the metadata definition in Table-37 rows 22-30 for http://www.dgiwg.org/std/gpkg/1.0/conf/validity/data-validity/metadata. Uses portions of the ATS for Extension F.8, Metadata.

```
ATS produces a validation result of each row in gpkg_metadata, including its
contents and rows in gpkg_metadata_reference associated with each
gpkg_metadata row.

If ATS 1.2 for GeoPackage Extension F.8 Pass
Then continue to perform more specific validation of the gpkg_metadata and
gpkg_metadata_reference tables.
For each row in gpkg_metadata with id _m_
  Set default Metadata-gpkg(_m_) to Pass.
  If _m_ is unique in gpkg_metadata table
    Validate table for _m_ using [4] GPKG F.8 ATS
/extensions/metadata/table_def
    If GPKG F.8 ATS /extensions/metadata/table_def for _m_ Fail, then set
Metadata-gpkg(_m_) to Fail.
    Else
      If [ gpkg_metadata(_m_).md_scope is equal to an integer value
corresponding to the values in Table 31
        AND gpkg_metadata(_m_).md_standard_id is a valid DMF URL or URL for
national metadata
        AND gpkg_metadata(_m_).mime_type = "text/xml" ]
      Then
        If gpkg_metadata(_m_).md_standard_id is a valid DMF URL Then
          Validate gpkg_metadata(_m_).metadata XML content
          Metadata-XML(_m_) = result of DMF ATS for a DMF Metadata Set [1]
DGIWG Metadata Foundation ATS A.4.
  Note: DMF metadata validation same as ATS 5.1.
        Else
```

```
          Note - use national ATS for national metadata validation.
          If ATS exists for national metadata,
          Then Metadata-XML(_m_) = result of ATS for national metadata.
          Else Metadata-XML(_m_) is Fail.
        End If
        Metadata-gpkg(_m_) = Metadata-XML(_m_) (result of metadata XML
validation for _m_)
      Else
        Set Metadata-gpkg(_m_) to Fail.
      End If

Check the other metadata validation in GPKG extension F.8.

      If Metadata-gpkg(_m_) is Pass AND
        IF any of the GPKG F.8 gpkg_metadata validity are False
        Then Set Metadata-gpkg(_m_) to Fail.
        Else

ATS continues to validate gpkg_metadata_reference if Metadata-gpkg(_m_) is
Pass from previous tests
          Validate gpkg_metadata_reference table for _m_ in [4] GPKG F.8 ATS
/extensions/metadata_reference/table_def .
          If GPKG F.8 ATS /extensions/metadata/table_def for _m_ Fail, then
set Metadata-gpkg(_m_) to Fail.
Table is valid, validate contents of each gpkg_metadata_reference for _m_
          Then
            For each row in gpkg_metadata_reference with md_file_id = _m_
              Validate every field in rows using [4] GPKG F.8 ATS
/extensions/metadata/metadata_reference/
              If GPKG F.8 ATS extensions/metadata/metadata_reference/ Fail
for _m_ and metadata_reference.md_file_id
              Then set Metadata-gpkg(_m_) to Fail.
            End for
          End If
        End If
      End If
    End If
  Else
    _m_ is not unique, set Metadata-gpkg(_m_) to Fail.
  End If
End For
```

A.6 Conformance Class Tile Matrix Set

Table A-6.1 ATS 6.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/tile/size-matrix
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Tile Matrix Size GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/tile/size-matrix)
Requirements:	/req/tile/size-matrix
Conformance Class:	tile-matrix-set
Test type:	Capability
Test method:	Validate the tile matrix size in DGIWG GeoPackage has width and height value equal to 256 to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/tile/size-matrix.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set. If any
pyramid tile set fails, the entire tile_matrix_set fails.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  Set default Tile_Matrix_Set-size(_t_) to Pass.

  If ATS 3.2 is Fail or Null for _t_
  Then
    Verify the Pyramid tile matrix size is valid for a tile_matrix_set that
is not defined in [6] OGC 17-083r2 (ATS 3.2 Fail for _t_)
    For each pyramid tile _z_ in gpkg_tile_matrix
      If tile_width != 256 or tile_height != 256 for _z_ Then Set
Tile_Matrix_Set-size(_t_) to Fail.
    End For
  Else
    Set Tile_Matrix_Set-size(_t_) to Fail.
  End If
End For
```

Table A-6.2 ATS 6.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/tile/size-data
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Tile Size GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/tile/size-data)
Requirements:	/req/tile/size-data

Conformance Class:	tile
Test type:	Capability
Test method:	Validate every pyramid data tile in DGIWG GeoPackage has width and height value equal to 256 to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/tile/size-data.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set. If any
data tile fails, the entire tile_matrix_set fails.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  Set default Tile_Matrix_Set-data_size(_t_) to Pass.
  If ATS 3.2 is Fail or Null for _t_
  Then
    Verify the Table data size in the Pyramid tile matrix size is valid for a
gpkg_tile_matrix_set that is not defined in [6] OGC 17-083r2 (ATS 3.2 Fail)
        For each pyramid tile _z_ in gpkg_tile_matrix
          For each data tile _d_ in gpkg_tile_matrix(_z_).table_name
            If width of gpkg_tile_matrix(_z_).table_name.tile_data(_d_) !=
256 Then Set Tile_Matrix_Set-data_size(_t_) to Fail.
            If height of gpkg_tile_matrix(_z_).table_name.tile_data(_d_) !=
256 Then Set Tile_Matrix_Set-data_size(_t_) to Fail.
          End For
        End For
  End If
End For
```

A.7 Conformance Class Zoom

Table A-7.1 ATS 7.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/factor
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Zoom levels for GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/factor)
Requirements:	/req/zoom/factor
Conformance Class:	zoom
Test type:	Capability

Test method:	Validate every zoom pixel size in DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/factor.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set. If any
zoom level fails, the entire tile_matrix_set fails.

Case 1: gpkg_tile_matrix_set is consistent with [6] OGC 17-083r2, which has
zoom levels in factors of 2. Verify all consecutive zoom levels are adjacent.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  Set default Tile_Matrix_Set-size(_t_) to Pass.
  If ATS 3.2 is Pass for _t_
  Then
    For each zoom_level _z_ in gpkg_tile_matrix, starting at z = 1
      If zoom level _z_ is not an adjacent zoom level to (_z_ - 1)
      Then Set Tile_Matrix_Set-zoom(_t_) to Fail
      End If
    End For
  Else
    Set Tile_Matrix_Set-data_zoom(_t_) to Fail.
  End If

Case 2: gpkg_tile_matrix_set zoom levels defined in gpkg_tile_matrix; zoom
factor 2.0 within 1/100.

  Else
    Assumes first level of gpkg_tile_matrix is p = 1.
    For each zoom_level _z_ in gpkg_tile_matrix, starting at z = 1
      If [pixel_x_size(_z_-1) / pixel_x_size(_z_)] not equal to 2.0 within
1/100 OR [pixel_y_size(_z_-1)/pixel_y_size(_z_)] not equal to 2.0 within
1/100
      Then Set Tile_Matrix_Set-size(_t_) to Fail.
      End If
    End For
  End If

End For
```

Table A-7.2 ATS 7.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/matrix-sets-multiple
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Matrix Sets GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/matrix-sets-multiple)
Requirements:	/req/zoom/matrix-sets-multiple
Conformance Class:	zoom
Test type:	Capability
Test method:	Validate every mandatory tile matrix set with multiple zoon levels in DGIWG GeoPackage to be compliant with all or a subset of the zoom levels for the corresponding CRS in Annex D of [6] OGC 17-083r2.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set. If any
zoom level fails, the entire tile_matrix_set fails.

Case 1: gpkg_tile_matrix_set is consistent with [6] OGC 17-083r2. Verify all
consecutive zoom levels are adjacent.

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  Set default Tile_Matrix_Set-multiple-zoom(_t_) to Pass.
  If ATS 3.2 is Pass for _t_
  Then
    For each zoom_level _z_ in gpkg_tile_matrix, starting at z = 1
      If pkg_tile_matrix(_z_).zoom_level is not adjacent (difference of 1) to
zoom level gpkg_tile_matrix(_z_ - 1).zoom_level
      Then Set Tile_Matrix_Set-multiple-zoom(_t_) to Fail.
    End For
  Else

Case 2: not an OGC 2D TMS, this requirement is not applicable.

    Set Tile_Matrix_Set-multiple-zoom(_t_) to Null.
  End If
End For
```

Table A-7.3 ATS 7.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/matrix-sets-one

Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Matrix Sets GPKG (http://www.dgiwg.org/gpkg/1.0/conf/zoom/matrix-sets-one)
Requirements:	/req/zoom/matrix-sets-one
Conformance Class:	zoom
Test type:	Capability
Test method:	Validate every tile matrix set in DGIWG GeoPackage that does not have a set of zoom levels for the corresponding CRS in Annex D of [6] OGC 17-083r2 has only one zoom level to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/zoom/matrix-sets-one.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set if the
set only has 1 zoom level; else the test result is not applicable (NULL).

For each tile matrix set _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  Set default Tile_Matrix_Set-one-zoom(_t_) to Pass.
  If count of rows in gpkg_tile_matrix(_t_).zoom_level = 1
  Then
    If gpkg_tile_matrix(_t_).zoom_level != zoom_level in
gpkg_tile_matrix(_t_).table_name
    Then Set Tile_Matrix_Set-one-zoom(_t_) to Fail.
    Else
      Note - gpkg_tile_matrix(_t_) has > 1 zoom levels.
      Set Tile_Matrix_Set-one-zoom(_t_) to Null.
  End If
End For
```

A.8 Conformance Class Bounding Box (bbox)

Table A-8.1 ATS 8.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.0/conf/bbox/crs
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the CRS BBOX GPKG (http://www.dgiwg.org/std/gpkg/1.0/conf/bbox/crs)
Requirements:	/req/bbox/crs
Conformance Class:	bbox
Test type:	Capability
Test method:	Validate every CRS bounding box in DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.0/conf/bbox/crs.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set - Pass
within CRS MBR, Fall outside CRS MBR or invalid CRS in tile matrix set.

// Note: _t_ is table_name in gpkg_contents and gpkg_tile_matrix_set
For each gpkg_tile matrix_set  _t_ in gpkg_contents where the
gpkg_contents.data_type of _t_ = "tiles",
  If ATSX.Y for _t_ is Pass (the crs/srs is valid)
  Then
    Where the srs_id of gpkg_tile_matrix_set for _t_ equals the corresponding
crs _c_ in gpkg_spatial_ref_sys.srs_id
    If the bounds of tile_matrix_set _t_ are within the bounds of crs _c_
within 1 meter
      Set Tile_Matrix_Set-bbox(_t_) to Pass.
      A within expression below can be used when it accounts for E/W lon x
and N/S lat y.
       gpkg_tile_matrix_set(_t_).min_x >= gpkg_spatial_ref_sys(_c_).min_x AND
       gpkg_tile_matrix_set(_t_).min_y >= gpkg_spatial_ref_sys(_c_).min_y AND
       gpkg_tile_matrix_set(_t_).max_x <= gpkg_spatial_ref_sys(_c_).max_x AND
       gpkg_tile_matrix_set(_t_).max_y <= gpkg_spatial_ref_sys(_c_).max_y
    Then
      Set Tile_Matrix_Set-bbox(_t_) to Fail
    End If
  Else (crs/srs__id is invalid in gpkg_tile_matrix_set)
    Set Tile_Matrix_Set-bbox(_t_) to Fail
  End If
End For
```

A.9 Conformance Class Gridded Data CRS

Table A-9.1 ATS 9.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-2d-allowed
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to the Gridded CRS GPKG (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-2d-allowed)
Requirements:	/req/crs/gridded-crs-2d-allowed
Conformance Class:	gridded-crs
Test type:	Capability
Test method:	Validate the two dimensional (2D) Coordinate Reference Systems (CRS) in each Gridded Coverage Data Set used in DGIWG GeoPackage complies with http://www.dgiwg.org/std/gpkg/1.1/conf/req/crs/gridded-crs-2d-allowed .

```
ATS returns NULL (N/A), Pass or Fail compliance for each each
2d_gridded_coverage CRS in gpkg_tile_matrix_set.

This ATS verifies gpkg_tile_matrix_set has a valid 2D srs_id for 2D gridded
data.

Assumption: if quantity_definition is not equal to elevation or height, the
tile matrix set is 2 dimensional.

For each tile matrix set _g_ in gpkg_contents where the
gpkg_contents.data_type of _g_ = "2d_gridded_coverage"
  _srs_id_ = gpkg_tile_matrix_set(_g_).srs_id
  If the _srs_id_ equals the corresponding gpkg_spatial_ref_sys.srs_id
  Then
    Set the default tile-matrix-set-CRS(_g_) to Pass.
    Determine if the tile-matrix-set is 2D.
    If the gpkg_2d_gridded_coverage_ancillary table quantity_definition
column for _g_ does NOT contain the word "elevation" or "height" (case
insensitive)
    Then gpkg_tile-matrix-set(_g_) is two dimensional.

      Determine if this _srs_id_ is an allowed 2D SRS
      If the gpkg_spatial_ref_sys.srs_name and srs_id of _srs_id_ does not
match one of the CRS in Table 11 or Table 12 with CRS Dimension = 2 that
corresponds to the srs_name or srs_id defined in section 7.2 tables 16
through 21
```

```
      Then
        set tile-matrix-set-CRS(_g_) to Fail (invalid CRS for 2D)
      End If
    Else
      Set tile-matrix-set-CRS(_g_) to NULL (tile-matrix-set(_g_) is 3D, test
is not applicable)
    End If
  Else
    set tile-matrix-set-CRS(_g_) to Fail (_srs_id_ for _g_ not defined in
gpkg_spatial_ref_sys)
  End If
End For
////
Without assumption, this is an alternative ATS implementation that is less
stringent.
  Assume GeoPackage is 2D if the srs_name or _srs_id_ is one of the 2D SRS in
the 2D CRS tables. Do not fail CRS if it doesn't have a 2D CRS; instead
report a warning if gpkg_tile_matrix_set contains a 2D SRS used and
quantity_definition is equal to Height or Elevation.
////
```

Table A-9.2 ATS 9.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-3d-allowed
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to the Gridded CRS GPKG (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-3d-allowed)
Requirements:	/req/crs/gridded-crs-3d-allowed
Conformance Class:	gridded-crs
Test type:	Capability
Test method:	Validate the three dimensional (3D) Coordinate Reference Systems (CRS) in each Gridded Coverage Data Set used in DGIWG GeoPackage complies with http://www.dgiwg.org/std/gpkg/1.1/conf/req/crs/gridded-crs-3d-allowed .

```
ATS returns NULL (N/A), Pass or Fail compliance for each each
2d_gridded_coverage CRS in gpkg_tile_matrix_set.

This ATS verifies gpkg_tile_matrix_set has a valid 3D srs_id for 3D gridded
data.
ATS 9.3 validates the horizontal and vertical CRS contained in the WKT
```

```
COMPOUNDCRS.

Assumption: if quantity_definition is equal to elevation or height, the tile
matrix set is 3 dimensional.

For each tile matrix set _g_ in gpkg_contents where the
gpkg_contents.data_type of _g_ = "2d_gridded_coverage"
  _srs_id_ = gpkg_tile_matrix_set(_g_).srs_id
  If the _srs_id_ equals the corresponding gpkg_spatial_ref_sys.srs_id
  Then
    Set the default tile-matrix-set-CRS(_g_) to Pass.
    Determine if the tile-matrix-set(_g_) is 3D.
    If the gpkg_2d_gridded_coverage_ancillary table quantity_definition
column for _g_ contains the word "elevation" or "height" (case insensitive)
    Then gpkg_tile-matrix-set(_g_) is three dimensional.

      Determine if this _srs_id_ is an allowed 3D SRS
      Set _srs_id_ to gpkg_spatial_ref_sys.srs_id
      If the gpkg_spatial_ref_sys.srs_name or
gpkg_spatial_ref_sys.organizaion_coordsys_id or _srs_id_ matches one of the
3D CRS in table 14 OR contains an srs_name or srs_id in the Compound CRS
defined in section 7.2 tables 27, 28 and 29.
      Then
        set tile-matrix-set-CRS(_g_) to Pass (CRS value match for 3D)
      Else
//      srs_id or organizaion_coordsys_id or srs_name does NOT match one of
the 3D CRS for gridded coverage
//      Examine the WKT for a valid horizontal and vertical CRS
        set _wkt_def_ = gpkg_spatial_ref_sys.definition(_srs_id_)
        set _wkt_def_063_ = gpkg_spatial_ref_sys.definition_12_063(_srs_id_)

        If (_wkt_def_ is NOT NULL and contains "COMPOUNDCRS" or "COMPD_CS")
OR
         (_wkt_def_063_ is NOT NULL and contains "COMPOUNDCRS" or "COMPD_CS")
        Then
// For Compound 3D SRS, the WKT is examined to determine if the CRS is
compliant, same as ATS 9.3 for requirement /req/crs/gridded-crs-compound-wkt
          set tile-matrix-set-CRS(_g_) to the same result as ATS 9.3 for CRS
_srs_id_
        Else
          set tile-matrix-set-conform(_g_) to Fail (no CRS match and not a
Compound SRS)
        End If
      End If
    Else
```

```
      set tile-matrix-set-CRS(_g_) to NULL (tile-matrix-set(_g_) is 2D, test
is not applicable)
    End If
  Else
    set tile-matrix-set-CRS(_g_) to Fail (_srs_id_ for _g_ not defined in
gpkg_spatial_ref_sys)
  End If
End For
```

Table A-9.3 ATS 9.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-compound-wkt
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to the Gridded Compound CRS WKT, when applicable (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-compound-wkt)
Requirements:	/req/crs/gridded-crs-compound-wkt
Conformance Class:	gridded-crs
Test type:	Capability
Test method:	Validate every Gridded Data compound CRS used in DGIWG GeoPackage contains WKT compliant according to Table-27, Table-28, and Table-29 with optional and mandatory elements appropriate for the CRS / SRS as defined by WKT2 [9]) to comply with DGIWG GeoPackage http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-compound-wkt.

```
ATS returns NULL (N/A), Pass or Fail compliance for each 2d_gridded_coverage
CRS in gpkg_spatial_ref_sys that has 3D gridded data.

Assumptions.

1. This ATS validates the WKT for 3D gridded data including the CRS for the
horizontal component and existance of a vertical CRS. In this ATS, any unique
srs_id is valid. The COMPOUND CRS is valid WKT1 or WKT2 and contains a 3D CRS
defined by one of the allowed horizontal CRS ID and a VERTCS. The srs_id and
organization_coordsys_id in Tables 28, 29, and 30 are examples.

2. The 3D Compound srs_id values reocmmended in secton 7.2.4 are not
enforced.
```

```
3. For 3D Gridded COMPOUNDCRS, the WKT in
gpkg_spatial_ref_sys.definition_12_063 is verified before
gpkg_spatial_ref_sys.defintion because the WKT extension is required for 2D
gridded coverage.

4. The WKT for COMPOUNDCRS is first examined in
gpkg_spatial_ref_sys.definition_12_063 per Geopackage WKT Extension v1.1;
else examine gpkg_spatial_ref_sys.definition .

5. If the 3D CRS WKT is not COMPOUND, the WKT is validated in ATS for
requirement /req/crs/wkt.

For each tile matrix set _g_ in gpkg_contents where the
gpkg_contents.data_type of _g_ = "2d_gridded_coverage"
  Set the default tile-matrix-set-conform(_g_) to Null.
  _srs_id_ = gpkg_tile_matrix_set(_g_).srs_id

  If the _srs_id_ equals the corresponding gpkg_spatial_ref_sys.srs_id
  Then
    Validate this _srs_id_ WKT if it is a COMPOUNDCRS.
    _wkt_def_ = gpkg_spatial_ref_sys.definition(_srs_id_)
    _wkt_def_063_ = gpkg_spatial_ref_sys.definition_12_063(_srs_id_)

// Check definition_12_063 first
    set _crs_wkt_ to _wkt_def_063_

START WKT-COMPOUND-VALIDATION
    If _crs_wkt_ is NOT NULL and contains "COMPOUNDCRS" or "COMPD_CS"
    Then
        Validate the _crs_wkt_ is consistent with the WKT definition in
Table-27, Table-28, and Table-29  by the following.
        // WGS84 geographic COMPOUNDCRS

        If _crs_wkt_ is valid WKT1 or WKT2 format.
        Then
        // Specific WKT SRS are conditional based on the Horizontal SRS ID
defined in COMPOUNDCRS WKT
          _comp_horiz_ID_ set to 0
          Retrieve WKT2 Geographic horizontal CRS
          _comp_horiz_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPOUNDCRS[ ... [GEOGCRS ... [ID"EPSG",comp_ID]
          If _comp_horiz_ID_ is 0, THEN
             Retrieve WKT1 Geograpchic horizontal CRS
             _comp_horiz_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPD_CS[ ... [GEOGCRS ... [AUTHORIY,"EPSG",comp_ID]
```

```
          End If
          If _comp_horiz_ID_ is 0, THEN
             Retrieve WKT1 Geograpchic horizontal CRS
             _comp_horiz_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPD_CS[ ... [GEOGCS ... [AUTHORIY,"EPSG",comp_ID]
          End If
          If _comp_horiz_ID_ is 0, THEN
            Retrieve WKT2 projected horizontal CRS
             _comp_horiz_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPOUNDCRS[ ... [PROJCS ... [ID,"EPSG",comp_ID]
          End IF
          If _comp_horiz_ID_ is 0, THEN
            Retrieve WKT1 projected horizontal CRS
             _comp_horiz_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPD_CS[ ... [PROJ_CS ... [AUTHORIY,"EPSG",comp_ID]
          End If,

          If _comp_horiz_ID_ is 0
          Then
 // WGS84 geographic COMPOUNDCRS
 //       Convention (not enforced): _srs_id_ >= 100100 and <= 100199
 //       Either crs_definition file or CRS Name match in Annex A validation
is acceptable; crs_definition is dynamic
              If _comp_horiz_ID_ is equal to the id : code attribute for any
crs with type : enum = GeodeticCRS in https://www.dgiwg.org/def/crs/100100
              OR _comp_horiz_ID_ is equal to one of 'CRS AUTH ID' Annex C
Table C-1 where CRS Name contains 'WGS 84 (G*)' (* is an integer number)
              Then
// Check existence of VERTCRS and code value for the WGS84 compound crs
definition
                _comp_vert_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPOUNDCRS / VERTCRS or COMPOUNDCRS / VERT_CS
                If WKT _comp_vert_ID_ is not NULL AND _comp_vert_ID_ is equal
to the id : code attribute for any crs with type : enum = VerticalCRS in
https://www.dgiwg.org/def/crs/100100
                Then set tile-matrix-set-conform(_g_) to Pass
                Else set tile-matrix-set-conform(_g_) to Fail
                End If
              Else
// UTM N or UTM S
//        Convention (not enforced): _srs_id_ >= 100200 and <= 100299
//        Either crs_definition file or numeric validation is acceptable;
crs_definition is dynamic
                If _comp_horiz_ID_ is equal to the id : code attribute for
any crs with type : enum = ProjectedCRS in
```

```
https://www.dgiwg.org/def/crs/100200
                OR _comp_horiz_ID_ > 32600 AND _comp_horiz_ID_ < 32661 OR
_comp_horiz_ID_ > 32700 AND _comp_horiz_ID_ < 32761
                Then
// Check existence of VERTCRS and code value for the WGS84 compound crs
definition
                  _comp_vert_ID_ = WKT _crs_wkt_ extracted comp_ID value from
COMPOUNDCRS / VERTCRS or COMPOUNDCRS / VERT_CS
                  If WKT _comp_vert_ID_ is not NULL AND _comp_vert_ID_ is
equal to the id : code attribute for any crs with type : enum = VerticalCRS
in https://www.dgiwg.org/def/crs/100200
                  Then set tile-matrix-set-conform(_g_) to Pass
                  Else set tile-matrix-set-conform(_g_) to Fail (missing
vertical SRS)
                  End If
// UPS - Polar
//       Convention (not enforced): _srs_id_ >= 100300 and <= 100399
               Else
 //        Either crs_definition file or numeric validation is acceptable;
crs_definition is dynamic
                  If _comp_horiz_ID_ is equal to the id : code attribute for
any crs with type : enum = ProjectedCRS in
https://www.dgiwg.org/def/crs/100300
                  OR _comp_horiz_ID_ = 5041 OR _comp_horiz_ID_ = 5042
                  Then
// Check existence of VERTCRS
                    _comp_vert_ID_ = WKT _crs_wkt_ extracted comp_ID value
from COMPOUNDCRS / VERTCRS or COMPOUNDCRS / VERT_CS
                    If WKT _comp_vert_ID_ is not NULL AND _comp_vert_ID_ is
equal to the id : code attribute for any crs with type : enum = VerticalCRS
in https://www.dgiwg.org/def/crs/100300
                    Then set tile-matrix-set-conform(_g_) to Pass
                    Else set tile-matrix-set-conform(_g_) to Fail (missing
vertical SRS)
                  End If
                Else
// Failed to find _comp_horiz_ID_ in any of the 3D COMPOOUNDCRS, this srs_id_
is allowed by DGIWG
                    set tile-matrix-set-conform(_g_) to Fail (invalid
Horizontal SRS - failed all 3 tests)
                End If
              End If
          Else
              set tile-matrix-set-conform(_g_) to Fail  (no Horizontal
Compound _srs_id_ for 3D)
```

```
          End If
        Else
          set tile-matrix-set-conform(_g_) to Fail (invalid WKT for
COMPOUNDCRS)
        End If
END WKT-COMPOUND-VALIDATION
    Else
      If _wkt_def_ is NOT NULL
      Then
// Check definition WKT second
        set _crs_wkt_ to _wkt_def_ ( definition(_srs_id_) )
        Execute WKT-COMPOUND-VALIDATION above with _wkt_def_
      Else
        set tile-matrix-set-conform(_g_) to Fail (WKT in both definition
fields is NULL)
      End If
    End If
  Else
    set tile-matrix-set-conform(_g_) to Fail (_srs_id_ for _g_ not defined in
gpkg_spatial_ref_sys)
  End If
End For (tile_matrix_set loop)

Return tile-matrix-set-conform (NULL for N/A, Pass, or Fail)
```

Table A-9.4 ATS 9.4

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-epoch-wkt
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to the Compound CRS EPOCH WKT, when applicable (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-epoch-wkt)
Requirements:	/req/crs/gridded-crs-epoch-wkt
Conformance Class:	gridded-crs
Test type:	Capability
Test method:	Validate every Gridded Data compound CRS used in DGIWG GeoPackage contains WKT compliant with the DYNAMIC FRAMEEPOCH and EPOCH column value as defined by GeoPackage WKT extension 1.1.0 [11] to comply with DGIWG GeoPackage http://www.dgiwg.org/std/gpkg/1.1/conf/gridded-crs/gridded-crs-epoch-wkt. Conditional based upon whether the EPOCH field is set in gpkg_spatial_ref_sys. If so, verify the COMPOUNDCRS in the definition_12_063 column of gpkg_spatial_ref_sys contains the FRAMEEPOCH field.

```
ATS returns NULL (N/A), Pass or Fail compliance for each 2d_gridded_coverage
CRS in gpkg_spatial_ref_sys

Assumption: The WKT for COMPOUNDCRS is in
gpkg_spatial_ref_sys.definition_12_063 per Geopackage WKT Extension v1.1 for
FRAMEEPOCH to be present.

The WKT2 for COMPOUNDCRS is in gpkg_spatial_ref_sys.definition_12_063 per
Geopackage WTK Extension v. 1.1.

Note 1: This test does not apply to projected CRS, only the COMPOUNDCRS with
WGS84 geographic coordinates use the DYNAMIC FRAMEEPOCH.
Note 2: Epoch value is defined in OGC 18-005r4, OGC Abstract Specification
Topic 2: Referencing by coordinates, https://docs.opengeospatial.org/as/18-
005r4/18-005r4.html#68 .

For each tile matrix set _g_ in gpkg_contents where the
gpkg_contents.data_type of _g_ = "2d_gridded_coverage"
  Set the default tile-matrix-set-conform(_g_) to Null.
  _srs_id_ = gpkg_tile_matrix_set(_g_).srs_id
```

```
  If the _srs_id_ equals the corresponding gpkg_spatial_ref_sys.srs_id
  Then
    If gpkg_spatial_ref_sys(_srs_id).EPOCH is not NULL
    Then
     If gpkg_spatial_ref_sys(_srs_id).EPOCH is a valid Epoch DATE (YYYY or
YYYY.DDD) where DDD is between 1 and 365.
     Then
      _wkt_def_063 = gpkg_spatial_ref_sys.definition_12_063(_srs_id_)

      If _wkt_def_063is NOT NULL and contains "COMPOUNDCRS"
        _comp_frame_epoch_ = date field extracted value from _wkt_def_063
COMPOUNDCRS[ ... [DYNAMIC[FRAMEEPOCH[date]]
        If _comp_frame_epoch_ is present in _wkt_def_063
        Then
          If _comp_frame_epoch_ is a valid Epoch DATE (YYYY or YYYY.DDD)
where DDD is between 1 and 365.
          Then
            set tile-matrix-set-conform(_g_) to Pass
          Else
            set tile-matrix-set-conform(_g_) to Fail (invalid date in
FRAMEEPOCH)
          End If
        Else
          set tile-matrix-set-conform(_g_) to Fail (no FRAMEEPOCH present in
WKT)
        End If
      Else
        set tile-matrix-set-conform(_g_) to Fail (WKT is null or COMPOUNDCRS
not in WKT)
      End If
    Else
       set tile-matrix-set-conform(_g_) to NULL (EPOCH field not present for
this _srs_id_)
    End If
  Else
      set tile-matrix-set-conform(_g_) to Fail (_srs_id_ for _g_ not defined
in gpkg_spatial_ref_sys)
  End If
End For (tile_matrix_set loop)

Return tile-matrix-set-conform (NULL for N/A, Pass, or Fail)
```

A.10 Conformance Class Gridded Data

Table A-10.1 ATS 10.1

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/gridded-coverage-extension-core
Test purpose:	Verify that a DGIWG GeoPackage using the Gridded Coverage Extension (F.11) is conformant to the Core Tiled Gridded Coverage Core Conformance Class (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/gridded-coverage-extension-core)
Requirements:	/req/gridded/gridded-coverage-extension-core
Conformance Class:	gridded
Test type:	Capability
Test method:	Validate a DGIWG GeoPackage with extension 2d_gridded_coverage to be in conformance with [5] OGC 17-066r2 . Use the exectuable table suite for test class org.opengis.cite.gpkg12.extensions.coverage in the OGC GeoPackage Conformance Class Tiled Gridded Coverage Core - [5] OGC 17-066r2 defined in https://cite.opengeospatial.org/teamengine/about/gpkg12/1.2/ (use current version).

Gridded Data tests A.10.2 and A.10.3 are specific for 2d_gridded_coverage containing elevation and bathymeric data, which are tested assuming the quantity_definition column contains "elevation" or "height". Tests A.10.3 and A.10.4 apply to any 2d_gridded_coverage data.

Table A-10.2 ATS 10.2

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/field-name-elevation
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to expected value of field_name for elevation data (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/field-name-elevation)
Requirements:	/req/gridded/field-name-elevation
Conformance Class:	gridded
Test type:	Capability
Test method:	Validate a DGIWG GeoPackage with data_type of 2d_gridded_coverage, for each tile set containing elevation data has the expected quantity_definition field_name for elevation compliant with http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/field-name-elevation.

```
ATS returns NULL (N/A), Pass or Fail compliance for each each
2d_gridded_coverage CRS in gpkg_tile_matrix_set.

This ATS verifies gpkg_tile_matrix_set has a valid quantity_definition and
field_name that are consistent for elevation and bathymetry.

Assumption: if quantity_definition is equal to elevation or height, the tile
matrix set has same data_value. This ATS needs to be expanded if other pre-
defined types of 2D or 3D gridded_coverages are defined.

For each tile matrix set _g_ in gpkg_contents
  set default tile-matrix-set-gridded_value(_g_) to Null (not applicable to
requirement)

  If gpkg_contents.data_type of _g_ = "2d_gridded_coverage"
  Then
      Determine if the tile-matrix-set is elevation or bathymetry.
      If the gpkg_2d_gridded_coverage_ancillary table quantity_definition
column for _g_ contains the word "elevation" or "height" (case insensitive)
      Then
        If the gpkg_2d_gridded_coverage_ancillary table field_name column for
_g_ contains the word "elevation" or "height" (case insensitive)
        Then
           set tile-matrix-set-gridded_value(_g_) to Pass
        Else
          set tile-matrix-set-gridded_value(_g_) to Fail (field_name invalid
for elevation or bathymetry)
        End If
      Else
         No test applies, tile_matrix_set(g) is another type of gridded_data
(2D or 3D atmpospheric, for example)
      End If
     End If
  End If
End For
```

Table A-10.3 ATS 10.3

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/center-value-elevation
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to expected value of center value for elevation data (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/center-value-elevation)
Requirements:	/req/gridded/center-value-elevation

Conformance Class:	gridded
Test type:	Capability
Test method:	Validate a DGIWG GeoPackage with data_type of 2d_gridded_coverage, for each tile set containing elevation data has the expected center value for elevation compliant with http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/center-value-elevation.

```
ATS returns NULL (N/A), Pass or Fail compliance for each each
2d_gridded_coverage CRS in gpkg_tile_matrix_set.

This ATS verifies gpkg_tile_matrix_set has a valid quantity_definition and
field_name that are consistent for elevation and bathymetry.

Assumption: if quantity_definition is equal to elevation or height, the tile
matrix set has same data_value. This ATS needs to be expanded if other pre-
defined types of 2D or 3D gridded_coverages are defined.

For each tile matrix set _g_ in gpkg_contents
  set default tile-matrix-set-gridded_cell_encoding_center(_g_) to Null (not
applicable to requirement)

  If gpkg_contents.data_type of _g_ = "2d_gridded_coverage"
  Then
      Determine if the tile-matrix-set is elevation or bathymetry.
      If the gpkg_2d_gridded_coverage_ancillary table quantity_definition
column for _g_ contains the word "elevation" or "height" (case insensitive)
      Then
        If the gpkg_2d_gridded_coverage_ancillary table grid_cell_encoding
column for _g_ equals "grid-value-is-center" (case insensitive)
        Then
           set tile-matrix-set-gridded_cell_encoding_center(_g_) to Pass
        Else
          set tile-matrix-set-gridded_cell_encoding_center(_g_) to Fail
(encoding value invalid for elevation or bathymetry)
        End If
      Else
        No test applies, tile_matrix_set(g) is another type of gridded_data
(2D or 3D atmpospheric, for example)
      End If
     End If
  End If
End For
```

Table A-10.4 ATS 10.4

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/zoom-factor
Test purpose:	Verify that a DGIWG GeoPackage with Gridded Data is conformant to the zoom factor (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/zoom-factor)
Requirements:	/req/gridded/zoom-factor
Conformance Class:	gridded
Test type:	Capability
Test method:	Validate every zoom pixel size in DGIWG GeoPackage with Gridded Data tile sets to be compliant with http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/zoom-factor.

```
ATS produces a validation result of each row in gpkg_tile_matrix_set. If any
zoom level fails, the entire tile_matrix_set fails.

For each tile matrix set _g_ in gpkg_contents where the
gpkg_contents.data_type of _g_ = "2d_gridded_coverage",
  Set default Tile_Matrix_Set-size(_g_) to Pass.

  Assumes first level of gpkg_tile_matrix is z = 0.

    For each zoom_level _z_ in gpkg_tile_matrix, starting at z = 1
      If [pixel_x_size(_z_-1) / pixel_x_size(_z_)] not equal to 2.0 within
1/100 OR [pixel_y_size(_z_-1)/pixel_y_size(_z_)] not equal to 2.0 within
1/100
      Then Set Tile_Matrix_Set-size(_g_) to Fail.
      End If
    End For

End For
```

Table A-10.5 ATS 10.5

Test identifier:	http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/metadata-gridded-user
Test purpose:	Verify that a DGIWG GeoPackage is conformant to the Gridded Coverage Metadata GPKG (http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/metadata-gridded-user)
Requirements:	/req/gridded/metadata-gridded-user
Conformance Class:	metadata
Test type:	Capability
Test method:	Validate every Gridded Coverage tileSet, tile, attributeType or attribute user-defined metadata in DGIWG GeoPackage to be compliant with http://www.dgiwg.org/std/gpkg/1.1/conf/gridded/metadata-gridded-user.\ **Assumption** The GeoPackage has valid result of GPKG Options ATS for Requirement 119 and the ATS for the Metadata Extension Annex F.8 Requirements 96-99. These ATS validate the column_name and row_id of tile attributes in gpkg_metadata_reference correspond to the tables, columns and row values in gpkg_contents, gpkg_tile_matrix_set, gpkg_tile_matrix and pyramid tiles for the tile_matrix_set.

```
ATS produces a validation result for each row of tile matrix set or tile
metadata in the gpkg_metadata table; null for those metadata that are not
tile matrix sets or tiles.

For each row in gpkg_metadata with id _m_
  Set default Metadata-tile(_m_) to NULL.
  If gpkg_metadata.scope(_m_) is equal to "model" or "tile" or
"attributeType" or "attribute"
  Then
    Check for valid metadata definition.

    If gpkg_metadata(_m_).md_standard_id is a valid URL AND
gpkg_metadata(_m_).metadata is XML content AND gpkg_metadata(_m_]).mime_type
= "text/xml"
    Then
      Check the gpkg_metadata_reference table row exists for this
gpkg_metadata row.

      For each reference_scope, table_name, column_name, row_id from
gpkg_metadata_reference.md_file_id = _m_
```

```
        If results are not empty result set (at least one reference_scope
exists for this metadata record)
        Then
          To validate requirement for gridded data, distinguish between
data_type = "2d_gridded_data" and data_type = "tiles" or "features"

          For each dataset _g_ in gpkg_contents

            If gpkg_metadata_reference.table_name =
gpkg_contents(_g_).table_name
            Then
              If gpkg_contents(_g_).data_type = "2d_gridded_data"
              Then this gpkg_metadata_reference is for 2d_gridded data

                If (reference_scope = "table" and md_scope(_m_) ="model") OR
                  (reference_scope = "row/col" and md_scope(_m_) ="tile") OR
                  (reference_scope = "column" and md_scope(_m_)
="attributeType") OR
                  (reference_scope = "row/col" and md_scope(_m_)
="attribute")
                Then
                    If table_name exists in gpkg_tile_matrix_set
                    Then
                      Set Metadata-tile(_m_) to Pass
                    Else
                      Set Metadata-tile(_m_) to Fail (invalid reference_scope
table for gridded data)
                    End If
                 Else
                    Set Metadata-tile(_m_) to Fail (invalid reference_scope
for 2d_gridded_data)
                End If
              Else
                  Set Metadata-tile(_m_) to Null (does not apply, contents
are not 2d_gridded_data)
              End If
            Else
              Set Metadata-tile(_m_) to Fail (invalid table in gpkg_metadata)
            End If
          End For
       Else
          Do nothing (no metadata in gpkg_metadata)
       End If
    Else
      Set Metadata-tile(_m_) to Fail (invalid metadata record)
```

```
    End If
  Else
    Do nothing (no applicable metadata in gpkg_metadata_reference)
  End If
End For
```

Annex B GeoPackage Metadata Reference

The relationships between DMF and NMF in Table-B-1 illustrate which DMF terms can be translated from one standard to another.

Table B-1: Mapping of DMF/Core to NMF Core metadata

DMF Identifier	DMF Title	NMF (v3.0) Core Metadata Concept
MDSID	Metadata Set Identifier	
MDPTMD (DMF/Common)	Parent Metadata Set (DMF/Common)	Parent Metadata Citation
MDDLOC	Metadata Default Locale	Metadata Language, Metadata Character Set
MDRPTY	Metadata Responsible Party	Metadata Point of Contact
MDDATE	Metadata Date Stamp	Metadata Date
MDSTD	Metadata Standard	Metadata Standard Title, Metadata Standard Edition
MDSCST (DMF/Common)	Metadata Security Constraint (DMF/Common)	Metadata Classification, Metadata Classification System
RSTITLE	Resource Title	Resource Title
RSABSTR	Resource Abstract	Resource Abstract
RSPURP	Resource Purpose	
RSTYPE	Resource Type Code	Metadata Scope Code
RSTYPN	Resource Type Name	Metadata Scope Name
RSED	Resource Edition	
RSEDDAT	Resource Edition Date	
RSID	Resource Identifier	Resource Identifier
RSKWDS	Resource Keyword Set	Resource Keywords
THUMB	Resource Thumbnail	
RSSRES	Resource Spatial Resolution	
RSRPTP	Resource Spatial Representation Type	
RSTOPIC	Resource Topic Category	Resource Topic Category Code

DMF Identifier	DMF Title	NMF (v3.0) Core Metadata Concept
RSDLOC	Resource Default Locale	Resource Language, Resource Character Set
RSTLOC	Resource Other Locale	
DGITYP	Geospatial Information Type	
RSGFLV	Resource Georeferencing Level	
RSPREF	Resource Representation Form	
RSTHEME	Resource Theme	
RSSERI	Name of Resource Series	
RSSHNA	Resource Sheet Name	
SRTYPE	Service Type	Service Type Name
SRTVER	Service Type Version	Service Type Version
SRSTD	Service Standard	
SRCPLING	Service Coupling Type	Coupled Resource Type
SROPRS	Resource Operated by the Service	
SROPER	Service Operation	Service Operation Name
SRCORS	Service Coupled Resource	Coupled Resource
RSEXT	Resource Extent	Resource Geographic Location, Resource Temporal Extent
RSRSYS	Resource Reference System	Resource Coordinate Reference System
RSDATE	Resource Reference Date	Resource Date
RSRPTY	Resource Responsible Party	Resource Point of Contact
RSSCST	Resource Security Constraint	Resource Classification, Resource Classification System
RSUSE	Resource Use Limitation	
RSLCST	Resource Legal Constraint	
RSLING	Resource Lineage	
RSDFMT	Resource Distribution Format	
RSONLLC	Resource Online Location	
		Resource Category (NMF Extension)

Table B-2 Example row in gpkg_metadata table for a GeoPackage DMF Metadata

id	md_scope	md_standard_uri	mime_type	metadata
1	dataset	DMF v2.0 https://dgiwg.org/std/dmf/2.0	text/xml	Complete DMF for entire GeoPackage (Figure B-1)

Table B-3 Example row in gpkg_metadata table for a GeoPackage NMIS / NMF Metadata

id	md_scope	md_standard_uri	mime_type	metadata
1	dataset	NMIS v3.0 https://nsgreg-api.nga.mil/schema/nas/8.0	text/xml	Complete NMIS for entire GeoPackage (Figure B-2)

Figure-B-1 provides an example XML instance document for DMF 2.0.

Figure B-1 - DMF v2.0 - Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!--.................................-->
<!-- DMF 2.0 sample for OSM vector Data in GeoPackage   -->
<!--       Encoding: ISO 19139                          -->
<!--         Date: 2019-10-2                            -->
<!--           Version : 1                              -->
<!--          Author: DGIWG                             -->
<!--.................................................. -->
<gmd:MD_Metadata xsi:schemaLocation="
urn:dgiwg:xmlns:dmf:2.0:iso-g1:profile:all www.dgiwg.org\xmlns\dmf\2.0\iso-
g1\profile\all\all.xsd" xmlns:all="http://www.dgiwg.org/xmlns/dmf/iso-
g1/all/1.0" xmlns:xs="http://www.w3.org/2001/XMLSchema" xmlns:xlink=
"http://www.w3.org/1999/xlink" xmlns:gco="http://www.isotc211.org/2005/gco"
xmlns:egco=" http://www.dgiwg.org/xmlns/dmf/iso-g1/egco/1.0" xmlns:gsr=
"http://www.isotc211.org/2005/gsr" xmlns:gss=
"http://www.isotc211.org/2005/gss" xmlns:gts=
"http://www.isotc211.org/2005/gts" xmlns:gmd=
"http://www.isotc211.org/2005/gmd" xmlns:gmx=
"http://www.isotc211.org/2005/gmx" xmlns:srv=
"http://www.isotc211.org/2005/srv" xmlns:gmi=
"http://www.isotc211.org/2005/gmi" xmlns:ngmp=
"urn:int:nato:geometoc:geo:metadata:ngmp:1.0" xmlns:xsi=
"http://www.w3.org/2001/XMLSchema-instance" xmlns:gml=
```

```
"http://www.opengis.net/gml/3.2" xmlns:sensor=
"http://www.opengis.net/spec/spec/sensorml/2.0/req/core">

        <!-- MDDLOC.language [1] -->
        <gmd:language>
            <gmd:LanguageCode codeList=
"http://www.isotc211.org/2005/resources/Codelist/ML_gmxCodelists.xml#Language
Code" codeListValue="eng">eng</gmd:LanguageCode>
        </gmd:language>
        <!-- MDDLOC.encoding [1] -->
        <gmd:characterSet>
            <gmd:MD_CharacterSetCode codeList=
"http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#MD_Characte
rSetCode" codeListValue="utf8">utf8</gmd:MD_CharacterSetCode>
        </gmd:characterSet>

    <!-- MDRPTY [1..*] -->
    <gmd:contact>
        <!-- MDRPTY.party [1] -->
        <gmd:CI_ResponsibleParty>
            <!-- MDRPTY.party.orgName [0..1] -->
            <gmd:organisationName>
                <gco:CharacterString>DGIWG</gco:CharacterString>
            </gmd:organisationName>
            <gmd:contactInfo>
                <gmd:CI_Contact>
                    <gmd:phone>
                        <gmd:CI_Telephone>
                            <!-- MDRPTY.party.phone [0..*] -->
                            <gmd:voice>
                                <gco:CharacterString>001 571-557-
5450</gco:CharacterString>
                            </gmd:voice>
                        </gmd:CI_Telephone>
                    </gmd:phone>
                    <gmd:address>
                    <!-- MDRPTY.party.address [0..*] -->
                        <gmd:CI_Address>
                            <gmd:deliveryPoint>
                                <gco:CharacterString>NGA Visitor Center, 7500
GEOINT Dr, Springfield, VA 22150</gco:CharacterString>
                            </gmd:deliveryPoint>
                            <!-- MDRPTY.party.city [0..1] -->
                            <gmd:city>
                                <gco:CharacterString>
```

```
Springfield</gco:CharacterString>
                            </gmd:city>
                            <!-- MDRPTY.party.postalCode [0..1] -->
                            <gmd:postalCode>
                                <gco:CharacterString>
22150</gco:CharacterString>
                            </gmd:postalCode>
                            <!-- MDRPTY.party.country [0..1] -->
                            <gmd:country>
                                <gco:CharacterString>United States
(USA)</gco:CharacterString>
                            </gmd:country>
                            <!-- MDRPTY.party.email [0..*] -->
                            <!-- <gmd:electronicMailAddress> -->
                                <!--
<gco:CharacterString>christopherstow594@mod.gov.uk</gco:CharacterString> -->
                            <!-- </gmd:electronicMailAddress> -->
                        </gmd:CI_Address>
                    </gmd:address>
                </gmd:CI_Contact>
            </gmd:contactInfo>
            <!-- MDRPTY.role [1] -->
            <gmd:role>
                <gmd:CI_RoleCode codeList=
"http://api.nsgreg.nga.mil/codelist/RoleCode" codeListValue="originator"
>Originator</gmd:CI_RoleCode>
            </gmd:role>
        </gmd:CI_ResponsibleParty>
    </gmd:contact>

        <!-- MDDATE [1] -->
        <gmd:dateStamp>
            <gco:Date>20201201T12:58:03Z</gco:Date>
        </gmd:dateStamp>

        <!-- MDSTD [1] -->
        <!-- MDSTD.title [1] -->
        <gmd:metadataStandardName>
            <gmx:Anchor>urn:dgiwg:metadata:dmf:2.0:profile:all</gmx:Anchor>
        </gmd:metadataStandardName>
        <!-- MDSTD.version [1] -->
        <gmd:metadataStandardVersion>
            <gco:CharacterString>2.0</gco:CharacterString>
        </gmd:metadataStandardVersion>
```

```
        <!-- RSTITLE [1] -->
        <gmd:title>
            <gco:CharacterString>Puerto Rico Sample</gco:CharacterString>
        </gmd:title>

        <!-- RSABSTR [1] -->
        <gmd:abstract>
            <gco:CharacterString>OSM Vector Dataset created composed of
waterbodies, roads and points of interest (POI) from OSM data downloaded from
Geofabrik.</gco:CharacterString>
        </gmd:abstract>

        <!-- RSPURP [1] -->
        <gmd:purpose>
        <gco:CharacterString>The purpose of this dataset is to provide a
sample dataset for GeoPackage development and use.</gco:CharacterString>
        </gmd:purpose>

        <!-- RSED [0..1] -->
        <gmd:edition>
            <gco:CharacterString>OSM</gco:CharacterString>
        </gmd:edition>

        <!-- RSEDDAT [0..1] -->
        <gmd:editionDate>
            <gco:Date>20201201</gco:Date>
        </gmd:editionDate>

        <!-- RSID [0..*] -->
        <gmd:identifier>
            <gmd:RS_Identifier>
                <!-- RSID.code [1] -->
                <gmd:code>
                    <gco:CharacterString>Geofabrik OSM</gco:CharacterString>
                </gmd:code>
            </gmd:RS_Identifier>
        </gmd:identifier>

        <!-- RSSRES [0..*] -->
        <gmd:spatialResolution>
            <gmd:MD_Resolution>
            <!-- RSSRES.equivalentScale [0..1] -->
                <gmd:equivalentScale>
                    <gmd:MD_RepresentativeFraction>
                        <gmd:denominator>
```

```
                            <gco:Integer>25000</gco:Integer>
                        </gmd:denominator>
                    </gmd:MD_RepresentativeFraction>
                </gmd:equivalentScale>
                <!--RSSRES.distance [0..1] not set -->
            </gmd:MD_Resolution>
        </gmd:spatialResolution>

        <!-- RSRPTP [0..1] -->
        <gmd:spatialRepresentationType>
            <gmd:MD_SpatialRepresentationTypeCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
RoleCode" codeListValue="vector">
Vector</gmd:MD_SpatialRepresentationTypeCode>
        </gmd:spatialRepresentationType>

        <!-- RSDLOC [1] -->
        <!-- RSDLOC.language [1] -->
        <gmd:language>
            <gmd:LanguageCode codeList=
"http://www.isotc211.org/2005/resources/Codelist/ML_gmxCodelists.xml#Language
Code" codeListValue="eng">eng</gmd:LanguageCode>
        </gmd:language>
        <!-- RSDLOC.encoding [1] -->
        <gmd:characterSet>
            <gmd:MD_CharacterSetCode codeList=
"http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#MD_Characte
rSetCode" codeListValue="utf8">utf8</gmd:MD_CharacterSetCode>
        </gmd:characterSet>

        <!-- DGITYP.keyword [1..*] -->
        <gmd:descriptiveKeywords>
            <gmd:MD_Keywords>
            <!-- DGITYP.keyword [1..*] -->
                <gmd:keyword>
                    <ngmp:NGMP_GeospatialInformationTypeCode codeList=
"http://www.dgiwg.org/resources/Codelist/dgiwgCodelists.xml#MD_GeospatialInfo
rmationTypeCode" codeListValue="vector2D">Vector
2D</ngmp:NGMP_GeospatialInformationTypeCode>
                </gmd:keyword>
                <!-- DGITYP.thesaurus [0..1] -->
                <gmd:thesaurusName>
                    <gmd:CI_Citation>
                        <!-- DGITYP.thesaurus.title [1] -->
                        <gmd:title>
```

```
<gco:CharacterString>NGMP_GeospatialInformationTypeCode</gco:CharacterString>
                        </gmd:title>
                        <!-- DGITYP.thesaurus.referenceDate [0..*] -->
                        <gmd:date>
                            <gmd:CI_Date>
                        <!-- DGITYP.thesaurus.referenceDate.date [1] -->
                                <gmd:date>
                                    <gco:Date>20110916</gco:Date>
                                </gmd:date>
                                <!-- DGITYP.thesaurus.referenceDate.type [1]
-->
                                <gmd:dateType>
                                    <gmd:CI_DateTypeCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
DateTypeCode" codeListValue="creation">Creation</gmd:CI_DateTypeCode>
                                </gmd:dateType>
                            </gmd:CI_Date>
                        </gmd:date>
                    </gmd:CI_Citation>
                </gmd:thesaurusName>
            </gmd:MD_Keywords>
        </gmd:descriptiveKeywords>

        <!-- RSGFLV [0..1] -->
        <gmd:descriptiveKeywords>
            <gmd:MD_Keywords>
            <!-- RSGFLV.keyword [1..*] -->
                <gmd:keyword>
                    <ngmp:NGMP_GeoreferencingLevelCode codeList=
"http://www.dgiwg.org/resources/Codelist/dgiwgCodelists.xml#MD_Geopositioning
LevelCode" codeListValue="georectified">
georectified</ngmp:NGMP_GeoreferencingLevelCode>
                </gmd:keyword>
                <!-- RSGFLV.thesaurus [0..1] -->
                <gmd:thesaurusName>
                    <gmd:CI_Citation>
                        <!-- RSGFLV.thesaurus.title [1] -->
                        <gmd:title>
                            <gco:CharacterString>
NGMP_GeoreferencingLevelCode</gco:CharacterString>
                        </gmd:title>
                        <!-- RSGFLV.thesaurus.referenceDate [0..*] -->
                        <gmd:date>
                            <gmd:CI_Date>
```

```
                                <!-- RSGFLV.thesaurus.referenceDate.date [1]
-->
                                <gmd:date>
                                    <gco:Date>20190125</gco:Date>
                                </gmd:date>
                                <!-- RSGFLV.thesaurus.referenceDate.type [1]
-->
                                <gmd:dateType>
                                    <gmd:CI_DateTypeCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
DateTypeCode" codeListValue="creation">Creation</gmd:CI_DateTypeCode>
                                </gmd:dateType>
                            </gmd:CI_Date>
                        </gmd:date>
                    </gmd:CI_Citation>
                </gmd:thesaurusName>
            </gmd:MD_Keywords>
        </gmd:descriptiveKeywords>

        <!-- RSTHEME [0..1] -->
        <gmd:descriptiveKeywords>
            <gmd:MD_Keywords>
                <!-- RSTHEME.keyword [1..*] -->
                <gmd:keyword>
                    <ngmp:NGMP_ThematicCode codeList=
"http://www.dgiwg.org/resources/Codelist/dgiwgCodelists.xml#RS_ThematicCode"
codeListValue="extraction">Extraction</ngmp:NGMP_ThematicCode>
                </gmd:keyword>
                <gmd:keyword>
                    <ngmp:NGMP_ThematicCode codeList=
"http://www.dgiwg.org/resources/Codelist/dgiwgCodelists.xml#RS_ThematicCode"
codeListValue="associatedSupportStruct">Associated Support
Structures</ngmp:NGMP_ThematicCode>
                </gmd:keyword>
                <!-- RSTHEME.thesaurus [0..1] -->
                <gmd:thesaurusName>
                    <gmd:CI_Citation>
                        <!-- RSTHEME.thesaurus.title [1] -->
                        <gmd:title>
                            <gco:CharacterString>
NGMP_ThematicCode</gco:CharacterString>
                        </gmd:title>
                        <!-- RSTHEME.thesaurus.referenceDate [0..*] -->
                        <gmd:date>
                            <gmd:CI_Date>
```

```
                                <!-- RSTHEME.thesaurus.referenceDate.date [1]
-->
                                <gmd:date>
                                    <gco:Date>20190125</gco:Date>
                                </gmd:date>
                                <!-- RSTHEME.thesaurus.referenceDate.type [1]
-->
                                <gmd:dateType>
                                    <gmd:CI_DateTypeCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
DateTypeCode" codeListValue="creation">Creation</gmd:CI_DateTypeCode>
                                </gmd:dateType>
                            </gmd:CI_Date>
                        </gmd:date>
                    </gmd:CI_Citation>
                </gmd:thesaurusName>
            </gmd:MD_Keywords>
        </gmd:descriptiveKeywords>

        <!-- RSEXT [0..*] -->
        <gmd:extent>
            <gmd:EX_Extent>
                <gmd:geographicElement>
                    <!-- RSEXT.boundingBox [0..*] -->
                    <gmd:EX_GeographicBoundingBox>
                        <!-- RSEXT.boundingBox.west [1] -->
                        <gmd:westBoundLongitude>
                            <gco:Decimal>-67.010729</gco:Decimal>
                        </gmd:westBoundLongitude>
                        <!-- RSEXT.boundingBox.east [1] -->
                        <gmd:eastBoundLongitude>
                            <gco:Decimal>-66.408509</gco:Decimal>
                        </gmd:eastBoundLongitude>
                        <!-- RSEXT.boundingBox.south [1] -->
                        <gmd:southBoundLatitude>
                            <gco:Decimal>18.408509</gco:Decimal>
                        </gmd:southBoundLatitude>
                        <!-- RSEXT.boundingBox.north [1] -->
                        <gmd:northBoundLatitude>
                            <gco:Decimal>18.500550</gco:Decimal>
                        </gmd:northBoundLatitude>
                    </gmd:EX_GeographicBoundingBox>
                </gmd:geographicElement>
            </gmd:EX_Extent>
        </gmd:extent>
```

```
        <!-- RSRSYS (horizontal) [0..*] -->
        <gmd:MD_ReferenceSystem>
            <gmd:referenceSystemIdentifier>
                <gmd:RS_Identifier>
                <!-- RSRSYS.code [1] -->
                    <gmd:code>

<gco:CharacterString>http://www.opengis.net/def/crs/EPSG/0/4326</gco:Characte
rString>
                    </gmd:code>
                    <!-- RSRSYS.namespace [0..1] -->
                    <gmd:codeSpace>
                        <gco:CharacterString>
//www.opengis.net/def/crs/EPSG</gco:CharacterString>
                    </gmd:codeSpace>
                    <!-- RSRSYS.description [0..1] not set -->
                </gmd:RS_Identifier>
            </gmd:referenceSystemIdentifier>
        </gmd:MD_ReferenceSystem>

            <!-- DMF/Core -->
            <!-- RSRSYS (vertical) [0..*] -->
        <gmd:referenceSystemInfo>
            <gmd:MD_ReferenceSystem>
                <gmd:referenceSystemIdentifier>
                    <gmd:RS_Identifier>
                    <!-- RSRSYS.code [1] -->
                        <gmd:code>
                            <gmx:Anchor xlink:href=
"http://www.opengis.net/def/crs/EPSG/0/5773">EGM96</gmx:Anchor>
                        </gmd:code>
                        <!-- RSRSYS.namespace [0..1] not set -->
                        <!-- RSRSYS.description [0..1] not set -->
                    </gmd:RS_Identifier>
                </gmd:referenceSystemIdentifier>
            </gmd:MD_ReferenceSystem>
        </gmd:referenceSystemInfo>

        <!-- DMF/Core -->
        <!-- RSDATE [1..*] -->
        <gmd:date>
            <gmd:CI_Date>
                <!-- RSDATE.date [1] -->
                <gmd:date>
```

```
                    <gco:Date>20201201</gco:Date>
                </gmd:date>
                <!-- RSDATE.type [1] -->
                <gmd:dateType>
                    <gmd:CI_DateTypeCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
DateTypeCode" codeListValue="creation">Creation</gmd:CI_DateTypeCode>
                </gmd:dateType>
            </gmd:CI_Date>
        </gmd:date>

        <!-- DMF/Core -->
        <!-- RSRPTY [0..*] -->
        <gmd:pointOfContact>
        <!-- RSRPTY.party [1] -->
            <gmd:CI_ResponsibleParty>
                <!-- RSRPTY.party.orgName [0..1] -->
                <gmd:organisationName>
                    <gco:CharacterString>"DGIWG"</gco:CharacterString>
                </gmd:organisationName>
                <gmd:contactInfo>
                    <gmd:CI_Contact>
                        <gmd:phone>
                            <gmd:CI_Telephone>
                                <!-- RSRPTY.party.phone [0..*] -->
                                <gmd:voice>
                                    <gco:CharacterString>See web site for
contact details</gco:CharacterString>
                                </gmd:voice>
                            </gmd:CI_Telephone>
                        </gmd:phone>
                    </gmd:CI_Contact>
                </gmd:contactInfo>
                <!-- RSRPTY.role [1] -->
                <gmd:role>
                    <gmd:CI_RoleCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
RoleCode" codeListValue="originator">Originator</gmd:CI_RoleCode>
                </gmd:role>
            </gmd:CI_ResponsibleParty>
        </gmd:pointOfContact>

        <!-- RSSCST [0..*] -->
        <gmd:resourceConstraints>
            <gmd:MD_SecurityConstraints>
```

```
                <!-- RSSCST.level [1] -->
                <gmd:classification>
                    <gmd:MD_ClassificationCode codeList=
"http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#MD_Classifi
cationCode" codeListValue="unclassified">
Unclassified</gmd:MD_ClassificationCode>
                </gmd:classification>
                <!-- RSSCST.system [0..1] -->
                <gmd:classificationSystem>
                    <gco:CharacterString>USA</gco:CharacterString>
                </gmd:classificationSystem>
            </gmd:MD_SecurityConstraints>
        </gmd:resourceConstraints>
        <gmd:resourceConstraints>

        <!-- DMF/Core -->
        <!-- RSUSE [0..*] -->
            <gmd:MD_Constraints>
                <gmd:useLimitation>
                    <gco:CharacterString>For evaluation
only</gco:CharacterString>
                </gmd:useLimitation>
            </gmd:MD_Constraints>
        </gmd:resourceConstraints>

        <!-- RSLCST [0..*] -->
        <gmd:resourceConstraints>
            <gmd:MD_LegalConstraints>
            <!-- RSLCST.statement [0..*] not set -->
                <!-- RSLCST.access [0..*] -->
                <gmd:accessConstraints>
                    <gmd:MD_RestrictionCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#MD_
RestrictionCode" codeListValue="restricted"/>
                </gmd:accessConstraints>
                <!-- RSLCST.use [0..*] -->
                <gmd:useConstraints>
                    <gmd:MD_RestrictionCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#MD_
RestrictionCode" codeListValue="copyright"/>
                </gmd:useConstraints>
                <!-- RSLCST.other [0..*] -->
                <gmd:otherConstraints>
                    <gco:CharacterString>DGIWG use only</gco:CharacterString>
                </gmd:otherConstraints>
```

```
            </gmd:MD_LegalConstraints>
        </gmd:resourceConstraints>

        <!-- RSLING [1] -->
            <gmd:lineage>
                <gmd:LI_Lineage>
                    <gmd:statement>
                        <gco:CharacterString>Features created by Open Street
Map open source community using various imagery sources and collection
methods.</gco:CharacterString>
                    </gmd:statement>
                    <gmd:processStep>
                        <gmd:LI_ProcessStep>
                        <!-- DMF/Common -->
                            <!-- RSPRST [0..*] -->
                            <!-- RSPRST.description [1] -->
                            <gmd:description>
                                <gco:CharacterString>Download and generate
data vector datasets in GeoPackage format</gco:CharacterString>
                            </gmd:description>
                            <!-- RSPRST.rationale [0..1] -->
                            <gmd:rationale>
                                <gco:CharacterString>Time
constraints</gco:CharacterString>
                            </gmd:rationale>
                            <!-- RSPRST.date [0..1] -->
                            <gmd:dateTime>
                                <gco:DateTime>2019-01-
25T12:58:03Z</gco:DateTime>
                            </gmd:dateTime>
                            <!-- RSPRST.processor [0..*] -->
                            <gmd:processor>
                            <!-- RSPRST.processor.party [1] -->
                                <gmd:CI_ResponsibleParty>
                                    <!-- RSPRST.processor.party.orgName
[0..1] -->
                                    <gmd:organisationName>
                                        <gco:CharacterString>
"USA"</gco:CharacterString>
                                    </gmd:organisationName>
                                    <!-- RSPRST.processor.party.phone [0..*]
-->
                                    <gmd:contactInfo>
                                        <gmd:CI_Contact>
                                            <gmd:phone>
```

```
                                                    <gmd:CI_Telephone>
                                                        <gmd:voice>

<gco:CharacterString>+001 571-557-5450</gco:CharacterString>
                                                        </gmd:voice>
                                                    </gmd:CI_Telephone>
                                                </gmd:phone>
                                            </gmd:CI_Contact>
                                        </gmd:contactInfo>
                                        <!-- RSPRST.processor.role [1] -->
                                        <gmd:role>
                                            <gmd:CI_RoleCode codeList=
"http://standards.iso.org/iso/19115/resources/Codelists/cat/codelists.xml#CI_
RoleCode" codeListValue="processor">Processor</gmd:CI_RoleCode>
                                        </gmd:role>
                                    </gmd:CI_ResponsibleParty>
                                </gmd:processor>
                            </gmd:LI_ProcessStep>
                        </gmd:processStep>
                    </gmd:LI_Lineage>
                </gmd:lineage>

            <!-- RSDFMT [1..*] -->
            <gmd:distributionInfo>
                <gmd:MD_Distribution>
                    <gmd:distributionFormat>
                    <!-- RSDFMT.citation.title [1] -->
                        <gmd:MD_Format>
                            <gmd:name>
                                <gco:CharacterString>
GeoPackage</gco:CharacterString>
                            </gmd:name>
                            <!-- RSDFMT.citation.version [0..1] -->
                            <gmd:version>
                                <gco:CharacterString>OGC</gco:CharacterString>
                            </gmd:version>
                            <!-- RSDFMT.decompression [0..1] -->
                            <gmd:fileDecompressionTechnique>
                                <gco:CharacterString>zip</gco:CharacterString>
                            </gmd:fileDecompressionTechnique>
                        </gmd:MD_Format>
                    </gmd:distributionFormat>
                </gmd:MD_Distribution>
            </gmd:distributionInfo>
```

```
        <!-- RSONLLC [0..*] -->
        <gmd:transferOptions>
            <gmd:MD_DigitalTransferOptions>
                <gmd:onLine>
                    <gmd:CI_OnlineResource>
                        <!-- RSONLLC.location [1] -->
                        <gmd:linkage>

<gmd:URL>https://www.arcgis.com/apps/webappviewer/index.html?id=10fc70d81a974
52fa232336637d69313</gmd:URL>
                        </gmd:linkage>
                        <!-- RSONLLC.function [0..1] -->
                        <gmd:function>
                            <gmd:CI_OnLineFunctionCode codeList=
"http://www.isotc211.org/2005/resources/Codelist/gmxCodelists.xml#CI_OnLineFu
nctionCode" codeListValue="download">Download</gmd:CI_OnLineFunctionCode>
                        </gmd:function>
                    </gmd:CI_OnlineResource>
                </gmd:onLine>
            </gmd:MD_DigitalTransferOptions>
        </gmd:transferOptions>
</gmd:MD_Metadata>
```

Figure-B-2 provides an example XML instance document for NMIS / NMF 3.0

Figure B-2 - NMIS / NMF v3.0 - Example 1

```
<?xml version="1.0" encoding="UTF-8"?>
<!--
**************************************************************************
**************************************************
* This is an example XML document that conforms to the NSG Application Schema
(NAS) under
* NSG Metadata Foundation (NMF) 3.0 Identification and Access Profile of the
NAS.  This example includes IC-ISM
* security markings and defines Resource Geographic Location by geographic
description with a temporal extent
* expressed as a period.
**************************************************************************
**************************************************-->
<nas:MD_Metadata xmlns:nas="https://nsgreg-api.nga.mil/schema/nas/8.0"
xmlns:cit="http://standards.iso.org/iso/19115/-3/cit/1.0"
xmlns:gco="http://standards.iso.org/iso/19115/-3/gco/1.0"
xmlns:gex="http://standards.iso.org/iso/19115/-3/gex/1.0"
```

```
xmlns:lan="http://standards.iso.org/iso/19115/-3/lan/1.0"
xmlns:mcc="http://standards.iso.org/iso/19115/-3/mcc/1.0"
xmlns:mco="http://standards.iso.org/iso/19115/-3/mco/1.0"
xmlns:mdb="http://standards.iso.org/iso/19115/-3/mdb/1.0"
xmlns:mri="http://standards.iso.org/iso/19115/-3/mri/1.0"
xmlns:mrs="http://standards.iso.org/iso/19115/-3/mrs/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:ism="urn:us:gov:ic:ism"
xmlns:ntk="urn:us:gov:ic:ntk"
xmlns:rr="urn:us:gov:ic:revrecall"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="https://nsgreg-api.nga.mil/schema/nas/8.0 ./nmis.xsd"
ism:DESVersion="13"
ism:ISMCATCESVersion="2"
ntk:DESVersion="10"
rr:DESVersion="1"
ism:resourceElement="true" ism:createDate="2016-05-30" ism:classification="U"
ism:ownerProducer="USA" ism:compliesWith="USGov">
    <mdb:defaultLocale>
        <lan:PT_Locale>
            <lan:language>
                <lan:LanguageCode codeList="https://nsgreg-
api.nga.mil/codelist/iso639-2" codeListValue="eng"/>
            </lan:language>
            <lan:country>
                <lan:CountryCode codeList="http://api.nsgreg.nga.mil/N2L?"
codeListValue="ge:GENC:3:3-3:USA"/>
            </lan:country>
            <lan:characterEncoding>
                <lan:MD_CharacterSetCode codeList="https://nsgreg-
api.nga.mil/codelist/IANACharset" codeListValue="UTF-8"/>
            </lan:characterEncoding>
        </lan:PT_Locale>
    </mdb:defaultLocale>
    <mdb:parentMetadata>
        <cit:CI_Citation>
            <cit:title>
                <gco:CharacterString>LTDS_39N078W_A-P</gco:CharacterString>
            </cit:title>
        </cit:CI_Citation>
    </mdb:parentMetadata>
    <mdb:metadataScope>
        <mdb:MD_MetadataScope>
            <mdb:resourceScope>
```

```
                <mcc:MD_ScopeCode codeList="https://nsgreg-
api.nga.mil/codelist/ScopeCode" codeListValue="dataset"/>
            </mdb:resourceScope>
            <mdb:name>
                <gco:CharacterString>cell</gco:CharacterString>
            </mdb:name>
        </mdb:MD_MetadataScope>
    </mdb:metadataScope>
    <mdb:contact xlink:href="#NgaOrganization"/>
    <mdb:dateInfo>
        <cit:CI_Date>
            <cit:date>
                <gco:DateTime>2016-05-30T09:30:47Z</gco:DateTime>
            </cit:date>
            <cit:dateType>
                <cit:CI_DateTypeCode codeList="https://nsgreg-
apih.nga.mil/codelist/DateTypeCode" codeListValue="creation"/>
            </cit:dateType>
        </cit:CI_Date>
    </mdb:dateInfo>
    <mdb:metadataStandard>
        <cit:CI_Citation>
            <cit:title>
                <gco:CharacterString>National System for Geospatial
Intelligence (NSG) Application Schema (NAS)</gco:CharacterString>
            </cit:title>
            <cit:alternateTitle>
                <gco:CharacterString>NAS</gco:CharacterString>
            </cit:alternateTitle>
            <cit:edition>
                <gco:CharacterString>8.0</gco:CharacterString>
            </cit:edition>
        </cit:CI_Citation>
    </mdb:metadataStandard>
    <mdb:identificationInfo>
        <nas:DataIdentification>
            <mri:citation>
                <cit:CI_Citation>
                    <cit:title>
                        <gco:CharacterString>Local Topographic Data Store
(LTDS) data sample for cell 39N078WA.</gco:CharacterString>
                    </cit:title>
                    <cit:date>
                        <cit:CI_Date>
                            <cit:date>
```

```
                                <gco:DateTime>2016-05-
30T09:30:47Z</gco:DateTime>
                            </cit:date>
                            <cit:dateType>
                                <cit:CI_DateTypeCode codeList=
"https://nsgreg-api.nga.mil/codelist/DateTypeCode" codeListValue=
"distribution"/>
                            </cit:dateType>
                        </cit:CI_Date>
                    </cit:date>
                    <cit:identifier>
                        <mcc:MD_Identifier>
                            <mcc:authority>
                                <cit:CI_Citation>
                                    <cit:title>
                                        <gco:CharacterString>NGA/S2
Topographic Feature Data Management (TFDM) system.</gco:CharacterString>
                                    </cit:title>
                                    <cit:date>
                                        <cit:CI_Date>
                                            <cit:date>
                                                <gco:DateTime>2016-04-
15T09:30:47Z</gco:DateTime>
                                            </cit:date>
                                            <cit:dateType>
                                                <cit:CI_DateTypeCode
codeList="https://nsgreg-api.nga.mil/codelist/DateTypeCode" codeListValue=
"creation"/>
                                            </cit:dateType>
                                        </cit:CI_Date>
                                    </cit:date>
                                </cit:CI_Citation>
                            </mcc:authority>
                            <mcc:code>
                                <gco:CharacterString>urn:uuid:65ab5357-d6cf-
4b25-a73a-8d76ed6aj367</gco:CharacterString>
                            </mcc:code>
                        </mcc:MD_Identifier>
                    </cit:identifier>
                    <cit:series>
                        <cit:CI_Series>
                            <cit:name>
                                <gco:CharacterString>Local Topographic Data
Store (LTDS) data samples for cell 39N078W.</gco:CharacterString>
                            </cit:name>
```

```
                        </cit:CI_Series>
                    </cit:series>
                </cit:CI_Citation>
            </mri:citation>
            <mri:abstract>
                <gco:CharacterString>A small-footprint sample of NAS-
conformant data located in the southern region of New Jersey, USA. It
contains a variety of urban, terrain, shoreline, and littoral vector-based
feature data, some of which are vertical obstructions to
aircraft.</gco:CharacterString>
            </mri:abstract>
            <mri:pointOfContact>
                <cit:CI_Responsibility>
                    <cit:role>
                        <cit:CI_RoleCode codeList="https://nsgreg-
api.nga.mil/codelist/RoleCode" codeListValue="originator"/>
                    </cit:role>
                    <cit:party>
                        <cit:CI_Organisation>
                            <cit:name>
                                <gco:CharacterString>US National Geospatial-
Intelligence Agency</gco:CharacterString>
                            </cit:name>
                        </cit:CI_Organisation>
                    </cit:party>
                </cit:CI_Responsibility>
            </mri:pointOfContact>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>
boundaries</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>
intelligenceMilitary</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>
inlandWaters</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>location</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>oceans</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
```

```
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>
structure</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>
transportation</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:topicCategory>
                <mri:MD_TopicCategoryCode>
utilitiesCommunication</mri:MD_TopicCategoryCode>
            </mri:topicCategory>
            <mri:extent>
                <gex:EX_Extent>
                    <gex:geographicElement>
                        <gex:EX_GeographicDescription>
                            <gex:geographicIdentifier>
                                <mcc:MD_Identifier>
                                    <mcc:authority xlink:href=
"#NewJerseyGazetteer"/>
                                    <mcc:code>
                                        <gco:CharacterString>Southern New
Jersey</gco:CharacterString>
                                    </mcc:code>
                                </mcc:MD_Identifier>
                            </gex:geographicIdentifier>
                        </gex:EX_GeographicDescription>
                    </gex:geographicElement>
                    <gex:temporalElement>
                        <gex:EX_TemporalExtent>
                            <gex:extent>
                                <gml:TimePeriod gml:id=
"ResourceTemporalExtent">
                                    <gml:begin>
                                        <gml:TimeInstant gml:id=
"ResourceTemporalExtentBegin">
                                            <gml:timePosition
indeterminatePosition="unknown"/>
                                        </gml:TimeInstant>
                                    </gml:begin>
                                    <gml:end>
                                        <gml:TimeInstant gml:id=
"ResourceTemporalExtentEnd">
                                            <gml:timePosition>2010-01-
01</gml:timePosition>
```

```
                                        </gml:TimeInstant>
                                    </gml:end>
                                </gml:TimePeriod>
                            </gex:extent>
                        </gex:EX_TemporalExtent>
                    </gex:temporalElement>
                </gex:EX_Extent>
            </mri:extent>
            <mri:descriptiveKeywords>
                <mri:MD_Keywords>
                    <mri:keyword>
                        <gco:CharacterString>Boundaries and
Lines</gco:CharacterString>
                    </mri:keyword>
                    <mri:keyword>
                        <gco:CharacterString>General
Structures</gco:CharacterString>
                    </mri:keyword>
                </mri:MD_Keywords>
            </mri:descriptiveKeywords>
            <mri:descriptiveKeywords>
                <mri:MD_Keywords>
                    <mri:keyword>
                        <gco:CharacterString>Woeful Weeping
Willow</gco:CharacterString>
                    </mri:keyword>
                    <mri:keyword>
                        <gco:CharacterString>Conspicuously Cheerful
Charthouse</gco:CharacterString>
                    </mri:keyword>
                </mri:MD_Keywords>
            </mri:descriptiveKeywords>
            <mri:resourceConstraints>
                <mco:MD_SecurityConstraints>
                    <mco:classification>
                        <mco:MD_ClassificationCode codeList="https://nsgreg-
api.nga.mil/codelist/ClassificationCode" codeListValue="unclassified"/>
                    </mco:classification>
                    <mco:classificationSystem>
                        <gco:CharacterString>USA</gco:CharacterString>
                    </mco:classificationSystem>
                </mco:MD_SecurityConstraints>
            </mri:resourceConstraints>
            <mri:defaultLocale>
                <lan:PT_Locale>
```

```
                    <lan:language>
                        <lan:LanguageCode codeList="https://nsgreg-
api.nga.mil/codelist/ISO639-2" codeListValue="eng"/>
                    </lan:language>
                    <lan:country>
                        <lan:CountryCode codeList=
"http://api.nsgreg.nga.mil/N2L?" codeListValue="ge:GENC:3:3-3:USA"/>
                    </lan:country>
                    <lan:characterEncoding>
                        <lan:MD_CharacterSetCode codeList="https://nsgreg-
api.nga.mil/codelist/IANACharset" codeListValue="UTF-8"/>
                    </lan:characterEncoding>
                </lan:PT_Locale>
            </mri:defaultLocale>
            <nas:resourceCategory codeSpace="https://nsgreg-
api.nga.mil/codelist/ResourceCategoryCode">topographicFeatures</nas:resourceC
ategory>
        </nas:DataIdentification>
    </mdb:identificationInfo>
</nas:MD_Metadata>
```

Annex C GeoPackage to EPSG Registry Crosswalk

This Annex provides a crosswalk between the Coordinate Reference Systems (CRS) defined in Clause 7.1 Spatial Reference Systems and the corresponding CRS identifiers in the EPSG Registry.

Table C-1: GeoPackage to EPSG CRS Crosswalk

CRS Name	CRS AUTH ID	EPSG Source
EGM2008 height Vertical	EPSG:3855	https://epsg.org/crs_3855/EGM2008-height.html [EPSG:3855]
Lambert Conic Conformal (1SP) coordinate operation method	EPSG:9801	https://epsg.org/coord-operation-method_9801/Lambert-Conic-Conformal-1SP.html [EPSG:9801]
Lambert Conic Conformal (2SP) coordinate operation method	EPSG:9802	https://epsg.org/coord-operation-method_9802/Lambert-Conic-Conformal-2SP.html [EPSG:9802]
WGS 84 Geographic 2D lat/lon	EPSG:4326	https://epsg.org/crs_4326/WGS-84.html [EPSG:4326]
WGS 84 Geographic 3D lat/lon/hae	EPSG:4979	https://epsg.org/crs_4979/WGS-84.html [EPSG:4979]
WGS 84 UTM Zone 1N	EPSG:32601	https://epsg.org/crs_32601/WGS-84-UTM-zone-1N.html [EPSG:32601]
WGS 84 UTM Zone 60N	EPSG:32660	https://epsg.org/crs_32660/WGS-84-UTM-zone-60N.html [EPSG:32660]
WGS 84 / Pseudo Mercator	EPSG:3857	https://epsg.org/crs_3857/WGS-84-Pseudo-Mercator.html [EPSG:3857]
WGS 84 / UPS North (E,N)	EPSG:5041	https://epsg.org/crs_5041/WGS-84-UPS-North-E-N.html [EPSG:5041]
WGS 84 / UPS South (E,N)	EPSG:5042	https://epsg.org/crs_5042/WGS-84-UPS-South-E-N.html [EPSG:5042]
WGS 84 / World Mercator	EPSG:3395	https://epsg.org/crs_3395/WGS-84-World-Mercator.html [EPSG:3395]
EGM2008 height	EPSG:3855	https://epsg.org/crs_.html [EPSG:3855], https://epsg.org/crs_3855.html [EPSG:3855]
EGM96 height	EPSG:5733	https://epsg.org/crs_.html [EPSG:5733], https://epsg.org/crs_5733.html [EPSG:5733]

CRS Name	CRS AUTH ID	EPSG Source
EGM84 height	EPSG:5798	https://epsg.org/crs_.html [EPSG:57098], https://epsg.org/crs_5793.html [EPSG:5798]
WGS 84 + EGM2008 height Compound	EPSG:9518, EPSG:3855	https://epsg.org/crs_9518/WGS-84-EGM2008-height.html [EPSG:9518], https://epsg.org/crs_3855/EGM2008-height.html [EPSG:3855]
WGS 84 Geographic 2D (G730}	EPSG:9053	https://epsg.org/crs_9053/WGS-84-G730.html [EPSG:9053]
WGS 84 Geographic 2D (G873}	EPSG:9054	https://epsg.org/crs_9054/WGS-84-G873.html [EPSG:9054]
WGS 84 Geographic 2D (G1150}	EPSG:9055	https://epsg.org/crs_9055/WGS-84-G1150.html [EPSG:9055]
WGS 84 Geographic 2D (G1674}	EPSG:9056	https://epsg.org/crs_9056/WGS-84-G1674.html [EPSG:9056]
WGS 84 Geographic 2D (G1762}	EPSG:9057	https://epsg.org/crs_9057/WGS-84-G1162.html [EPSG:9057]
WGS 84 Geographic 2D (G2139}	EPSG:9755	https://epsg.org/crs_9755/WGS-84-G2139.html [EPSG:9755]
WGS 84 Geographic 3D (G730}	EPSG:7657	https://epsg.org/crs_7657/WGS-84-G730.html [EPSG:7657]
WGS 84 Geographic 3D (G873}	EPSG:7659	https://epsg.org/crs_7659/WGS-84-G873.html [EPSG:7659]
WGS 84 Geographic 3D (G1150}	EPSG:7661	https://epsg.org/crs_7661/WGS-84-G1150.html [EPSG:7661]
WGS 84 Geographic 3D (G1674}	EPSG:7663	https://epsg.org/crs_7663/WGS-84-G1674.html [EPSG:7663]
WGS 84 Geographic 3D (G1762}	EPSG:7665	https://epsg.org/crs_7665/WGS-84-G1162.html [EPSG:7665]
WGS 84 Geographic 3D (G2139}	EPSG:9754	https://epsg.org/crs_9754/WGS-84-G2139.html [EPSG:9754]

Annex D: GeoPackage Use Cases.

D-1: Generic GeoPackage design considerations

Intended use

The spatial extent, vector and raster content, use of extensions, CRS, and metadata of a GeoPackage will generally be based on the intended use and the existing capabilities of the system(s) that will use the GeoPackage. The created GeoPackages could vary greatly in size from containing large area coverage of vector data or tiles to a small, purpose-built area limited to necessary data. The GeoPackage may vary in complexity from a simple container to store and exchange vector data to a tailored self-contained bundle providing data, tile, extensions, and additional tables to fully enable a targeted system or capability. This annex outlines four use cases for GeoPackage and specific steps and considerations for each.

Coordinate reference system (CRS) (see section 7.1).

1. When the GeoPackage is intended primarily for storage and transmission of vector data the geographic CRS EPSG 4326 is preferred to the other projected CRS as it allows the client to select any projected CRS available on the system.
2. When the GeoPackage is intended for storage and transmission of tile raster maps and images, a projected CRS appropriate for the area of interest should be selected. This will most often be EPSG 3395 World Mercator with Universal Polar Stereographic used at high latitudes.
3. When both vector and raster content will be included in the GeoPackage, the requestor or producer will need to determine if an unprojected geographic CRS is preferred to provide more projections options or if a specific projected CRS is preferred to direct a suitable solution.

Vector data

1. Data Preparation: Geospatial vector data is normally segmented into layers; consideration should be given to what layers and attributes are needed. Feature layers may be filtered prior to creating a GeoPackage to improve performance and reduce size. Most client applications can filter by layer but filtering by attribute is limited, so pre-filtering layers can make the data more useful to an end-user.
2. Format and Schema: Vector datasets are often stored in formats other than SQL such as shapefiles and file geodatabase. The datasets will need to be translated. Tools are available to support this (e.g. GDAL OGR2OGR). Both native and translated SQL databases may contain text descriptors or coded values (e.g. FCODEs) for the features and attributes. When a richer description of the data is desired to make it more useful to non-GIS experts, the use of the GeoPackage Schema extension can provide richer information on the dataset. Building tables for this extension is a manual, upfront requirement but can be reused with

other GeoPackages using different datasets with the same schema.

3. Styles: Vector GeoPackages may be created with or without styles and symbology resources included. The community-style extension is still an evolving document that requires detailed up-front effort and should be used primarily when both provider and consumer have validated its performance. The extension identifies processes for semantic annotation tables to link vector data to stylesheets and associated symbology as separate tables. Stylesheets may be defined in both open standard and proprietary, vendor-unique encodings. This semantic annotation can be created for multiple stylesheets and encodings. Style requirements and design consideration (beyond GeoPackage scope) include providing scale-based rules for handling multi-resolution vector data and managing symbology files and formats as separate resources from styles to allow reuse and flexibility, depending on the use case.

a. A single stylesheet may be preferred to:
    - Ensure desired visualization is enforced (e.g. geospatial foundation for Common Operational Picture (COP) style).
    - Provide tailored GeoPackages based on client-unique rendering capabilities.

b. Multiple stylesheets and encodings may be preferred to:
    - Provide maximum encoding options for diverse clients.
    - Provide sets of compatible styles for different visualization needs (e.g. day and night modes)
    - Streamline storage and production within a central archive.

c. No style may be preferred to:
    - Keep resources (data, styles, symbology) separate with a web service.

1. Other extensions:

a. R-tree spatial index: This extension adds a new capability for spatially indexing geometry columns. The spatial index is established by creating a virtual table and a set of triggers. This index may be applied to multi-dimensional information such as geographical coordinates, rectangles, or polygons.

b. Vector tiles: Vector data may be provided as tiles using the Vector Tiles and Vector Tiles Attribute extensions. The tiles are prepared prior to packaging. Schema, styles, and R-tree may still be desired. Considerations for raster tiles may also apply.

Raster images tiles and tiles

1. Zoom level(s):

Raster tiles are defined by bounding box, CRS, and zoom levels. OGC publishes Tile Matrix Sets for common CRS, including all recommended for DGIWG profiles. Zoom levels start at a global scale and then add zoom levels at a higher resolution for pre-defined scales or by applying a process of quadtree or powers of "2". All DGIWG profiles use quadtrees as this provides better computational and visual performance. Increasing one zoom level replacing one tile at the lower zoom level with four at the next higher level, double the resolution but quadrupling the volume.

2. tileSets:

A tileSet can consist of a single map or image which can be displayed at a fixed scale or interpolated or extrapolated to multiple scales with distortion occurring as the tile moves further from the native resolution. Alternatively, a tileSet could be composed of multiple maps or images aligned with the native resolutions to minimize distortion. Even then, problems may arise at very large scales (e.g. 1: 500) where no available map or image is suitable at the resolution.

Gridded coverage data

When gridded coverage data is used in a GeoPackage, the OGC GeoPackage extension for Gridded Coverage Data should be used. Originally intended to include elevation data such as DGIWG Gridded Elevation Data (DGED) in a GeoPackage, it is also applicable to other coverage information to include weather and environmental observations. Coverage data may be provided as a single file (e.g. Digital Terrain Elevation Data (DTED) level 2 to accompany imagery tiles) or as a tile matrix set (e.g. multi-resolution DGED aligned to increase accuracy at higher zoom levels/closer ground sample distance).

Related data

The OGC GeoPackage extension for Related Tables allows users to associate aspatial information with either vector data or raster tiles. Related Tables allow inputs such as hand-held photos, sketches, or audio files to be associated with a feature or location on the dataset or tile.

Metadata

To allow users to fully leverage GeoPackages, proper metadata is required. This benefits consumers to determine which GeoPackages in a central storage facility are most appropriate for use. It also allows users to examine the contents of mission-specific GeoPackages to understand their contents including source data and extensions provided. The DGIWG profile establishes metadata requirements to comply with the DGIWG as well as the use of national profiles or both within the same GeoPackage.

D-2: Use Case 1: Disadvantaged, mobile, and autonomous users

1. Use GeoPackage as a data dissemination format for background maps in mobile / autonomous systems.
2. Mission applications: Supports packaging data configured to mission requirements and system constraints in DDIL environments where mobile systems will preload required content prior to mission execution as web services and online updates will not be reliably available. File size, transmission capacity, and system constraints of the consumer will impact the amount of data provided and the extensions used.
3. Consumer or producer defines requirement:
    a. Single CRS selected based on area and data available.
    b. Vector data for the area identified and prepared for GeoPackage.
        - Include optional schema extension if consumer plain language query or data analytics capabilities are desired.
        - Include optional style extension if desired to share common styles or provide specific encodings for the client.
        - Include optional R-tree extension if improved organization and management of vector data is beneficial to accommodate dataset size.
        - If vector data will be provided as vector tiles, use of the tiling process is also required.
        - The SQLite database of the GeoPackage supports the creation of a non-spatial index such as a feature attribute column to accelerate query response.
    c. Raster maps and/or images identified and prepared for GeoPackage.
        - Determine source inputs and required zoom levels.
    d. Gridded coverage data: Include optional extension if elevation data is required. Select appropriate DGED level based on mission need and capabilities of receiving system(s).
    e. Related data may be available and desired.
    f. Comply with metadata requirements: DGIWG (DMF), national (e.g. NMF), and/or others.
        i. If national metadata exists and DMF does not exist, transform national metadata into equivalent DMF.
        ii. Insert DMF into GeoPackage for the entire GeoPackage and levels of detail (vector layers, tilematrixsets, etc.) that describe the contents of the GeoPackage.
    g. Produce GeoPackage (see use case D-6)
    h. Validate GeoPackage

i. OGC CITE Team Engine GeoPackage test, including those for DGIWG profiles, when developed, will be used to validate conformance. Validation should be conducted on initial production and periodically revalidated.

4. Disseminate to the consumer using available online or offline transfer methods.

D-3: Use Case 2: Data exchange between processing sites

1. Use GeoPackage as an exchange format for raw data between processing sites.
2. Mission applications: Supports using GeoPackage as efficient storage and transport container for large volumes of data between network-connected processing centers. File size and bandwidth constraints are of less importance.
3. The Processing center defines requirements:
    a. Area of interest content provided.
    b. Content may be configured and segmented by data type (vector datasets, maps, images, elevation data, etc.) or may be configured by operational area (national or unit bounding box) or both.
    c. One or more CRS selected based on area and data available.
    d. Vector data for the area identified and prepared for GeoPackage.
        - Include optional schema extension if consumer plain language query or data analytics capabilities are desired and not available at receiving processing center(s).
        - Include optional style extension if desired to share common styles and not available at receiving processing center(s).
        - Include optional R-tree extension if improved organization and management of vector data is beneficial to accommodate dataset size or structure.
        - If vector data will be provided as vector tiles, use of the tiling process is also required.
        - The SQLite database of the GeoPackage supports the creation of a non-spatial index such as a feature attribute column to accelerate query response.
    e. Raster maps and/or images identified and prepared for GeoPackage
        - Determine source inputs and required zoom levels.
    f. Gridded coverage data: Include optional extension if elevation data is required. Select appropriate DGED level based on mission need and capabilities of receiving system(s).
    g. Related data may be provided if they already exist but will not normally be developed for data exchanges.

h. Comply with metadata requirements: DGIWG (DMF), national (e.g. NMF), and/or other.

   i. If national metadata exists and DMF does not exist, transform national metadata into equivalent DMF.

   ii. Insert DMF into GeoPackage for the entire GeoPackage and levels of detail (vector layers, tilematrixsets, etc.) that describe the contents of the GeoPackage.

4. Produce GeoPackage for exchange with another processing site (see use case D-6).

5. Validate GeoPackage

   a. OGC CITE Team Engine GeoPackage test, including those for DGIWG profiles when developed, will be used to validate conformance. Validation should be conducted on initial production and periodically revalidated.

6 Disseminate using available online transfer methods.

D-4: Use Case 3: Storage for processing systems and map servers

1. Use GeoPackage as a storage format in a processing system or Web Map server.

2. Mission applications: Provides a standard format for raster and vector content offered by a server that facilitates access and retrieval when indexed, allows for managed updates to selected content, and supports secure import/export. File size and bandwidth constraints are of some importance.

3. Service provider designs content sharing architecture.

   a. Area of interest content provided.

   b. Content may be configured and segmented by data type (vector datasets, maps, images, elevation data, etc.) or may be configured by operational area (national or unit bounding box) or both.

   c. One or more CRS selected based on area and data available.

   d. Vector data for the area identified and prepared for GeoPackage.

      - Include optional schema extension if consumer plain language query or data analytics capabilities are desired.

      - Include optional style extension if desired to share common styles or provide encodings options for multiple clients.

      - Include optional R-tree extension if improved organization and management of vector data is beneficial to accommodate dataset size or structure.

      - If vector data will be provided as vector tiles, use of the tiling process is also required.

e. Raster maps and/or images identified and prepared for GeoPackage.
    - Determine source inputs and required zoom levels.
f. Gridded coverage data: Include optional extension if elevation data is required. Select appropriate DGED level based on mission need and capabilities of receiving system(s).
g. Related data may be provided if they already exist but will not normally be developed on demand for web services.
h. Define metadata requirements: DGIWG, national, and/or others.

4. Produce GeoPackage (see section D-6)
5. Validate GeoPackage
    a. OGC CITE Team Engine GeoPackage test, including those for DGIWG profiles when developed, will be used to validate conformance. Validation should be conducted on initial production and periodically revalidated.
6. Disseminate to consumers using available online transfer methods.

D-5: Use Case 4: Support autonomous data collection and retrieval

1. Use GeoPackage as an exchange format for collection and retrieval tasks using mobile or handheld devices.
2. Mission applications: Supports packaging of data from field collections such as reconnaissance and engineering surveys in formats to facilitate delivery to tasking headquarters for analysis and integration into more robust data collections.
3. Deploying headquarters producers GeoPackage to support designated operation
    a. See Use Case 1 (D-2) for production considerations.
    b. See D-6 for production workflow.
4. Deployed element gathers information in executing the mission. Information could include:
    a. Observations documented as vector data identifying key elements (Size, Activity, Location, Unit, Time, Equipment, etc.). Observations may be plotted as Points of Interest (POI) and/or entered into a field-expedient or formatted database. Plotted observations may be symbolized according to the style available to the device.
    b. Hand-held imagery or scanned graphics of the mission area. Where possible, this imagery should be tagged to the location and time it was taken and related to relevant vector observations.
    c. Vector datasets and raster products provided from liaisons with friendly or neutral entities during the mission.
    d. Precision observations and measurements of the area as coverage datasets (elevation,

weather observations, hydrography data, chemical reconnaissance)

5. The collected information is prepared as a GeoPackage for transfer to analysts at mission headquarters.
    a. Core information will normally be preset to comply with GeoPackage profile requirements. Options to change core elements may be provided.
    b. Vector data will include data and metadata tables and may include schema, style, and related extension tables if enabled and used during collection.
    c. Geospatially referenced raster products will be included as individual tables or blobs or as tiles depending on the volume and format in which they were provided.
    d. Other information including handheld imagery and scanned graphics that are included as Related Tables linked to vector or raster tables as appropriate.
    e. Coverage data will use the gridded coverage extension.
6. Provide collected data back to tasking headquarters. Transfer GeoPackage for detailed analysis and integration into proper data holdings.

D-6: Generalized Workflow for generating GeoPackages

The process to produce a DGWIG compliant GeoPackage requires adding feature layers with the vector data, adding the imagery / raster data in tile matrix sets, and adding metadata for the entire GeoPackage and specific vector and / or imagery content.

The two principal workflows for GeoPackage production are described, with a section on metadata that is applicable to both workflows.

Workflow for generating a Vector GeoPackage

An end-user needs a selection of vector data layers of an area of operations; the end-user, or supporting Geo subject matter expert (SME) with access to the export utility zooms selects zooms to the desired area and selects the vector layers needed. The utility executes the ogr2ogr function and accesses the database and generates a vector GeoPackage, the user is notified when the process is complete and can download the data to the appropriate file transfer mechanism. This functionality provides an indexed vector GeoPackage with all associated attributes in their coded domain form.

This process describes the basic vector GeoPackage. If extensions for schema, styles, metadata, Related Tables, and R-tree are used, they will need to be manually developed and inserted into the process flow.

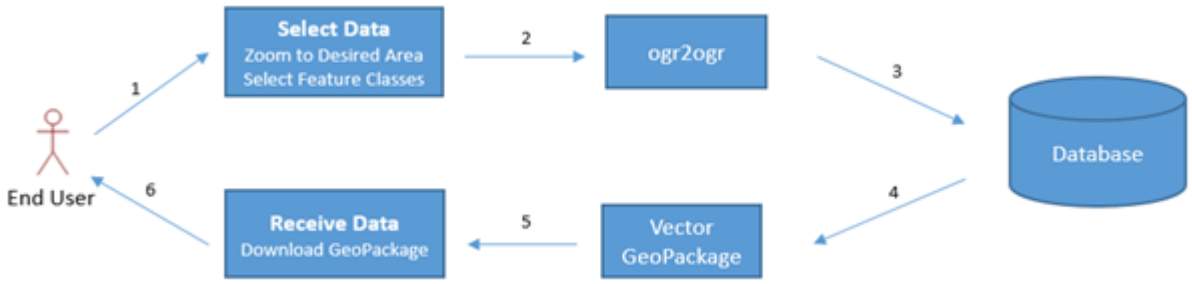

Figure D-1: Vector GeoPackage Workflow

Workflow for generating a Raster Tile GeoPackage

An end-user needs raster map products at a variety of scales over an area of operations, and an end-user or supporting Geo SME with access to the export utility zooms to the area of operation, selects the map products needed, output format, and zoom levels needed. The utility uses these Geospatial Data Abstraction Library (GDAL) functions: gdalbuildvrt to build a mosaic, gdal_translate to create the GeoPackage, and "gdaladdo" to create additional zoom levels. The gdalinfo function is used to access information about the GeoPackage. Once the GeoPackage is created, a notification is sent to the user so the user can download the data to the appropriate file transfer mechanism. The system may also provide presets so data sets may be tailored to particular operational or system requirements.

This process describes the basic raster tile GeoPackage. If extensions for metadata, gridded coverage data, and Related Tables are used, they will need to be manually developed and inserted into the process flow.

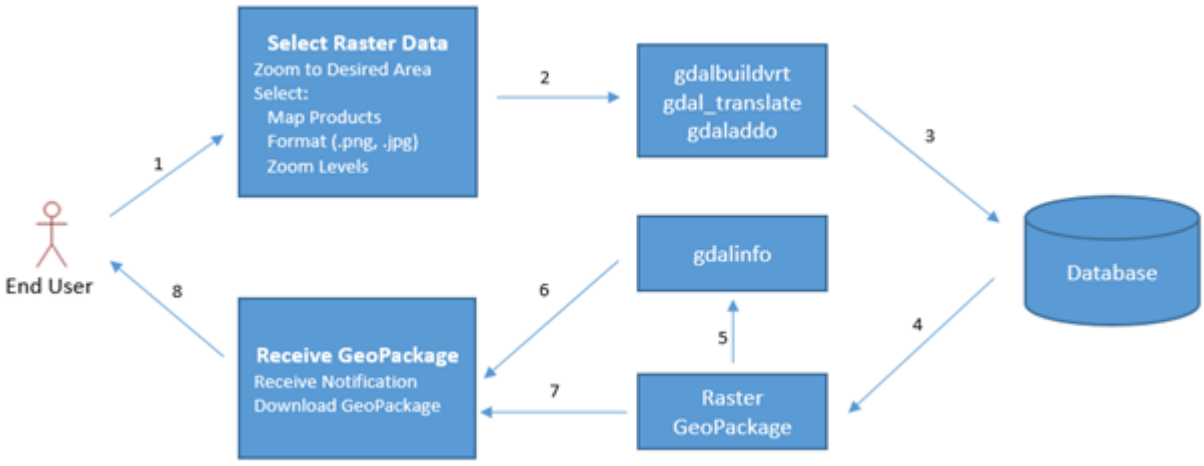

Figure D-2: Raster GeoPackage Workflow

Workflow for Metadata in GeoPackage

For DGIWG exchange, the vector and raster workflows for GeoPackage must supply at least one instance of DGIWG Metadata Foundation (DMF) for the entire GeoPackage. Users may utilize whatever tools are available for metadata population and update of the GeoPackage metadata

tables. The use cases for the metadata workflow are depicted in Figure-D-3. Since the DGIWG profile is a multi-national implementation of an OGC standard, when a nation produces raster and/or vector data in a GeoPackage to be exchanged with other DGIWG nations (e.g. NATO members), the GeoPackage would contain national metadata in the original product as required by the nation's GEOINT standards. Before the nation's GeoPackage products are exchanged with other nations, the GeoPackage producer can translate the national metadata into the corresponding DGIWG metadata format (DMF). The producer may choose to retain the GeoPackage national metadata and append the DMF content to the GeoPackage in rows of the GeoPackage metadata tables. The DMF metadata in the GeoPackage is understandable by all applications using the DGIWG profile. In addition, the systems of the producer nation and other DGIWG members have the option to view and use the national metadata information in addition to the DMF. Should the GeoPackage producer update the GeoPackage content and national metadata in the GeoPackage, a repeatable translation to DMF is performed before the exchange of the next version of the GeoPackage file with DGIWG.

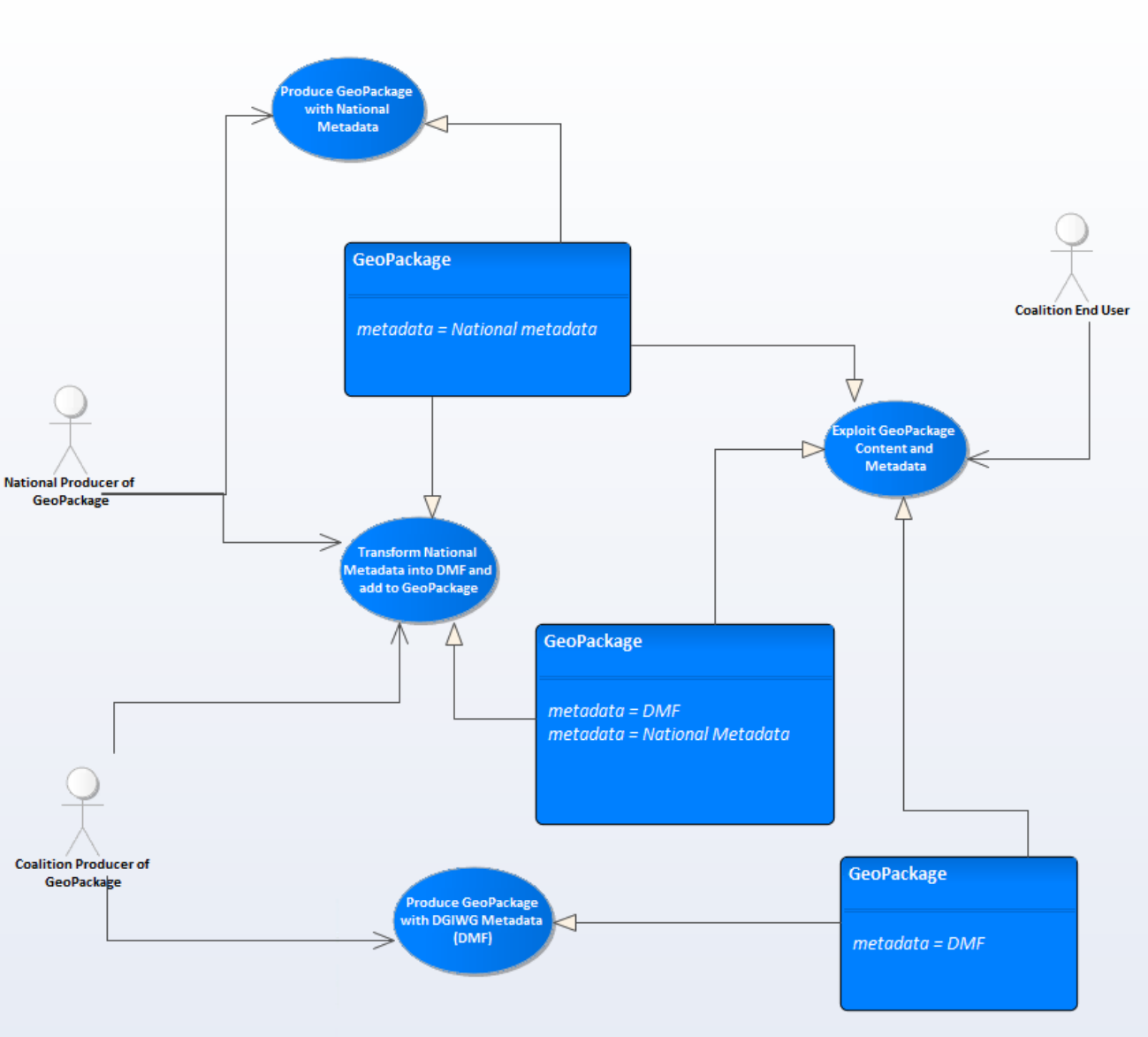

Figure D-3: GeoPackage Metadata Use Case

Annex E Footnotes and Inline References

This section contains footnotes and in-line document references from other sections of the document. Footnotes are annotated [#] in the hyperlink to this section.

[1] Per OGC Two Dimensional Tile Matrix Set, the coordindate order for the CRS84 http://www.opengis.net/def/crs/OGC/1.3/CRS84 is lon, lat order and for EPSG:4326 http://www.opengis.net/def/crs/EPSG/0/4326 is lat, lon (Note 4 in section D.2 of OGC 17-083r2).

[2] "WGS 84" is the srs_name specified by EPSG, which is ambiguous because it is also used for 4326. Some experts in Oil and Gas Producers (OGP) for the EPSG state that WGS 84 is fundamentally a geocentric Cartesian (X,Y,Z) system (EPSG:4978). From that, geographic 3D coordinates (including HAE) may be derived (EPSG:4979). If you drop the HAE you end up with geographic 2D (EPSG:4326).

[3] EPSG defines this as a geographic 3D CRS. This definition is encoded per WKT2 / ISO 19162. The WKT definition used here is one for a geocentric CRS with the same datum, ellipsoid, and axes. Note: the unspecified unit of measure for ellipsoidal height is meters.

[4] "WGS 84" is the srs_name specified by EPSG, which is ambiguous because it is also used for 4979. This CRS is also known as Plate Carree, Cylindrical Equirectangular, Simple Cylindrical, WGS 84 Geodetic, or WGS 84 Lat/Lon. The NSG DSE MGR name for it is "World Geodetic System 1984 - Geographic 2D". See [1] regarding CRS84.

[5] WGS 84 Realizations from 3D GEOINT Standard, NGA.STND.0079_1.0_3DGEO, Table 6-1 on page 43, and stated in Annex C.

[6] Journal of Surveying Engineering, Vol. 148 No. 2

Quote:

The World Geodetic System 1984 (WGS84) has been consistent with the International Terrestrial Reference Frame (ITRF) since 1994. After the original release of WGS84 in 1987, there have been six subsequent realizations of the frame, designated WGS84 (G730), WGS84 (G873), WGS84 (G1150), WGS84 (G1674), WGS84 (G1762), and WGS84 (G2139).