

Open Geospatial Consortium

Submission Date: XXXX-XX-XX

Approval Date: XXXX-XX-XX

Publication Date: XXXX-XX-XX

External identifier of this OGC® document: <http://www.opengis.net/doc/IS/tms/2.0>

Internal reference number of this OGC® document: 17-083r3

Version: 2.0

Category: OGC® Implementation Standard

Editors: Joan Masó and Jérôme St-Louis

OGC Two Dimensional Tile Matrix Set and Tile Set Metadata

Copyright notice

Copyright © 2019-2021 Open Geospatial Consortium

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

Warning

This document is an OGC Members (draft) international standard. This document is available on a royalty free, non-discriminatory basis. Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type: OGC® Standard

Document subtype:

Document stage: Draft

Document language: English

License Agreement

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

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

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

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

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

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

Table of Contents

1. Scope	10
2. Conformance	11
3. References	13
4. Terms and Definitions	14
5. Conventions	19
5.1. Abbreviations	19
5.2. Identifiers	19
5.3. Common Classes	19
5.3.1. Multilingual text encoding	20
5.3.2. Description, Title and Keywords	20
5.3.3. BoundingBox	22
5.3.4. CRSType	24
5.3.5. WebLink	25
6. TileMatrixSet	27
6.1. Overview	27
6.1.1. Tile Matrix	27
6.1.2. Tile Matrix Set	30
6.1.3. Well-known scale sets	32
6.1.4. Tile based coordinates in a tile matrix set	32
6.1.5. Variable width tile matrices	33
6.2. TileMatrixSet Requirements Classes	34
6.2.1. TileMatrixSet requirements class	34
6.2.2. VariableMatrixWidth requirements class	40
7. TileMatrixSet encodings	43
7.1. JSON encoding	43
7.1.1. JSON TileMatrixSet requirements class	43
7.1.2. JSON VariableMatrixWidth requirements class	44
7.2. XML encoding	45
7.2.1. XML TileMatrixSet requirements class	45
7.2.2. XML VariableMatrixWidth requirements class	45
8. TileSetMetadata	47
8.1. Overview	47
8.1.1. TileMatrixSet limits	47
8.1.2. TileSet metadata	47
8.2. Requirements classes	48
8.2.1. TileMatrixSetLimits requirements class	48
8.2.2. TileSetMetadata requirements class	50
9. TileSetMetadata encodings	63
9.1. JSON encoding	63

9.1.1. JSON TileMatrixSetLimits requirements class	63
9.1.2. JSON TileSetMetadata requirements class	63
9.2. XML encoding	65
9.2.1. XML TileMatrixSetLimits requirements class	65
9.2.2. XML TileSetMetadata requirements class	65
Annex A: Conformance Class Abstract Test Suite (Normative)	67
A.1. Conformance Class TileMatrixSet	67
A.1.1. Model	67
A.2. Conformance Class VariableMatrixWidth	67
A.2.1. Model	67
A.2.2. Coalescence	68
A.2.3. Conformance Class JSON TileMatrixSet	68
A.2.4. Model	68
A.2.5. IETF	69
A.2.6. Schema	69
A.2.7. Media type	69
A.2.8. Conformance Class JSON VariableMatrixWidth	69
A.2.9. Model	70
A.2.10. IETF	70
A.2.11. Schema	70
A.2.12. Conformance Class XML TileMatrixSet	70
A.2.13. Model	71
A.2.14. Schema	71
A.2.15. Media Type	71
A.2.16. Conformance Class XML VariableMatrixWidth	71
A.2.17. Model	72
A.2.18. Schema	72
A.2.19. Conformance Class TileMatrixSetLimits	72
A.2.20. Model	72
A.2.21. Conformance Class TileSetMetadata	73
A.2.22. Identifier	73
A.2.23. Model	73
A.2.24. Conformance Class JSON TileMatrixSetLimits	74
A.2.25. Model	74
A.2.26. IETF	74
A.2.27. Schema	75
A.2.28. Conformance Class JSON TileSetMetadata	75
A.2.29. Model	75
A.2.30. IETF	75
A.2.31. Schema	76
A.2.32. Media type	76
A.2.33. Conformance Class XML TileMatrixSetLimits	76

A.2.34. Model	76
A.2.35. Conformance Class XML TileSetMetadata	77
A.2.36. Model	77
A.2.37. Schema	77
A.2.38. Media Type	78
Annex B: Schema Documents (Normative)	79
B.1. JSON Schema	79
B.2. XML Schema	79
Annex C: Well-known scale sets (Informative)	80
C.1. GlobalCRS84Scale	80
C.2. GlobalCRS84Pixel	81
C.3. GoogleCRS84Quad	82
C.4. GoogleMapsCompatible	84
C.5. WorldMercatorWGS84	85
Annex D: Common TileMatrixSet definitions (Informative)	88
D.1. WebMercatorQuad	88
D.2. WorldCRS84Quad	90
D.2.1. Variant 1: World CRS84 Quad (recommended)	90
D.2.2. Variant 2: World EPSG:4326 Quad	92
D.3. WorldMercatorWGS84Quad	94
D.4. Universal Transverse Mercator WGS84 Quad family (UTM##WGS84Quad)	96
D.5. Arctic Universal Polar Stereographic WGS 84 Quad (UPSArcticWGS84Quad)	100
D.6. Antarctic Universal Polar Stereographic WGS84 Quad (UPSAntarcticWGS84Quad)	103
D.7. European ETRS89 Lambert azimuthal equal-area Quad (EuropeanETRS89_LAEAQuad)	105
D.8. Canadian NAD83 Lambert Conformal Conic (CanadianNAD83_LCC)	107
Annex E: Variable width TileMatrixSet definitions (Informative)	109
E.1. GNOSISGlobalGrid	109
E.2. CDBGlobalGrid	114
Annex F: Example encodings for Common TileMatrixSets (Informative)	117
F.1. JSON schema for TileMatrixSet	117
F.2. XML schema for TileMatrixSet	122
F.3. Web Mercator Quad	128
F.3.1. Web Mercator Quad (JSON encoding)	128
F.3.2. Web Mercator Quad (XML encoding)	133
F.4. World CRS84 Quad	139
F.4.1. World CRS84 Quad (recommended, JSON encoding)	139
F.4.2. World CRS84 Quad (recommended, XML encoding)	145
F.4.3. World EPSG:4326 Quad (JSON encoding)	151
F.4.4. World EPSG:4326 Quad (XML encoding)	156
F.5. World Mercator WGS84 Quad	162
F.5.1. World Mercator WGS84 Quad (JSON encoding)	162
F.5.2. World Mercator WGS84 Quad (XML encoding)	168

F.6. Universal Transverse Mercator WGS84 Quad for zone 31	173
F.6.1. UTM WGS84 Quad for zone 31 (JSON encoding).....	174
F.6.2. UTM WGS84 Quad for zone 31 (XML encoding).....	179
F.7. Arctic Universal Polar Stereographic WGS 84 Quad	185
F.7.1. Arctic UPS WGS 84 Quad (JSON enconding).....	185
F.7.2. Arctic UPS WGS 84 Quad (XML encoding).....	190
F.8. Antarctic Universal Polar Stereographic WGS84 Quad.....	196
F.8.1. Antarctic UPS WGS84 Quad (JSON encoding).....	196
F.8.2. Antarctic UPS WGS84 Quad (XML encoding).....	202
F.9. European ETRS89 Lambert Azimuthal Equal-Area Quad	208
F.9.1. European ETRS89 Lambert azimuthal equal-area Quad (JSON encoding)	208
F.9.2. European ETRS89 Lambert Azimuthal Equal-Area Quad (XML encoding)	212
F.10. Canadian Lambert Conformal Conic NAD83	216
F.10.1. Canadian Lambert Conformal Conic NAD83 (JSON encoding).....	216
F.10.2. Canadian Lambert Conformal Conic NAD83 (XML encoding)	222
Annex G: Example encodings for Variable Matrix Width TileMatrixSets (Informative)	229
G.1. GNOSIS Global Grid	229
G.1.1. GNOSIS Global Grid (JSON encoding).....	229
G.1.2. GNOSIS Global Grid (XML encoding)	231
G.2. CDB Global Grid	234
G.2.1. CDB Global Grid (JSON encoding)	234
G.2.2. CDB Global Grid (XML encoding)	236
Annex H: Example encodings for TileSetMetadata (Informative)	240
H.1. Example JSON encoding of TileSetMetadata	240
H.2. TileSetMetadata JSON Schema	244
H.3. TileMatrixSetLimits JSON Schema	255
H.4. Example XML encoding of TileSetMetadata	256
H.5. TileSetMetadata XML Schema	261
H.6. TileMatrixSetLimits XML Schema	274
Annex I: Pseudocode (Informative)	277
I.1. From BBOX to tile indices	277
I.2. From tile indices to BBOX	277
Annex J: Changes from previous version (Informative)	279
J.1. TileSetMetadata	279
J.2. OWS Common dependency removed	279
10. Extending TileMatrixSets for additional dimensions (Informative)	280
10.1. Extension approaches	280
10.1.1. No explicit tiling of extra dimensions	280
10.1.2. Same division for all tile matrices	281
10.1.3. Octrees and Orthotrees (Hyperoctrees)	282
10.2. Extended properties for TileMatrix & limits	283
10.2.1. Example TileMatrix extended to 4 dimensions (2 extra dimensions)	284

10.3. Data contained in tiles	285
10.3.1. Vector Features	286
10.3.2. Coverages	286
10.3.3. Point Clouds	286
10.3.4. Point Features instantiating 3D models	286
10.3.5. Batched 3D Models	287
10.4. Relationship with <i>3D Tiles</i> and <i>i3s</i>	288
Annex K: Revision History	289
Annex L: Bibliography	290

i. Abstract

The OGC Tile Matrix Set standard defines the rules and requirements for a tile matrix set as a way to index space based on a set of regular grids defining a domain (tile matrix) for a limited list of scales in a Coordinate Reference System (CRS) as defined in [OGC 08-015r2] Abstract Specification Topic 2: Spatial Referencing by Coordinates. Each tile matrix is divided into regular tiles. In a tile matrix set, a tile can be univocally identified by a tile column a tile row and a tile matrix identifier. This document presents a data structure defining the properties of the tile matrix set in both UML diagrams and in tabular form. This document also presents a data structure to define a subset of a tile matrix set called tile matrix set limits. XML and JSON encodings are suggested both for tile matrix sets and tile matrix set limits. Finally, the document offers practical examples of tile matrix sets both for common global projections and for specific regions.

ii. Keywords

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

ogcdoc, OGC document, tiles, maps, tile matrix set

iii. Preface

In 2007 OGC approved and released the Web Map Tile Service standard [OGC 07-057r7] (WMTS). This OGC standard provides a definition of a “tile matrix set.” Over time, other OGC standards dealing with tiles in other ways needed to use the same definition. Unfortunately, these OGC standards could not use the tile matrix set definition directly because the definition was formally linked to the tile service. This document frees the concept of a tile matrix set from the WMTS standard so that other standards can reference the concept directly. This standard also adds an informative list of commonly used tile matrix sets. The submitters believe that other tile matrix concepts will emerge in the future. This document is anticipated to impact future revisions of other OGC standards such as GeoPackage.

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

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

iv. Security Considerations

The correct definition of a tile matrix set is crucial to be able to correctly geo-reference a tile. The application of the wrong tile matrix set could result in an incorrect geo-referencing of the tiles and the features represented in those tiles. In an emergency situation, such poor referencing

could result in sending first responders to the wrong location.

In a normal service interaction, the client requests the TileMatrixSet once and requests one or more tiles afterwards. The client needs to ensure that the TileMatrixSet definition has not been tampered with and corresponds to the correct TileMatrixSet. In practice this means that the client and server must use a mechanism to ensure that the service is really who it claims to be and that the message that travels from the server to the client has not been altered.

If a server points to a definition of a TileMatrixSet that is hosted elsewhere, in addition to the precautions stated above, the client must ensure that the service providing the definition of the TileMatrixSet is a trusted service. In addition, the synchronization of the tiles and the tile matrix set definition need to be ensured, guaranteeing that the tile matrix set definition has not been updated afterwards without the tile service knowing it.

v. Submitting organizations

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

- Universitat Autònoma de Barcelona (UAB-CREAF)
- Image Matters LLC
- Natural Resources Canada (NRCan)
- Ecere Corporation
- US Army Geospatial Center

vi. Submitters

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

Name	Affiliation
Joan Masó	Universitat Autònoma de Barcelona (UAB-CREAF)
Jeff Yutzler	Image Matters LLC
Peter Rushforth	Canada Centre for Mapping and Earth Observation, Natural Resources Canada
Jérôme St-Louis	Ecere Corporation
Jeff Harrison	US Army Geospatial Center

Chapter 1. Scope

This document specifies the concepts of a tile matrix set and tile set metadata, prioritizing their implementation in 2D space but with some considerations on how to extend this to nD. This standard also provides both XML and JSON encodings.

The Tile Matrix Set concept, initially developed in OGC Web Map Tile Service (WMTS) 1.0, is now provided as an independent standard that can be referenced by other standards such as *OGC API - Tiles*, *OGC GeoPackage*, *OGC CDB*, or the Natural Resources Canada (NRCan)-promoted specification *Map Markup Language (MapML)*. In addition, this standard ensures that the tile matrix set concept can be used to tile both gridded as well as vector data.

This document also contains an informative annex for [Common TileMatrixSet definitions \(Informative\)](#) with a library of proposed tile matrix set definitions for Mercator, Transverse Mercator, Polar Stereographic, Lambert Azimuthal Equal Area, and Lambert Conformal Conic. An additional annex for [Variable width TileMatrixSet definitions \(Informative\)](#) provides tile matrix set definitions, utilizing the variable width capabilities of this standard, which allows for roughly approximate equal area tiles for Plate Carrée projections.

Global identifiers for the Tile Matrix Sets defined in these annexes are registered with the OGC Naming Authority in the [Tile Matrix Set Register](#) [<http://www.opengis.net/def/tms>].

The Tile Set Metadata provides information about the intended use of a Tile Set as well as the origin, security level, tiling scheme, layers and feature properties contained within. A tile set is a series of tiles containing data and following a common tiling scheme. Tile Set Metadata is intended to facilitate retrieval of tile sets and describes the major characteristics of tile sets without actually accessing the tiles nor the content within a tile. The Tile Set Metadata was initially developed in phase two of the OGC Vector Tiles Pilot.

NOTE

A previous version of this standard had a JSON-LD encoding of the classes presented in [\[tilematrixset-model\]](#). This encoding was abandoned as no interest was detected.

Chapter 2. Conformance

This standard defines tile matrix set, tile matrix set limits and tile set metadata.

Requirements for the following standardization target types are considered.

- TileMatrixSet: This abstract class defines a data model for tile matrix sets [<http://www.opengis.net/spec/tms/2.0/req/tilematrixset>]. This abstract class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/tilematrixset>]. The target is a service or an encoding needing to define TileMatrixSet (e.g., a future version of WMTS service metadata or an OGC API - Tiles).
- TileMatrixSetLimits: This abstract class defines a data model for tile matrix sets limits [<http://www.opengis.net/spec/tms/2.0/req/tilematrixsetlimits>]. This abstract class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/tilematrixsetlimits>]. The target is a service, a client, or an encoding needing to define TileMatrixSetLimits (e.g., a future version of WMTS service metadata or an OGC API - Tiles).
- TileSetMetadata: This abstract class defines a data model for tile set metadata in [<http://www.opengis.net/spec/tms/2.0/req/tilesetmetadata>]. This abstract class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/tilesetmetadata>]. The target is a service, a client, or an encoding needing to declare conformity to a TileMatrixSet or/and a TileMatrixSetLimits (e.g., a future version of WMTS service metadata or an OGC API - Tiles).

NOTE

TileSetMetadata in this standard replaces and extends TileMatrixSetLink in the version 1.0

- XMLTileMatrixSet: This class defines an encoding in XML for tile matrix sets [<http://www.opengis.net/spec/tms/2.0/req/xml-tilematrixset>]. This class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/xml-tilematrixset>]. The target is a service, a client, or an encoding needing to define a TileMatrixSet in XML (e.g., a future version of WMTS service metadata).
- XMLTileMatrixSetLimits: This class defines an encoding in XML for tile matrix sets limits in [<http://www.opengis.net/spec/tms/2.0/req/xml-tilematrixsetlimits>]. This class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/xml-tilematrixsetlimits>]. The target is a service, a client, or an encoding needing to define a TileMatrixSetLimits in XML (e.g., a future version of WMTS service metadata or an OGC API - Tiles).
- XMLTileSetMetadata: This class defines an encoding in XML for tile set metadata [<http://www.opengis.net/spec/tms/2.0/req/xml-tilesetmetadata>]. This class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/xml-tilesetmetadata>]. The target is a service, a client or an encoding needing to declare conformity to a TileMatrixSet or/and a TileMatrixSetLimits using an XML encoding (e.g., a future version of WMTS service metadata).

NOTE

XMLTileSetMetadata in this standard replaces and extends XMLTileMatrixSetLink2d in the version 1.0

- **JSONTileMatrixSet:** This class defines an encoding in JSON for tile matrix sets [<http://www.opengis.net/spec/tms/2.0/req/json-tilematrixset>]. This class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/json-tilematrixset>]. The target is a service, a client, or an encoding needing to define a TileMatrixSet in JSON (e.g., a future version of an OGC API - Tiles).
- **JSONTileMatrixSetLimits:** This class defines an encoding in JSON for tile matrix sets limits in [<http://www.opengis.net/spec/tms/2.0/req/json-tilematrixsetlimits>]. This class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/json-tilematrixsetlimits>]. The target is a service, a client ,or an encoding needing to define a TileMatrixSet in JSON (e.g., a future version of an OGC API - Tiles).
- **JSONTileSetMetadata:** This class defines an encoding in JSON for tile set metadata in [<http://www.opengis.net/spec/tms/2.0/req/json-tilessetmetadata>]. This class has a single conformance class: [<http://www.opengis.net/spec/tms/2.0/conf/json-tilessetmetadata>]. The target is a service, a client, or an encoding needing to declare conformity to a TileMatrixSet or/and a TileMatrixSetLimits using a JSON encoding (e.g., a future version of an OGC API - Tiles).

NOTE

JSONTileSetMetadata in this standard replaces and extends JSONTileMatrixSetLink2d in the version 1.0

Conformance with this standard shall be verified using all the relevant tests specified in **Conformance Class Abstract Test Suite (Normative)** (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site ^[1].

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

[1] <http://www.opengeospatial.org/cite>

Chapter 3. References

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

- [1] OGC: OGC 08-005r2 Abstract Specification Topic 2: Spatial referencing by coordinates, 2010.
http://portal.opengeospatial.org/files/?artifact_id=39049
- [2] OGC: OGC 09-146r6 Coverage Implementation Schema (“CIS”) Version 1.1, 2017.
<http://docs.opengeospatial.org/is/09-146r6/09-146r6.html>
- [3] IETF: IETF RFC 7159, The JavaScript Object Notation (JSON) Data Interchange Format, 2014.
<https://www.ietf.org/rfc/rfc7159.txt>
- [4] W3C: W3C JSON-LD 1.0, A JSON-based Serialization for Linked Data, 2014.
<http://www.w3.org/TR/json-ld/>

Chapter 4. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r9], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

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

4.1

cell

minimum geometrical spaces delimited by the grid lines of a regular grid.

NOTE in 2D spaces, cells are often referred as pixels.

NOTE In this standard, the term *pixel* is reserved to the individual elements of a visualization device. Tiles are composed by regular grid cells that can be made partially coincident with the pixels of a visualization device for display purposes.

4.2

coordinate reference system

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

4.3

coordinate system

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

4.4

domain

well-defined set [ISO/TS 19103:2005]

NOTE A mathematical function may be defined on this set, i.e. in a function $f:A \rightarrow B$, A is the domain of the function f.

4.5

grid

network composed of two or more sets of curves in which the members of each set intersect the members of the other sets in an algorithmic way [ISO 19123:2005]

NOTE The curves partition a space into grid cells.

NOTE A grid can be used to define a tessellation of the space.

4.6

range set

set of all values a function f can take as its arguments vary over its domain [OGC 07-036]

4.7

raster tile

tile that contains information in a gridded form. Commonly the values of the grid represent colors of each cell in the grid for immediate pictorial representation on visualization devices, but can also be coverage subsets.

NOTE This concept is used in this standard as a contraposition of "vector tiles". Many of the existing implementations of WMTS 1.0 produce "raster tiles".

4.8

regular grid

grid whose grid lines have a constant distance along each grid axis [OGC 09-146r6]

NOTE A regular grid can be used to define a regular tessellation of the space.

4.9

space partitioning

process of dividing a geometric space (usually a Euclidean space) into two or more disjoint subsets (see also partition of a set). Space partitioning divides a space into non-overlapping regions. Any point in the space can then be identified to lie in exactly one of the regions [OGC 19-014r3]

4.10

square

regular quadrilateral with four equal sides and four 90 degree angles [OGC 19-014r3]

4.11

tessellation

partitioning of a space into a set of conterminous subspaces having the same dimension as the space being partitioned [ISO 19123]

NOTE

A tessellation composed of congruent regular polygons or polyhedra is a regular tessellation. One composed of regular, but non-congruent polygons or polyhedra is a semi-regular tessellation. Otherwise the tessellation is irregular.

4.12

tile

geometric shape with known properties that may or may not be the result of a tiling (tessellation) process. A tile consists of a single connected "piece" (topological disc) without "holes" or "lines" [OGC 19-014r3]

+ small rectangular representation of geographic data, often part of a set of such elements, covering a tiling scheme and sharing similar information content and graphical styling. A tile can be uniquely defined in a tile matrix by one integer index in each dimension. Tiles are mainly used for fast transfer (particularly in the web) and easy display at the resolution of a visualization device. Tiles can be grid based pictorial representations, coverage subsets, or feature based representations (e.g., vector tiles).

4.13

tile matrix

grid tile scheme that defines how space is partitioned into a set of conterminous tiles at a fixed scale.

NOTE

In 2d, a tile matrix constitutes a tessellation of the space that resembles a matrix characterized by a matrix width (columns) and a matrix height (rows).

NOTE

In this standard, a tile matrix is not used alone and it is part of a tile matrix set.

4.14

tile matrix set

tile scheme composed by collection of tile matrices defined at different scales covering approximately the same area and have a common coordinate reference system.

4.15

tile scheme

scheme that defines the unique properties of each individual tile in a tile set. These properties include a unique identifier for each tile, the tile origin, and the extent of the tile.

NOTE

A tile scheme is not restricted to a coordinate reference system or a tile matrix set and allows for other spatial reference systems such as DGGS and other organizations including irregular ones.

4.16

tile set

set of tiles - a collection of subsets of the space being partitioned. [OGC 19-014r3]+

+ series of actual tiles contain data and following a common tiling scheme.

4.17

tile set metadata

additional metadata beyond the common properties defining the tile set. Such metadata could be an abstract, the owner, the author, or other common metadata. [OGC 19-014r3]+

+ metadata describing common properties defining a tile set, layers and styles used to produce the tile set, the limits of the time matrix with actual data and common metadata such as abstract, owner, author, etc.

4.18

vector tile

tile that contains geometric features information that has been simplified at the tile scale resolution and clipped by the tile boundaries.

NOTE

The expression "vector tile" has stirred some controversy in the OGC. Actually, the OGC uses geometrical features to refer to things that are commonly known as vectors in many GIS tools. However, in this case, this standard recognizes the ubiquity of the expression in the sector and assumes that the concept is not associated to any particular technology or commercial brand.

4.19

well-known scale set

well-known combination of a coordinate reference system and a set of scales that a tile matrix set declares support for

Chapter 5. Conventions

This section provides details and examples for any conventions used in the document.

5.1. Abbreviations

2D

2-dimensional

3D

3-dimensional

CDB

Abbreviated reference for the OGC CDB Standard.

CRS

Coordinate Reference System

DGGS

Discrete Global Grid System

EPSG

European Petroleum Survey Group

JSON

JavaScript Object Notation

TMS

Tile Matrix Set

WMTS

Web Map Tile Service

XML

eXtensible Markup Language

5.2. Identifiers

The normative provisions in this specification are denoted by the URI

<http://www.opengis.net/spec/TileMatrixSet/2.0>

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

5.3. Common Classes

The following classes are extracted from the OWS Common 06-121r9. This way, the data structures presented in this document become independent from OWS Common and can be used

in the emerging OGC APIs that depends on OGC API Commons (avoiding a conflicting dependency with the old OWS Common)

5.3.1. Multilingual text encoding

Some text parameters specified with the data type `CharacterString` in UML are intended to have human-readable values, but not all humans can understand the same languages. This data class can be latter mapped to XML or JSON encodings.

The specified approach for allowing the language of a text value to be explicitly stated is indicated by the UML class diagram in the figure.

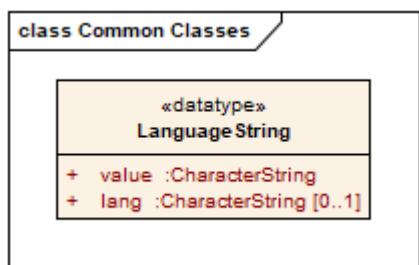


Figure 1. *LanguageString* UML model

The `value` parameter specifies the human-language string, and the `lang` parameter specifies the language (in IETF RFC 4646 syntax) of the string. If a `lang` parameter is not present, then no language has been specified for the string unless specified by another means.

In case that multiple languages are necessary in a single document instance, the element that is of the `LanguageString` type should be a list with one entry for each `lang` code.

NOTE

OGC APIs will use the language negotiation with HTTP headers. In this situation, it is expected that elements defined as a list of `LanguageString` will default into a single `CharacterString` that in JSON will default into a string data type. This does not preclude that in other environments the JSON encodings for language string can implement the `LanguageString`. In practice this means that a JSON schema for a `LanguageString` element should support both string and language string types.

5.3.2. Description, Title and Keywords

A basic set of data description parameters that includes human-readable title, description, and keywords is widely used in this standard and defined here as UML.

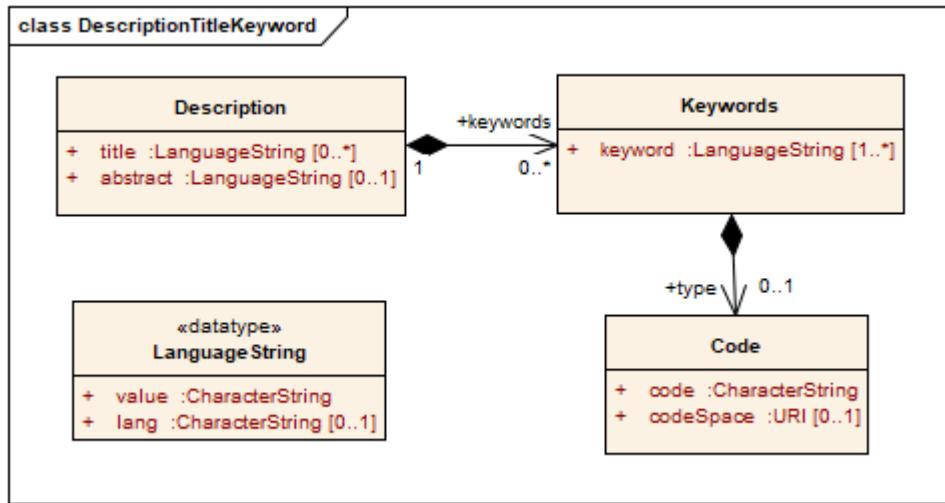


Figure 2. Description Title Keywords UML model

Names	Definition	Data type and values	Multiplicity and use
title	Title of this tile matrix set, normally used for display to a human	LanguageString data structure	Zero or more (optional) Include when available and useful Include one for each language represented
description	Brief narrative description of this tile matrix set, normally available for display to a human	LanguageString data structure	Zero or more (optional) Include when available and useful Include one for each language represented
keywords	Unordered list of one or more keywords	MD_Keywords class in ISO 19115	Zero or more (optional) One for each keyword authority used (one for each 'type' value)

Table 1. Parts of Description Title Keyword data elements

Keywords is described from MD_Keywords class in ISO 19115-1 and has a list of keywords of the same 'type' specified in the optional 'type' element,

Names	Definition	Data type and values	Multiplicity and use
keyword	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this dataset	LanguageString data structure	One or more (optional)
type	Type of the keyword	CodeType, as adaptation of MD_Identifier class ISO 19115	Zero or one (optional)

Table 2. Parts of Keyword data elements

NOTE

OGC APIs will use language negotiation with HTTP headers. In this situation, it is expected that elements defined as a list of LanguageString will default into a single CharacterString that in JSON will default into a string data type. In JSON encodings we are not considering namespaces or codespaces (optional in the model). This will result in a simplification of the keywords in the JSON encoding to a simple array of strings.

5.3.3. BoundingBox

A (basic) bounding box is one type of bounding box that is used in this document. The Bounding box data structure is specified in the following UML model and table.

The BoundingBox class describes a Minimum Bounding Rectangle (MBR) surrounding a feature (in the broader sense), in the supported CRS.

A 2DBoundingBox is another type of bounding box. This type is simplified from the basic BoundingBox data type for use only with the 2D geographic coordinate reference system. This is useful to specify the extent 2D part of tile matrix set.

A WGS84BoundingBox is another type of bounding box. This type is simplified from the basic BoundingBox data type for use only with the 2D geographic coordinate reference system which uses the WGS 84 geodetic datum, where longitude precedes latitude and both are recorded in decimal degrees.

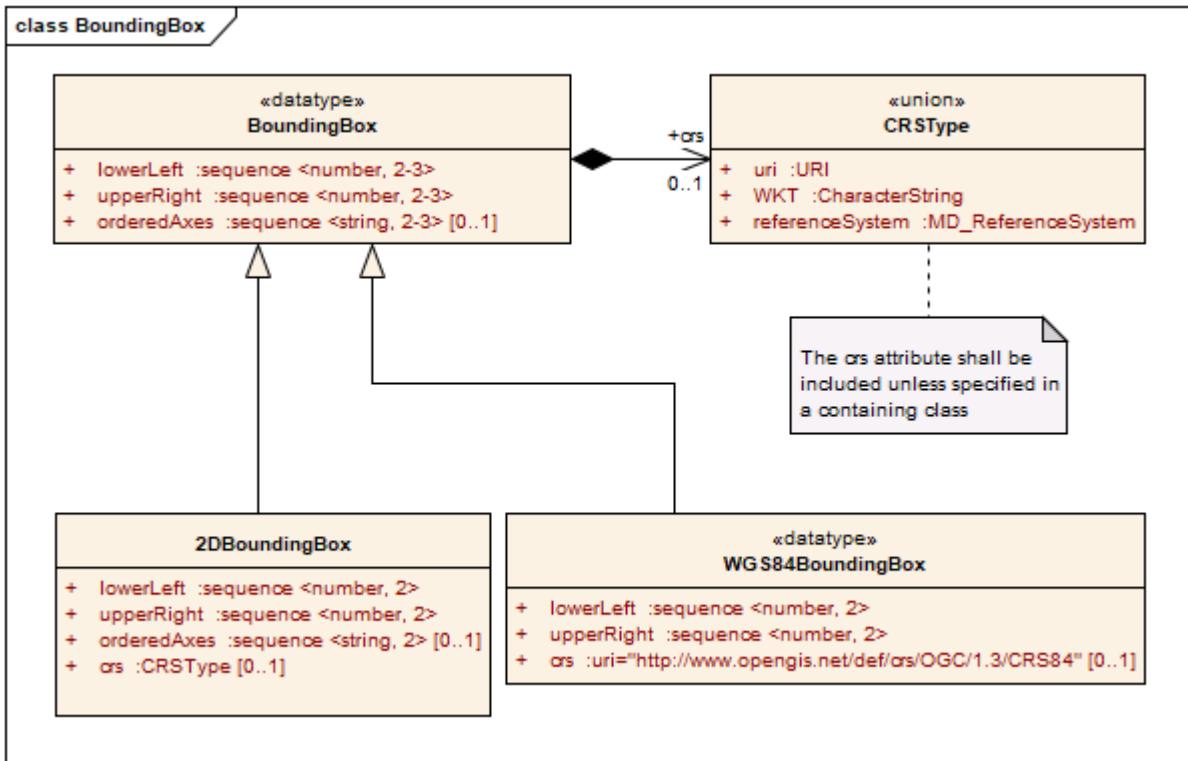


Figure 3. BoundingBox UML model

Names	Definition	Data type and values	Multiplicity and use
lowerLeft	Coordinates of bounding box corner at which the value of each coordinate normally is the algebraic minimum within this bounding box ^a	Ordered sequence of double values ^b	One (mandatory)
upperRight	Coordinates of bounding box corner at which the value of each coordinate normally is the algebraic maximum within this bounding box ^a	Ordered sequence of double values ^b	One (mandatory)
CRS	Reference or a definition of the CRS used by the lowerRight and upperRight coordinates	CRSType	Zero or one (optional) Include unless referenced elsewhere
orderedAxis	Ordered list of names of the dimensions defined in the CRS	Ordered sequence of strings	Zero or one (optional) ^c

Names	Definition	Data type and values	Multiplicity and use
^a Values other than the minimum and maximum may be used as discussed below.			
^b The number of axes included, and the order of these axes, as specified by the referenced CRS.			
^c The number of axes and names is specified by the referenced CRS definition, but may also be specified here for convenience. In particular, it makes the axis order more visible.			

Table 3. Parts of BoundingBox data structure

The bounding box contents defined will not always specify the MINIMUM rectangular BOUNDING region, if the referenced CRS uses an Ellipsoidal, Spherical, Polar, or Cylindrical coordinate system, as those terms are specified in OGC Abstract Specification Topic 2. Specifically, this box will not specify the minimum rectangular bounding region surrounding a geometry whose set of points span the value discontinuity in an angular coordinate axis. Such axes include the longitude and latitude of Ellipsoidal and Spherical coordinate systems. That geometry could lie within a small region on the surface of the ellipsoid or sphere.

Theoretically, there are cases where defining a bounding box could be problematic or impossible, such as angular axis of an Ellipsoidal, Spherical, Polar, or Cylindrical coordinate system. However, tiles need to be circumscribed to real coordinates and will deliberately avoid regions of the space where coordinates go to infinite or cannot be defined. For example, the [WorldMercatorWGS84Quad](#) tile matrix set (based on a cylindrical projection) should not be used close to the poles. Since tiles are conterminous, it is always possible to define a bounding box that includes them all.

5.3.4. CRSType

In this version of this standard we introduce the possibility define a CRS using a full description in addition to a reference to an external CRS catalogue. For descendent compatibility, CRSType can be defaulted as a URI but in here we extend it to a union of three possibilities.

Names	Definition	Data type and values
uri	A reference to a CRS. Typically a EPSG CRS reference	URI
wkt	A string defining the CRS using Well Known Text	CharacterString
referenceSystem	A reference system data structure as defined in the MD_ReferenceSystem of the ISO 19115	MD_ReferenceSystem

Table 4. Parts of CRSType data structure

5.3.5. WebLink

Many recent standards emphasize the usefulness of links as a way to relate a data structure instance to other data structures and make navigation through resources possible. Essential links are made explicit in the data structures of this document (recognizable by a URI data type) but other links can be added as needed for convenience when a WebLink is available. The data structure defined here allows the addition of other links. The definition is based on the web linking defined in the RFC8288 and the XML serialization present in section 4.2.7 of the RFC4287 and in the JSON serialization found in this IETF draft: <https://tools.ietf.org/id/draft-pot-json-link-01.html>

NOTE

In practice, some encodings can opt to specify the essential links as part of this data structure for convenience

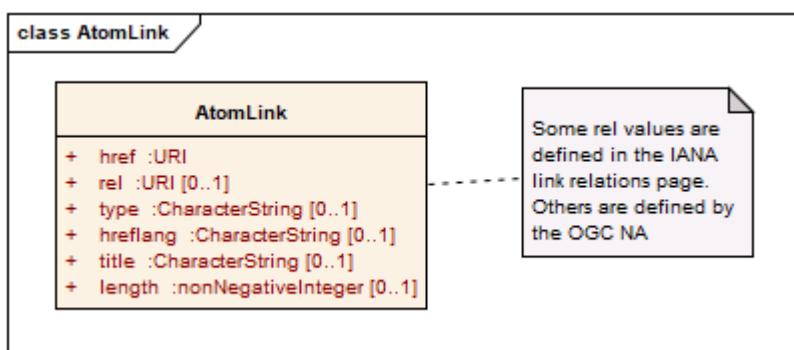


Figure 4. Web link UML model

Names	Definition	Data type and values	Multiplicity and use
href	Reference from this resource to a web resource	URI or a URI template	One (mandatory)
rel	Link relation type describing the meaning of the link.	CharacterString ^a	Zero or one (optional)
type	hint about the type of the representation that is expected to be returned from the href attribute	CharacterString ^b	Zero or one (optional)
hreflang	language of the resource pointed to by the href attribute	LanguageString data structureCharacterString ^c	Zero or one (optional)
title	human-readable information about the link	CharacterString ^c	Zero or one (optional) Include one for each language represented

Names	Definition	Data type and values	Multiplicity and use
length	hint about the length of the linked content in bytes	nonNegativeInteger	Zero or one (optional)
^a It can be a name or a URI. If a name is given, implementations must consider the link relation type equivalent to the same name registered within the IANA Registry of Link Relations. The OGC NA maintains other possible values.			
^b It should be a media type format as specified in section 4.2 of rfc6838			
^c As specified in RFC5646			

Table 5. Parts of WebLink data structure

Chapter 6. TileMatrixSet

6.1. Overview

As stated in [OGC 08-015r2] Abstract Specification Topic 2: Spatial Referencing by Coordinates, a coordinate system is a set of mathematical rules for specifying how coordinates are to be assigned to points in space. A CRS is a coordinate system that is related to the real world by a reference datum. An example of mathematical rules is the application of a sphere or an ellipsoid centered in the datum and the use of a projection to transform the sphere or the ellipsoid into a planar representation of the world. Usually, the resulting planar coordinates are expressed as real numbers that are distances to the origin of the projection. This section introduces a tile scheme called Tile Matrix Set that is defined on top of a CRS. A fundamental part of the definition of a Tile Matrix Set is the Tile Matrix.

6.1.1. Tile Matrix

A *Tile Matrix* is a tile scheme defined as a grid coverage. For the [OGC 09-146r6] CIS GeneralGridCoverage, the domain set of a grid describes the direct positions in multi-dimensional coordinate space, depending on the type of grid. In a grid-regular, simple equidistant grids are established. When a regular grid coverage is used to represent the world, the space becomes discrete in each dimension of the grid *domain range*. One possible discrete subdivision is the use of multidimensional atomic *grid cells*. Another is to divide the domain into regular intervals that can be assigned to integer numbers that enumerate and identify *tiles*. This grid of tiles domain range can be defined by:

- a. The extreme corner (called point of origin and corner of origin in a two-dimensional space) of the bounding box of regular grid coverage (e.g., the CRS coordinates of the top left corner of the top left extreme where the integer coordinates are 0);
- b. A tile size in CRS units for in each dimension of the CRS; and
- c. The size of the tile matrix in tile units (i.e., number of tiles) that closes the bounding box of the tiled space. Frequently the sizes of the two first dimensions are called *matrix width* and *matrix height*.

Tile matrix in a two-dimensional space.

Tiles are originally designed to be rendered in visualization devices that have the space quantized in pixels characterized by a size.

The tile size in CRS units of the first two spatial dimensions and the size of the visualization device pixels are related. The two spatial dimensions are aligned with the pixel axes of the device. The relation is a function of the following two parameters: In raster tiles, a second regular grid that is coincident with the original regular grid coverage domain range but denser (with smaller cell size but an exact submultiple of that size) is defined. Each grid cell of this new higher resolution grid is called a *grid cell*. The gridded space is defined by the same corner as the original tile matrix. The

grid cells are defined by equally dividing the original tiles into *grid cells* using the *number of rendering cells in a tile (tile width and tile height)*. In common tiled 2D visualization clients, a part of the *grid cell* is made coincident with the device pixels and this part of the grid is rendered in the device: the second grid is known as the *extrapolated device grid*. In other words, a tile is divided in a number of cells in each dimension of the CRS in a way that creates cells of the exact same size of the pixels of a visualization device.

- a. A scale (expressed as a scale denominator) and
- b. A grouping of *rendering pixels* in a tile forming the tile. Common grouping values are 256x256 or 512x512. Frequently the sizes of the two first dimensions are called *tile width* and *tile height*.

NOTE

Commonly *tile width* and *tile height* are equal but this constraint is not imposed by this standard.

The scale denominator is defined here with respect to a “standardized rendering pixel size” of 0.28 mm × 0.28 mm (millimeters). The definition is the same as used in Web Map Service WMS 1.3.0 [OGC 06-042] and in Symbology Encoding (SE) Implementation Specification 1.1.0 [OGC 05-077r4] that was later adopted by WMTS 1.0 [OGC 07-057r7]. Frequently, the true pixel size of the device is unknown and 0.28 mm was the actual size of a common display from 2005. This value is still being used as reference, even if current display devices are built with much smaller pixel sizes.

NOTE

Since the 1980s, the Microsoft Windows operating system has set their default standard display pixels per inch (PPI) to 96. This value results in an approximated 0.264 mm per pixel. The similarity of this value with the actual 0.28 mm adopted in this standard can create some confusion.

NOTE

Modern display devices (screens) have pixels so small that operating systems allow for defining a presentation scale factor bigger than one (e.g. 150%). In these circumstances, the actual size of the device pixels are not the same as the size used by the operating system.

Normally the *matrix width* is constant and in this circumstance, having a single scale factor using a single standardized rendering cell size for the two dimensions, results in cells that have the same size in the first two dimensions. This is commonly known as *square pixels*.

NOTE

The geometry above is different from WMS, which does allow non-square pixels (although many implementations fail to support non-square pixels properly).

NOTE

In map tiles, it is common that the range set represents colors of the cells and is stored in PNG or JPEG files of exactly one tile. Nevertheless, nothing prevents to store other kinds of values in other formats, such as TIFF files.

Vector tiles also make use of the *extrapolated device grid*, where vector tiles can be rendered for visualization purposes.

NOTE

Vector tiles expressed in pure vector formats, such as GeoJSON, do not make use of an *extrapolated device grid*. Other vector tiles formats (e.g., MBTiles) define an internal coincident grid denser than the *extrapolated device grid* and expresses the vector coordinates using indices in this denser grid.

For the case of a two-dimensional space, given the top left point of the tile matrix in CRS coordinates (`tileMatrixMinX`, `tileMatrixMaxY`), the width and height of the tile matrix in tile units (`matrixWidth`, `matrixHeight`), the *rendering cells in a tile* values (`tileWidth`, `tileHeight`), the coefficient to convert the coordinate reference system (CRS) units into meters (`metersPerUnit`), and the scale (1:`scaleDenominator`), the bottom right corner of the bounding box of a tile matrix (`tileMatrixMaxX`, `tileMatrixMinY`) can be calculated as follows:

$$\text{cellSize} = \text{scaleDenominator} \times 0.28 \times 10^{-3} / \text{metersPerUnit(crs)}$$

$$\text{tileSpanX} = \text{tileWidth} \times \text{cellSize}$$

$$\text{tileSpanY} = \text{tileHeight} \times \text{cellSize}$$

$$\text{tileMatrixMaxX} = \text{tileMatrixMinX} + \text{tileSpanX} \times \text{matrixWidth}$$

$$\text{tileMatrixMinY} = \text{tileMatrixMaxY} - \text{tileSpanY} \times \text{matrixHeight}$$

NOTE

In a CRS with coordinates expressed in meters, `metersPerUnit(crs)` equals 1.

NOTE

In CRS with coordinates expressed in degrees `metersPerUnit(crs)` equals $360 / (\text{EquatorialRadius}^2 * \pi)$ (360 degrees are equivalent to the EquatorialPerimeter). E.g. for WGS84 `metersPerUnit(crs)` is 111319.4908 meters/degree.

The tile space therefore looks like this:

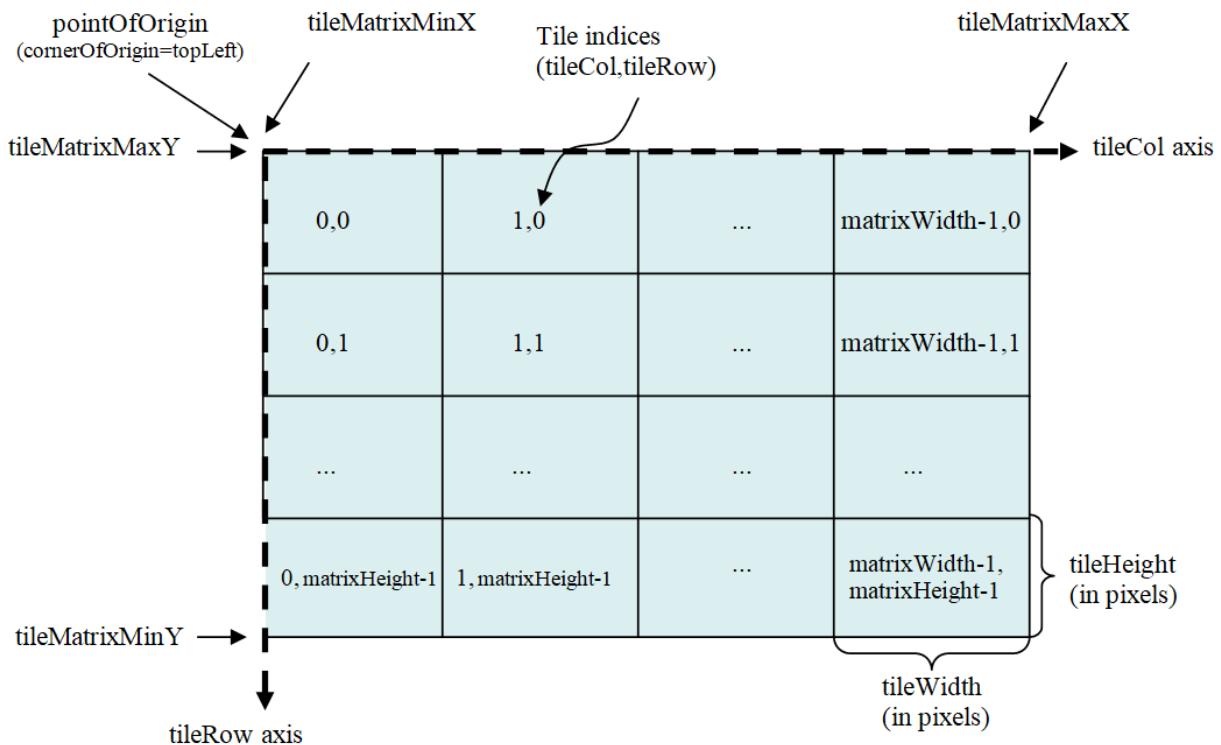


Figure 5. Tile Space (the corner of origin is topLeft)

Each tile in a tile matrix is identified by its *tileCol* and *tileRow* indices that have their 0,0 origin in one of the corners of the tile matrix. When the topLeft corner is used, *tileCol* increases towards the right and *TileRow* towards the bottom, as shown in Figure 1 (bottomLeft corner can also be used as origin making *TileRow* increase towards the top). Annex I [pseudocode-informative] in this document includes pseudocode that illustrates the process for obtaining the tile indices that cover a bounding box rectangle and also the computation to get the CRS coordinates that bound a tile.

NOTE

A tile matrix can be implemented as a set of image files (e.g., PNG or JPEG) in a file folder, each file representing a single tile

NOTE

Section 6 of the TIFF specification v6 defines 2D tiles in the same way that has been done in this standard. All tiles in a tile matrix can be stored in a single TIFF file. The TIFF file includes only one set conterminous tiles sharing a common single scale.

6.1.2. Tile Matrix Set

Depending on the range of scales needed to be represented in the screen of a client, a single tile matrix is impractical and might force the software to spend too much time simplifying/generalizing the dataset prior to rendering.

Commonly, several tile matrices are progressively defined covering the expected ranges of scales needed for the application. A *Tile Matrix Set* is a tile scheme composed of a collection of tile

matrices, optimized for a particular scale and identified by a tile matrix identifier. Each Tile Matrix Set has an optional approximated bounding box, but each tile matrix has an exact bounding box that is deduced indirectly from other parameters. Tile matrix bounding boxes at each scale will usually vary slightly due to their cell alignment.

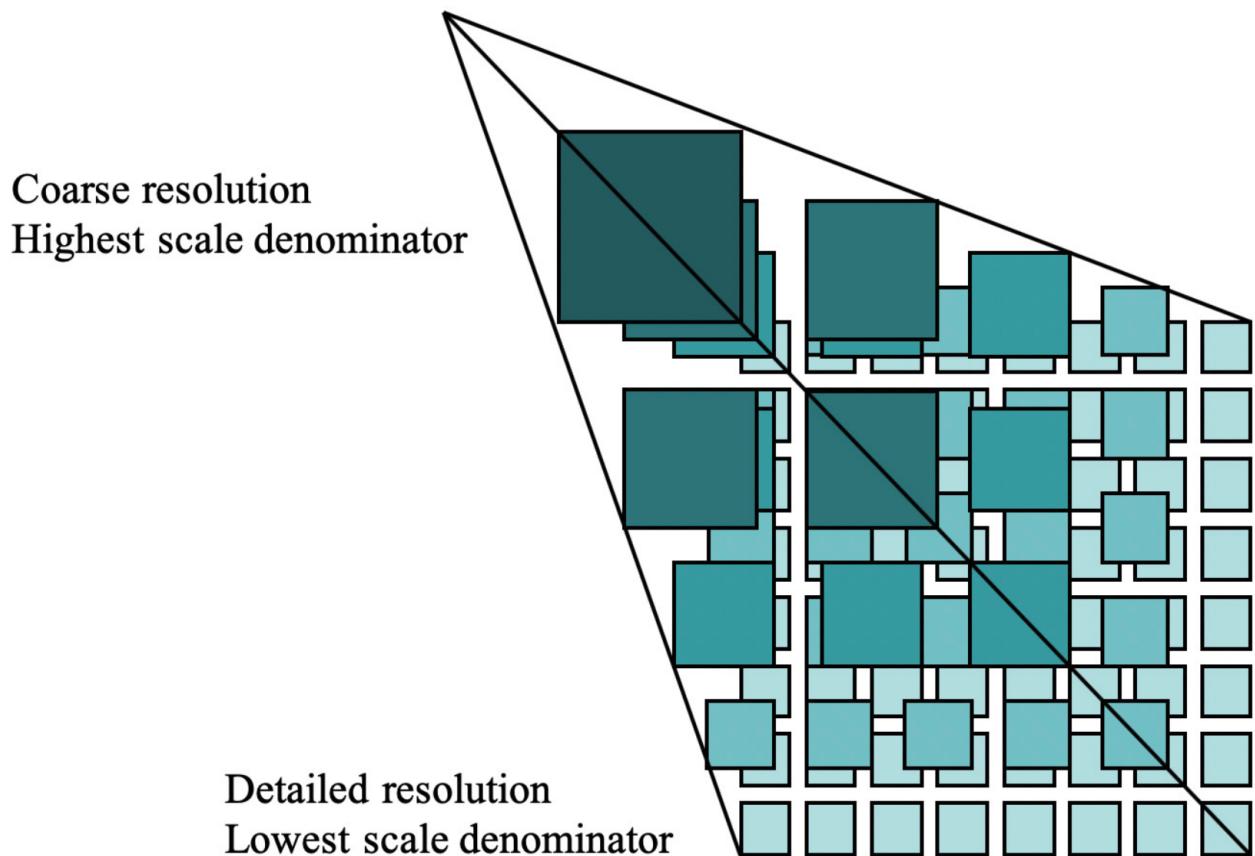


Figure 6. Tile Matrix Set representation

A Tile Matrix has a unique alphanumeric identifier in the Tile Matrix Set. Some tile-based implementations prefer to use a *zoom level* number which has the advantage of suggesting some order in the list of tile matrices. This standard does not use the *zoom level* concept but, to ease adoption of this standard in implementations that prefer numeric zoom levels, many Tile Matrix Sets defined in [Common TileMatrixSet definitions \(Informative\)](#) use numbers as Tile Matrix identifiers. If this is not the case, the index order in the list of tile matrices defined in a Tile Matrix Set could still be used as a *zoom level ordering* internally.

In some other standards, the tile matrix set concept is called an *image pyramid*, like in clause 11.6 of the OGC KML 2.2 [OGC 07-147r2] standard. JPEG2000 (ISO/IEC 15444-1) and JPIP (ISO/IEC 15444-9) also use a similar division of the space called *resolution levels*. Nevertheless, in those cases the pyramid is self-defined starting from the more detailed tile matrix (that uses square tiles), and constructing tiles of the next scales by successively aggregating 4 tiles of the previous scale, and so on (see Figure 2), and interpolating each 4 contiguous values of the previous scale into one in the next scale. That approach involves a more rigid structure which has scales related by powers of two and tiles that perfectly overlap tiles on the inferior scale denominators. Tile Matrix Sets presented in this document are more flexible, but KML superoverlays or JPEG2000-based implementations can use this standard with some extra rules to describe their tile matrix sets. This document describes some tile matrix sets with scale sets related by powers of two in

the [Common TileMatrixSet definitions \(Informative\)](#).

Each of the WMTS procedure-oriented architectural style operations and resource-oriented architectural style resources are described in more detail in subsequent clauses in this standard.

NOTE

Clients and servers have to be careful when comparing floating numbers with tolerance (double precision, 16-digit numbers, have to be used).

6.1.3. Well-known scale sets

When overlaying and presenting together tiles encoded in different tile matrix sets that do not have common sets of scale denominators and the same CRS in an integrated client, rescaling or re-projecting tiles to the common scale of the view might require re-sampling calculations that result in visual quality degradation.

The recommended way to prevent this problem is make use of the same global tile matrix set, for example, one of those available in [\[common-tilematrixset-definitions-informative\]](#). If the coverage of the data covers only a part of the tile matrix set area, the *tile matrix limits* can be used to inform about these limitations.

If using the same tile matrix set is not possible, using a common coordinate reference system and a common set of scales shared by as many layers and services as possible can also be a solution. Thus, the concept of well-known scale set (WKSS) is introduced.

Note that a WKSS only defines a small subset of what is needed to completely define a Tile Matrix Set. A WKSS is an optional feature that does not replace the need to define the Tile Matrix Set and its Tile Matrices. The original purpose of WKSS is no longer necessary if services share and reference common Tile Matrix Sets definitions such as the ones in [\[common-tilematrixset-definitions-informative\]](#).

A WKSS is a commonly used combination of a CRS and a set of scales. A tile matrix set can declare support for a WKSS set by referencing that WKSS. A client application can confirm that tiles in one tile matrix set are compatible with tiles in another tile matrix set merely by verifying that they declare a common WKSS. The informative [\[annex-c-well-known-scale-sets-informative\]](#) provides several WKSSs (and others could be incorporated in the future).

A tile matrix set conforms to a particular WKSS when it uses the same CRS and defines all scale denominators ranging from the largest scale denominator in the WKSS to some low scale denominator (in other words, it is not necessary to define all the lower scale denominators to conform to a WKSS).

6.1.4. Tile based coordinates in a tile matrix set

A tile in a tile-based coordinate can be referred by its tile position in the tile matrix dimensions and the tile matrix identifier in tile matrix set. In a two-dimensional space, a tile is identified by these 3 discrete index names: *tile row*, *tile column* and *tile matrix identifier*.

In raster tiles, a grid cell in the *extrapolated device grid* domain set can be identified by a set of floating point coordinates in the CRS and by one of two ways that does not present rounding issues, as follows.

- By the tile indices the grid cell is contained by (referred by its tile position in the tile matrix dimensions and the Tile Matrix identifier in the Tile Matrix Set) and the cell indices inside the tile (i, j, \dots). In a two-dimensional space, a tile is identified by 5 discrete indices that are named: *tile row*, *tile column*, *tile matrix identifier*, i and j . This is how GetFeatureInfo works in WMTS. This set of coordinates is called “*tile coordinates*.”
- By the position of the cell in grid defined by the *extrapolated device grid* domain set (that starts at the top left corner of the tiled space) of the tile matrix and the identifier of the Tile Matrix in Tile Matrix Set. In a two-dimensional space, a grid cell is identified by 3 discrete indices that are named: i , j and *tile matrix identifier*. Note that i and j can be very big integer numbers and, for very detailed scale, tile matrices might require integer 64-bit notation if stored as binary numbers. This set of indices is called “*tilematrix coordinates*.”

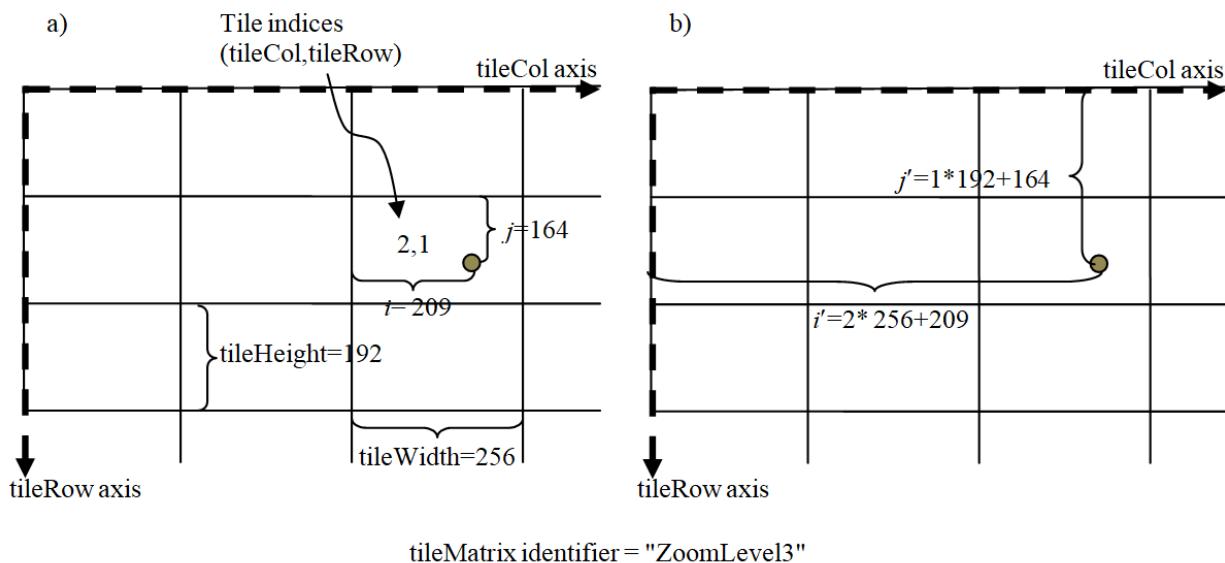


Figure 7. Tile coordinates (a) and Tile matrix coordinates (b) to identify grid cells

6.1.5. Variable width tile matrices

Until now, it has been assumed that *matrixWidth* is constant in for all tile rows. This is common usage for projections that do not distort the Earth too much. But when using Equirectangular Plate Carrée projection (see [common-tilematrixset-definitions-informative] subsection 2) the distortion increases for tiles closer to the poles. In the extreme, the upper row of the upper tile (the one representing the North Pole) contains a list of repeated values that represents almost the same position in the space. The same can be said for the lower row of the lower tile (the one representing the South Pole). When the tiles are represented in a flat projection, this is an effect that cannot be avoided, but when the data are presented in a virtual globe, the distortion results in redundant information in the poles that need to be eliminated by the client during the rendering. It would be better if the distortion is compensated by the server side instead.

The solution consists of reducing the number of tiles (*matrixWidth*) in the high latitude rows and

generating those tiles with a compressed scale in the i dimension (see Figure 5). To allow that, the tile model must be extended to specify coalescence coefficients (c) that reduce the number of tiles in the width direction by aggregating c horizontal tiles but keeping the `tileWidth` (and `tileHeight`). The coalescence coefficient will not be applied next to the Equator but will be used in medium and high latitudes (the higher the latitude the bigger the coefficient).

Even if tiles can coalesce, this does not change the indexing or the tile matrix set that will be the same as if no coalescence has been applied. For example, if the c coefficient is 4, the `tileCol` of the first tile will be 0, the `tileCol` of the second tile will be 4, the `tileCol` of the third tile will be 8 and so on. In other words, and for the same example, `tileCol` 0, 1, 2 and 3 points to the same tile.

NOTE

This decision is necessary to still be able to be able to define a rectangle in the space based on tile indices as we do in tile matrix limits section.

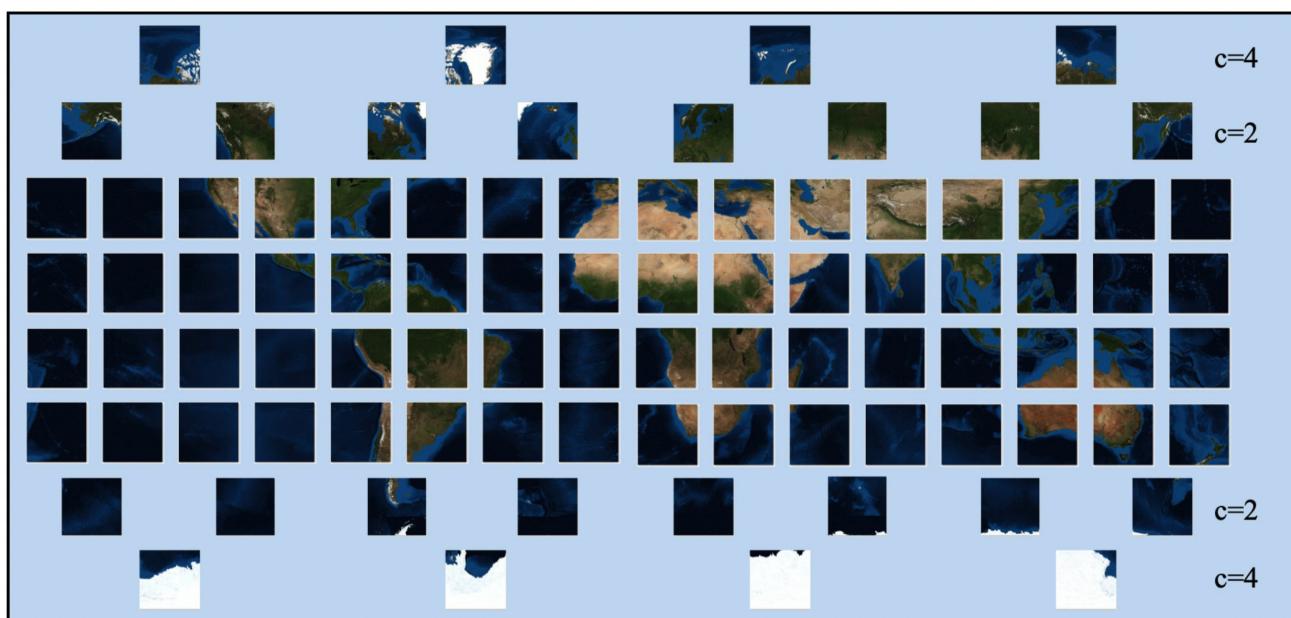


Figure 8. TileMatrix with variable matrix width

6.2. TileMatrixSet Requirements Classes

6.2.1. TileMatrixSet requirements class

Requirements class `tilematrixset` establishes how to describe a TileMatrixSet for a two-dimensional tile space. It is expected that tile matrix sets are defined once and that servers or encodings using or distributing tiles will declare the usage of a tile matrix set by linking to that tile matrix set.

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/tilematrixset	
Target type	tile matrix sets
Dependency	http://www.opengis.net/spec/tile-model/1.0/req/tile-set-scheme

Requirement 31

/req/tilematrixset/model

A tile matrix set SHALL be defined following the UML model as shown in [Figure 5](#) and the model description in [Table 6](#) and [Table 7](#).

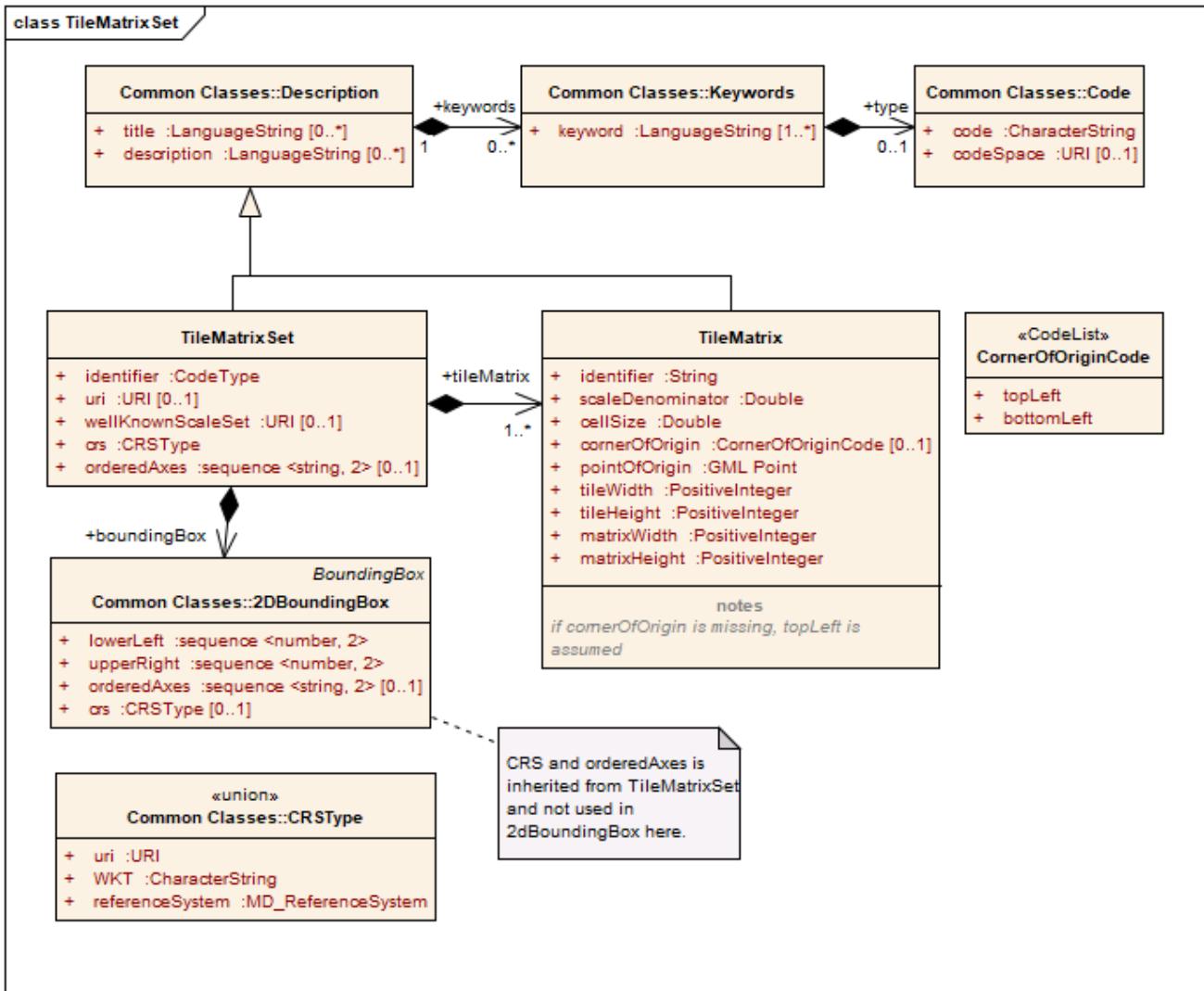


Figure 9. TileMatrixSet UML model

[Table 6](#) defines the structure of the TileMatrixSet.

Names	Definition	Data type and values	Multiplicity and use
identifier	Tile matrix set identifier ^g	CodeType, as adaptation of MD_Identifier class ISO 19115	One (mandatory)

Names	Definition	Data type and values	Multiplicity and use
title ^a	Title of this tile matrix set, normally used for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented ^f
description ^a	Brief narrative description of this tile matrix set, normally available for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented
keywords ^a	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this tile matrix set	MD_Keywords class in ISO 19115. See Table 1	Zero or more (optional) One for each keyword authority used
uri	Reference to an official source for this tileMatrixSet	URI type	Zero or One (optional) Include when an official source exists
crs	Coordinate reference system (CRS) ^j	CRSType type, see Table 4	One (mandatory) ⁱ
orderedAxes	Ordered list of names of the dimensions defined in the CRS ^k	Ordered sequence of strings	Zero or one (optional)
wellKnownScaleSet	Reference to a well-known scale set ^e	URI type ^h	Zero or one (optional) ^c
boundingBox	Minimum bounding rectangle surrounding the tile matrix set, in the crs ^b	2DBoundingBox data structure, see Table 3	Zero or one (optional)
tileMatrix	Description of a scale level and its tile matrix	TileMatrix data structure. See Table 7	One or more (mandatory) ^d

Names	Definition	Data type and values	Multiplicity and use
^a	The multilingual scoping rules in Multilingual text encoding >> apply.		
^b	In the same CRS as the TileMatrixSet, boundingBox should be considered informative about the area covered by this TileMatrixSet. It SHOULD NOT be used to calculate the position of the tiles in the CRS space; please use cornerOfOrigin and pointOfOrigin of the corresponding TileMatrix instead. If data is not available for all tiled space, TileMatrixSetLimits will declare what the tiles have data (see TileMatrixSetLimits requirements class).		
^c	When a tile matrix set conforms to a well-known scale set it can reference it by its URI. If used, the well-known scale set SHALL be consistent with the crs and with the scaleDenominators of the tileMatrix parameters.		
^d	Commonly more than one. Each tileMatrix of a tileMatrixSet SHALL have a unique (different) scaleDenominator.		
^e	Some possible values are defined the in Well-known scale sets (Informative) .		
^f	If no Title is specified, a client may display the Identifier value instead.		
^g	TileMatrixSet identifies SHALL be unique (different) for each TileMatrixSet of a server.		
^h	In WMTS 1.0 a URN was used as a reference to a well-known scale set on the URN in WMTS. Later, OGC adopted URIs (instead of URNs) to do references. The (OGC Naming Authority - Procedures document [https://docs.opengeospatial.org/pol/09-046r5.html#_naming_rule]) specifies rules to transform form URN to URI. Implement a on-the-fly translations based on these rules is possible.		
ⁱ	A single CRS can have several realizations and can be complemented by an epoch. In here it was decided that a TileMatrixSet is defined in a generic CRS. The CRS realization and epoch used by a concrete tileset is specified in the tileset metadata.		
^j	The CRS of the TileSets using this TileMatrixSet should be compatible with this CRS. See TileMatrixSet CRS Compatibility		
^k	This element is not intended to overwrite the CRS axis order but to make it visible to developers by repeating information that is already contained in the CRS definition.		

Table 6. Parts of TileMatrixSet data structure

A part from the general tile matrix set description an array of tile matrix elements is needed to define the distribution of tiles for each scale denominator.

Names	Definition	Data type and values	Multiplicity and use
identifier	Tile matrix identifier ^c	ows:CodeType, as adaptation of MD_Identifier class ISO 19115	One (mandatory)
title ^a	Title of this tile matrix, normally used for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented ^d
description ^a	Brief narrative description of this tile matrix, normally available for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented
keywords ^c	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this tile matrix	MD_Keywords class in ISO 19115. See Table 1	Zero or more (optional) One for each keyword authority used
scaleDenominator	Scale denominator of this tile matrix ^f	Double type	One (mandatory)
cellSize	Cell size of this tile matrix ^f	Double type	One (mandatory)
cornerOfOrigin	Corner of the tile matrix used as the origin for numbering tile rows and columns.	enumeration. See Table 8	Zero or one (optional). Default value is "topLeft"
pointOfOrigin ^e	Position in CRS coordinates of the corner of origin for this tile matrix.	Ordered sequence of double values ^b	One (mandatory)
tileWidth	Width of each tile of this tile matrix in cells	Positive integer type	One (mandatory)
tileHeight	Height of each tile of this tile matrix in cells	Positive integer type	One (mandatory)

Names	Definition	Data type and values	Multiplicity and use
matrixWidth	Width of the matrix (number of tiles in width)	Positive integer type	One (mandatory)
matrixHeight	Height of the matrix (number of tiles in height)	Positive integer type	One (mandatory)

^a The multilingual scoping rules in [Multilingual text encoding](#)>> apply.

^b The CRS and order of these coordinates shall be as specified by the *crs* of the parent *TileMatrixSet*. These are the precise coordinates of the corner of origin (e.g. the top-left corner) for the tile matrix, which is also the corner of the (0, 0) tile. See Figure 1.

^c These *TileMatrix* identifiers SHALL be unique (different) within the context of the parent *TileMatrixSet*. Many applications use a correlative numeric value as an identifier. Other alternatives are a rounded scale denominator or a rounded cell size. It is recommended to avoid repeating the *TileMatrixSet* identifier as part of the *TileMatrix* identifier.

^d If no *Title* is specified, client may display the *Identifier* value instead.

^e In previous versions was called *topLeftCorner* and *cornerOfOrigin* did not exist

^f The cell size of the tile can be obtained from the *scaleDenominator* by multiplying the latter by 0.28×10^{-3} / *metersPerUnit*. If the CRS uses *meters* as units of measure for the horizontal dimensions, then *metersPerUnit*=1; if it uses degrees, then *metersPerUnit*= $2\pi a/360$ (*a* is the Earth maximum radius of the ellipsoid; a.k.a the radius of the equator.).

Table 7. Parts of TileMatrix data structure

- NOTE** It may be desirable to define a tile matrix set with some general-scale tile matrices in one CRS (e.g., CRS:84) and with detailed-scale tile matrices in a different CRS (e.g., LCC projection). However, this standard does not allow mixing CRSs. Each tile matrix set declares a single CRS.
- NOTE** The width (matrixWidth) and height (matrixHeight) in tiles of each tile matrix is explicitly given, so the range of relevant tile indexes does not have to be calculated by the client application.
- NOTE** The bounding box of a tile matrix is not supplied explicitly because it can be calculated from *cornerOfOrigin*, *pointOfOrigin*, *tileWidth*, *tileHeight* and *scaleDenominator*.

Names	Definition
topLeft	Top left corner ^a
bottomLeft	Bottom left corner ^b
^a	The only possibility available in WMTS 1.0. Sometimes known as "xyz" in other non OGC specifications.
^b	Used by Tile Map Service. Sometimes known as "tms" in other non OGC specifications

Table 8. Parts of *CornerOfOriginCode* enumeration

TileMatrixSet CRS Compatibility

In general the CRS of the TileSets using a TileMatrixSet should be the same as the CRS of that TileMatrixSet. However there are some situations where there is some flexibility:

- The CRS of the TileSet is a realization of the datum ensemble CRS specified in the TileMatrixSet. E.g. TileSet CRS: EPSG:9057 (G1762 realization of WGS84) and TileMatrixSet CRS: EPSG:4326 (WGS84)
- The CRS of the TileSet includes additional dimensions beyond the 2 specified in the CRS of the TileMatrixSet. E.g. TileSet CRS: EPSG:4979 (lat, lon, h in WGS84) and TileMatrixSet CRS: EPSG:4326 (lat, lon in WGS84)
- The CRS of the TileSet and the CRS of the TileMatrixSet only differ in axes ordering. E.g. TileSet CRS: EPSG:4326 (lat, lon) and TileMatrixSet CRS: CRS84 (lon, lat). The order specified in the TileMatrixSet only affects the coordinates order for the PointOfOrigin and optional BoundingBox within the TileMatrixSet definition itself. The CRS of the TileSet affects the coordinates order for the data in the tiles, if applicable (e.g some formats of vector tiles)

6.2.2. VariableMatrixWidth requirements class

This requirements class provides the necessary support for variable matrix width tile matrix sets.

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/variablematrixwidth	
Target type	tile matrix sets
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixset

Requirement 32 /req/variablematrixwidth/model

When a tiled resource or dataset has variable width tiles, the resource *or dataset shall* define the variable matrix width in a tile matrix set using a variablematrixwidth data structure. This will result in the following UML model shown in [Figure 10](#) and model description in [Table 9](#).

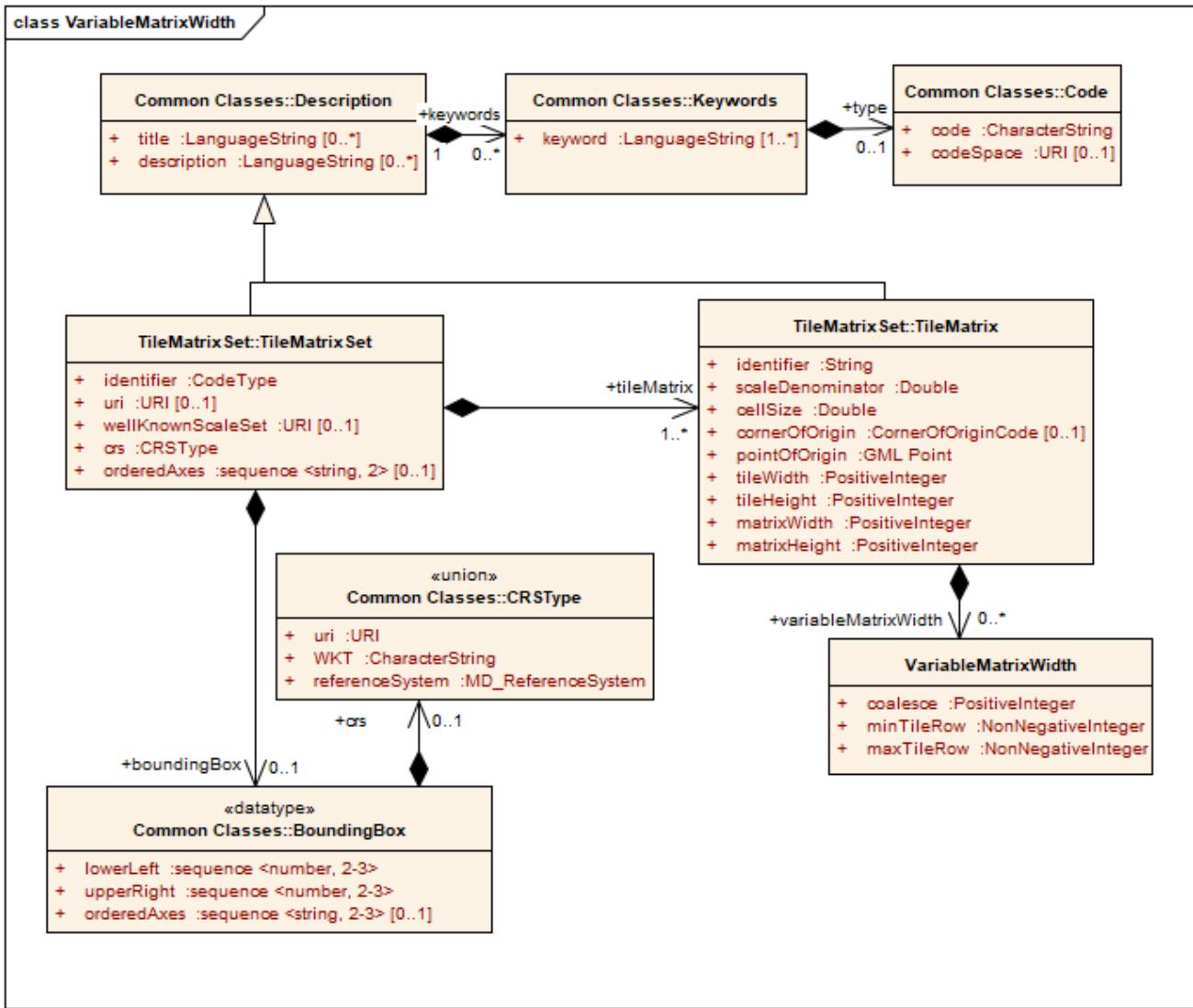


Figure 10. VariableMatrixWidth UML model

In order to make the description of the model more compact, only the tile rows that have coalesced (i.e., coalescence factor larger than 1) will be encoded.

Names	Definition	Data type and values	Multiplicity and use
coalesce	Coalescence factor	Positive integer type ^a	One (mandatory)
minTileRow	First tile row where the coalescence factor applies for this tilematrix	Non negative integer type ^b	One (mandatory)
maxTileRow	Last tile row where the coalescence factor applies for this tilematrix	Non negative integer type ^c	One (mandatory)

Names	Definition	Data type and values	Multiplicity and use
^a	Shall be more than 1. Rows with Coalescence factor of 1 shall not be described here.		
^b	From 0 to maxTileRow.		
^c	From minTileRow to matrixWidth-1 of the tileMatrix section of this tileMatrixSet.		

Table 9. Parts of VariableMatrixWidth data structure

Requirement 33	/req/variablematrixwidth/coalescence1
	Only the tile rows with a coalescence factor different from one shall be encoded. If a tile row is not mentioned in the VariableMatrixWidth description a coalescence factor of 1 shall be considered for that row.

Chapter 7. TileMatrixSet encodings

7.1. JSON encoding

The JSON encoding has been done respecting the naming and types of the original classes. However, there are some differences. [Table 10](#) describes some exceptions in the mapping between the class attributes and corresponding JSON element.

Class attribute	JSON element	Reason
identifier	id	Following common practice in GeoJSON and JSON-LD.
BoundingBox class	array of 4 numbers representing lowerLeft and topRight coordinates	Following GeoJSON suggested encoding.
LanguageString	string	JSON files are generally used in the OGC API as monolingual documents. The language of the document is determined by the HTTP language negotiation headers.
keywords	string array	JSON files are general used in the OGC API as monolingual documents. JSON does not use namespaces or codespaces.
	WKT	object

Table 10. propertiesSchema attributes and JSON Schema properties equivalences

In addition, all elements with multiplicity greater than one changes names to plural. Please be aware of irregular plurals such as [tileMatrix](#) that changes to [tileMatrices](#).

7.1.1. JSON TileMatrixSet requirements class

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/json-tilematrixset	
Target type	tile matrix sets
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixset

Requirement 34	/req/json-tilematrixset/model
A TileMatrixSet encoded in JSON <i>shall</i> implement the class TileMatrixSet	

Requirement 35**/req/json-tilematrixset/ietf**

A TileMatrixSet encoded in JSON *shall* conform to IETF RFC7159

Requirement 36**/req/json-tilematrixset/schema**

A TileMatrixSet encoded in JSON *shall* validate using the JSON schema for a tile matrix set (<http://schemas.opengis.net/tms/2.0/json/tileMatrixSet.json>).

Requirement 37**/req/json-tilematrixset/media-type**

A TileMatrixSet encoded in an independent JSON document *shall* use the media type application/json.

NOTE

A TileMatrixSet description can be embedded in other file formats, such as a Service Metadata document of a WMTS service. In this case, the media type of the containing document prevails.

An annex provides [Example encodings for Common TileMatrixSets \(Informative\)](#) in JSON.

7.1.2. JSON VariableMatrixWidth requirements class

Requirements Class<http://www.opengis.net/spec/tms/2.0/req/variablematrixwidth>

Target type	tile matrix sets
Dependency	http://www.opengis.net/spec/tms/2.0/req/variablematrixwidth
Dependency	http://www.opengis.net/spec/tms/2.0/req/json_tilematrixset

Requirement 38**/req/json-variablematrixwidth/model**

A VariableMatrixWidth encoded in JSON *shall* implement the class VariableMatrixWidth.

Requirement 39**/req/json-variablematrixwidth/ietf**

A VariableMatrixWidth encoded in JSON *shall* conform to IETF RFC7159

Requirement 40**/req/json-variablematrixwidth/schema**

A VariableMatrixWidth encoded in JSON *shall* validate using the JSON schema for a variable matrix width (<http://schemas.opengis.net/tms/2.0/json/tilematrixset.json#/definitions/variableMatrixWidth>)

An annex provides [Example encodings for Variable Matrix Width TileMatrixSets \(Informative\)](#) in JSON.

7.2. XML encoding

The XML encoding has been done respecting the naming and types of the original classes. However all attribute names have been transformed to UpperCamelCase.

7.2.1. XML TileMatrixSet requirements class

Requirements Class

<http://www.opengis.net/spec/tms/2.0/req/xml-tilematrixset>

Target type	tile matrix sets
-------------	------------------

Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixset
------------	---

Requirement 41 /req/xml-tilematrixset/model

A TileMatrixSet encoded in XML shall implement the class TileMatrixSet

Dependency	/req/tilematrixset
------------	--------------------

Requirement 42 /req/xml-tilematrixset/schema

A TileMatrixSet encoded in XML *shall* validate using the XML schema for a tile matrix set.

Requirement 43 /req/xml-tilematrixset/media-type

A TileMatrixSet encoded in an independent XML document shall use the media type application/xml.

NOTE

A TileMatrixSet description can be embedded in other file formats, such as a Service Metadata document of a WMTS service. In this case, the media type of the containing document prevails.

An annex provides [Example encodings for Common TileMatrixSets \(Informative\)](#) in XML.

7.2.2. XML VariableMatrixWidth requirements class

Requirements Class

<http://www.opengis.net/spec/tms/2.0/req/xml-variablematrixwidth>

Target type	tile matrix sets
-------------	------------------

Dependency	http://www.opengis.net/spec/tms/2.0/req/variablematrixwidth
------------	---

	Dependency
--	------------

Requirement 44 /req/xml-variablematrixwidth/model

A VariableMatrixWidth encoded in XML *shall* implement the class VariableMatrixWidth

Dependency: /req/variablematrixwidth

Dependency: /req/xml-tilematrixset

Requirement 45 **/req/xml-variablematrixwidth/schema**

A VariableMatrixWidth encoded in XML *shall* validate using the XML schema for a variable matrix width.

An annex provides [Example encodings for Variable Matrix Width TileMatrixSets \(Informative\)](#) in XML.

Chapter 8. TileSetMetadata

8.1. Overview

8.1.1. TileMatrixSet limits

If the tile matrix set for a dataset covering a bounding box defines the extreme corner adjusted to the actual content of this dataset, and later the bounding box needs to be extended, then the extreme corner of each TileMatrix will change, which will change the tile indices of any previous tile invalidating any previously cached tile. To overcome this problem, a dataset can optionally use a more generic TileMatrixSet that covers a bigger (or even global) area. In fact, that TileMatrixSet that defines an area that might be covered by the dataset in a future could easily be shared for many datasets and become a common TileMatrixSet.

To inform the client about the valid range of tile indices, the TileMatrixSetLimits concept is introduced. A list of TileMatrixSetLimit informs the minimum and a maximum limits of these indices for each TileMatrix that contains actual data. The area outside these limits is considered empty space.

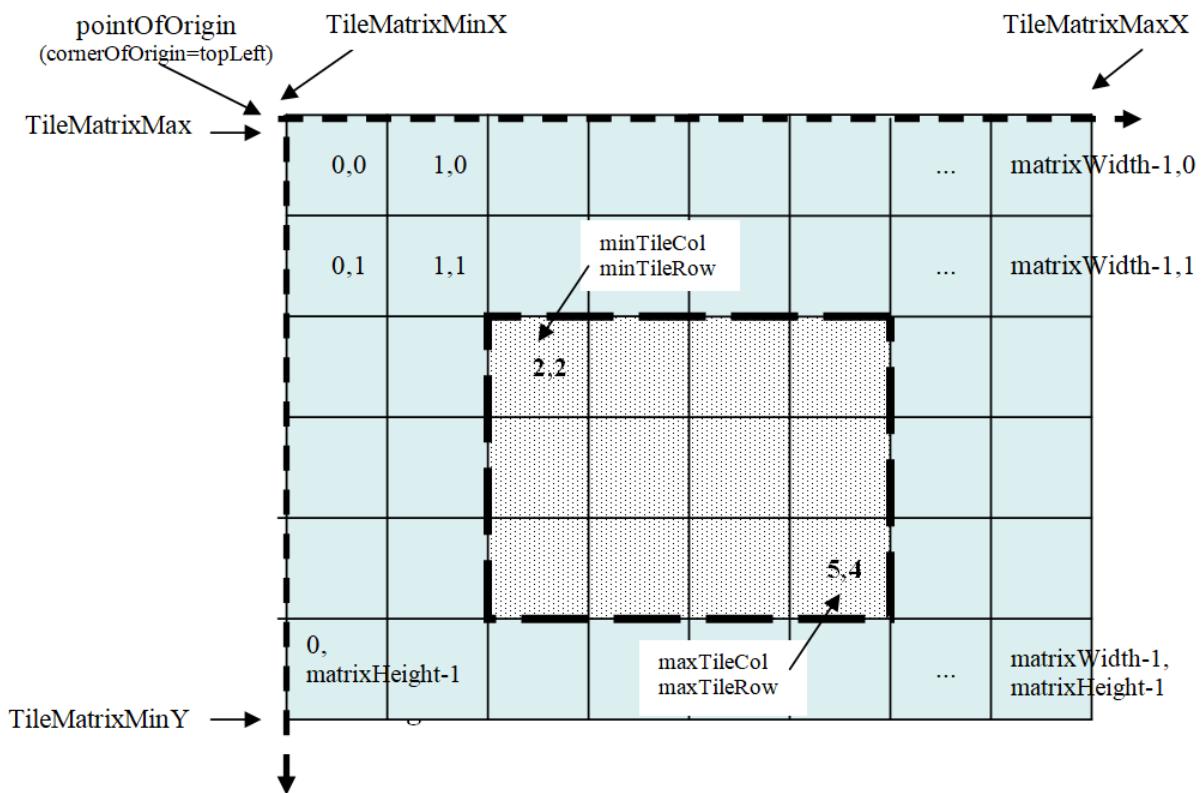


Figure 11. TileMatrixSet Limits

8.1.2. TileSet metadata

Tiles are identified by tileMatrix id, tileRow number and tileCol number. These three elements only have meaning if they are associated to a tileMatrixSet description that contains the necessary information (in terms of scaleDenominator, cellSize, pointOfOrigin and cornerOfOrigin)

to transform the indices into coordinates in a known CRS. The main purpose of the TileSetMetadata is to link the tileset with the tileMatrixSet description. In addition, the model contains elements describing the main characteristics of a tile set, the connection to the tileset with the original data collection and styles as well as a recommended center point to start the navigation.

8.2. Requirements classes

8.2.1. TileMatrixSetLimits requirements class

Requirements class tilematrixsetlimits establishes how to describe the limits for a tile set TileMatrixSet. It is expected that many TileMatrixSets are defined only once and reused many times. In these circumstances, the data used to create the tile set may only exist for a partial region or for a subset of scales. The array of TileMatrixSetLimits data structures allows for the declaration of a limited coverage of a tile matrix set. The identifying URI for this class is <http://www.opengis.net/spec/tms/2.0/req/tilematrixsetlimits>.

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/tilematrixsetlimits	
Target type	tile set metadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixset

Requirement 46	/req/tilematrixsetlimits/model
A	A tile matrix set limits SHALL consists of an array of Table 12 data structures with a multiplicity equal or inferior to the multiplicity of the tileMatrix of this tileMatrixSet and an optional boundingBox, as defined in the UML model shown in Figure 12
B	Each tileMatrix identifier SHALL be mentioned only once in this TileMatrixSetLimits. If a tileMatrix identifier is not mentioned, it should be interpreted as a tileMatrix that is not available.

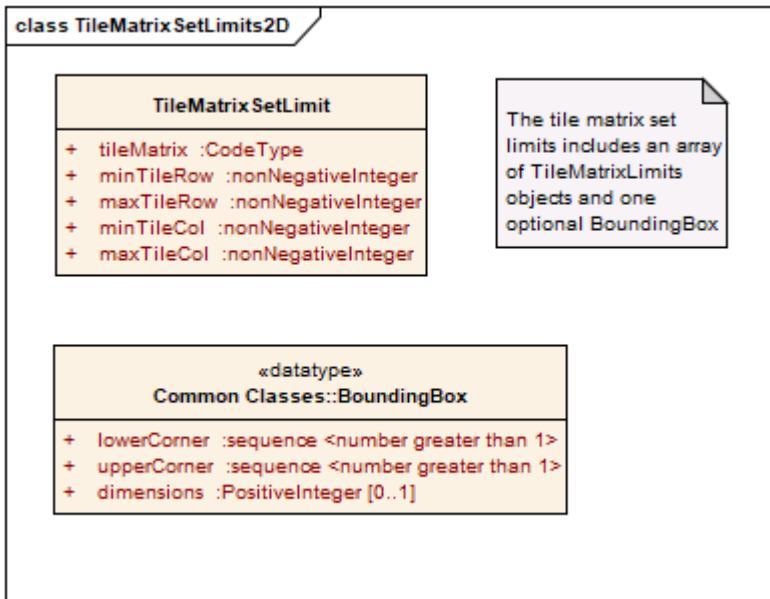


Figure 12. *TileMatrixLimit array UML model*

Names	Definition	Data type and values	Multiplicity and use
tileMatrixSetLimit	Indices limits for this tileMatrix	TileMatrixSetLimits data structure, see Table 12	one or more (mandatory) ^a

Table 11. *TileMatrixSetLimits array*

Names	Definition	Data type and values	Multiplicity and use
tileMatrix	Reference to a tileMatrix identifier	ows:CodeType, as adaptation of MD_Identifier class ISO 19115 ^a	One (mandatory)
minTileRow	Minimum tile row index valid for this layer.	Non negative integer type ^b	One (mandatory)
maxTileRow	Maximim tile row index valid for this layer.	Non negative integer type ^c	One (mandatory)
minTileCol	Minimum tile column index valid for this tile column.	Non negative integer type ^d	One (mandatory)
maxTileCol	Maximum tile column index valid for this tile column.	Non negative integer type ^e	One (mandatory)

Names	Definition	Data type and values	Multiplicity and use
^a	SHALL be an identifier to a tileMatrix element of this tileMatrixSet.		
^b	From 0 to maxTileRow.		
^c	From minTileRow to matrixWidth-1 of the tileMatrix of this tileMatrixSet.		
^d	From 0 to maxTileCol.		
^e	From minTileCol to tileHeight-1 of the tileMatrix of this tileMatrixSet.		

Table 12. Parts of TileMatrixLimit data structure

8.2.2. TileSetMetadata requirements class

Requirements class TileSetMetadata establishes how to describe TileSet Metadata for a two-dimensional tile space. The TileSetMetadata data structure allows for a dataset declaring the use of a tile matrix set defined elsewhere and, if needed, a limited coverage for this tile matrix set, the list of geodata resources used to create the tileset and a recommended center point. Each TileSet in a dataset collection should declare the use of a tile matrix set using this data structure. The identifying URI for this class is <http://www.opengis.net/spec/tms/2.0/req/tilesetmetadata>

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/tilesetmetadata	
Target type	tile set metadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixset
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixsetlimits

Requirement 47	/req/tilesetmetadata/identifier
A	A tiled resource or dataset SHALL declare support to a tile matrix set by one of the following two methods: a link to a tile matrix set definition (including the TileMatrixSet definition from Common TileMatrixSet definitions (Informative) and Example encodings for Variable Matrix Width TileMatrixSets (Informative)) as one of the links in the links list or by explicitly including a TileMatrixSet definition (as an object in the tileMatrixSet property).

B	In an OGC API or an OGC Web Service and if the TileMatrixSet used is registered with the OGC NA definition service, tileMatrixSetURI SHALL be populated with a full URI to the OGC NA Definitions Server (an optional tileMatrixSetDefinition pointing to a local definition can also be provided)
---	--

NOTE The **tileMatrixSet** property with an object describing the TileMatrixSet is not intended for OGC APIs or OGC web services and it is included here for offline formats and encodings where links to resources are not possible or sensible.

NOTE To determine if two resources or datasets use the same TileMatrixSet, compare their TileMatrixSet identifier. Alternatively, compare TileMatrixSet definitions for an equivalency (a simple calculation can be performed to verify whether or not two given tile matrices are aligned).

NOTE If the same TileMatrixSet is externally available in more than one format, it is recommended that the format selected is the closer to the original description document format. For example if a OGC API defines tiles using JSON, it is expected to link to a JSON definition of a TileMatrixSet.

Requirement 48	/req/tilessetmetadata/model
A	When a tiled resource or dataset has tiles available, the metadata describing the tiles (and optionally additional information such as the recommended center point and the data from which it was sourced) SHALL be described using the data model in UML model shown in Figure 13 and model description in Table 13 .
B	When a tiled resource or dataset has tiles available only for a region or regions of the complete tiled space, the resource or dataset SHALL declare partial support to a tile matrix set using one or more tile matrix limits data structures defined in Table 12 .
Dependency	/req/tilematrixset /req/tilematrixsetlimits

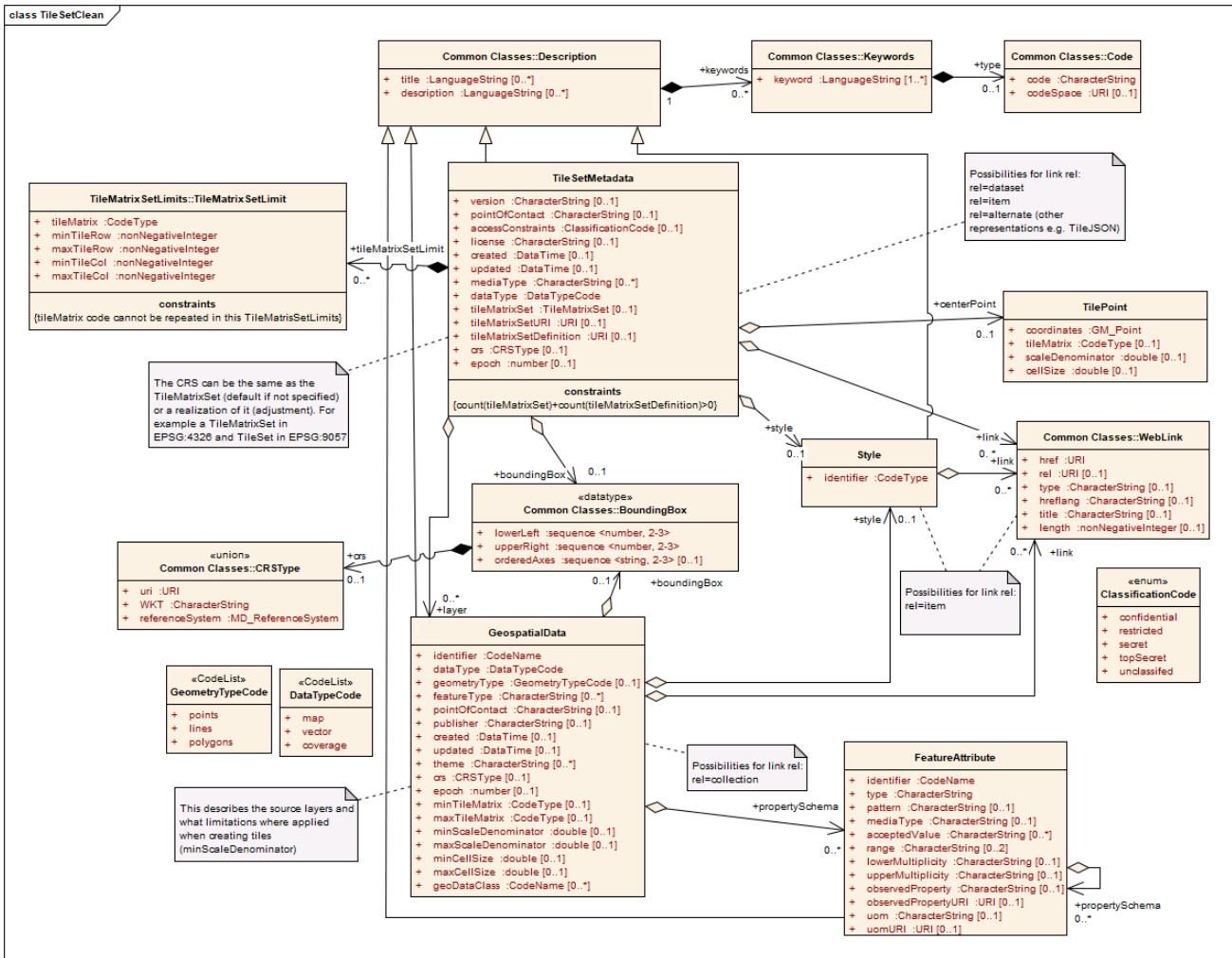


Figure 13. TileSetMetadata UML model

Names	Definition	Data type and values	Multiplicity and use
title ^a	Title of this tile matrix set, normally used for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented
description ^a	Brief narrative description of this tile matrix set, normally available for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented

Names	Definition	Data type and values	Multiplicity and use
keywords ^a	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this tile set	MD_Keywords class in ISO 19115 See Table 1	Zero or more (optional) One for each keyword authority used
version	Version of the Tile Set. Changes if the data behind the tiles has been changed	CharacterString	Zero or one (optional)
pointOfContact	Useful information to contact the authors or custodians for the Tile Set	CharacterString (e.g. e-mail address, a physical address, phone numbers, etc)	Zero or one (optional)
accessConstraints	Restrictions on the availability of the Tile Set that the user needs to be aware of before using or redistributing the Tile Set	ClassificationCode code list, see Table 17	Zero or one (optional)
license	License applicable to the tiles	CharacterString	Zero or one (optional)
mediaType	Media types available for the tiles	CharacterString restricted by RFC6838 section 4.2	Zero or more (optional) ^f
dataType	Type of data represented in the tiles	DataTypeCode code list	one (mandatory)
crs	Coordinate Reference System (CRS) ^j	CRSType type, see Table 4	Zero or one (optional)
epoch	Epoch of the Coordinate Reference System (CRS)	Number	Zero or one (optional)
tileMatrixSet	Tile matrix set definition	TileMatrixSet data strcuture. See Table 6	Zero or one (optional) ^d
tileMatrixSetURI	Reference to an official source for this tileMatrixSet	URI type ^b	Zero or One (optional) Include if this is one of the OGC NA TileMatrixSets

Names	Definition	Data type and values	Multiplicity and use
tileMatrixSetLimit	Limits for the TileRow and TileCol values for each TileMatrix in the tileMatrixSet	TileMatrixSetLimits data structure, see Table 11	Zero or more (optional) Should be include when the boundary of the data is a fragment of the boundary of the tileMatrixSet ^g
boundingBox	Minimum bounding rectangle surrounding the tileset	BoundingBox data structure, see Table 3^h	Zero or one (optional)
created	Timestamp indicating when the Tile Set was first produced	DateTime	Zero or one (optional)
updated	Timestamp of the last Tile Set change/revision	DateTime	Zero or one (optional)
layer	Layer elements represented in the tile set	GeospatialData data structure, see Table 14	Zero or more (optional)
style	Style used to generate the tiles in the tileset	Style data structure, see Table 16	Zero or one (optional) ⁱ
centerPoint	Location of a tile that nicely represents the tileset. Implementations may use this center value to set the default location or to present a representative tile in a user interface	TilePoint data structure, see Table 20.	Zero or more (optional)
link	Links to related resources	WebLink data structure, see [parts-of-atom-link-data-structure] . ^{d,e}	Zero or more (optional)

Names	Definition	Data type and values	Multiplicity and use
^a	The multilingual scoping rules in Multilingual text encoding >> apply.		
^b	Points to a definition of the TileMatrixSet in the OGC NA definition server (http://www.opengis.net/def/tms/) (including definitions from Common TileMatrixSet definitions (Informative) and Variable width TileMatrixSet definitions (Informative))		
^d	At least one of the TileMatrixSet, or a link with <code>rel=tiling-scheme</code> SHALL be provided.		
^e	Possible link 'rel' values are: 'dataset' for a URL pointing to the dataset, 'tiles' for a URL template to get the tiles; 'alternate' for a URL pointing to another representation of the TileSetMetadata (e.g a TileJSON file); 'tiling-scheme' for a definition of the TileMatrixSet.		
^f	Intended for offline use. In an online use you are supposed to provide links to the tiles that already have the mediaType specified.		
^g	If missing, there are no limits other than the ones imposed by the TileMatrixSet. If present the TileMatrices listed are limited and the rest not available at all.		
^h	If the bounding box does not specify a CRS, it is inherited from the CRS of the TileSet if available, otherwise from the TileMatrixSet.		
ⁱ	If style property mentions a style applied to all layers, the style property in layer should not be used.		
^j	It should be compatible with the CRS of the TileMatrixSet. In case the axis order is different from the TileMatrixSet the order of the CRS defined here prevails. See TileMatrixSet CRS Compatibility		

Table 13. Parts of TileSetMetadata data structure

A Layer can is a set of geographic objects (all of the same type) together in a way that can be presented to the user. IT can also be a coverage. Its elements are defined with a data structure defined in [Table 14](#).

Names	Definition	Data type and values	Multiplicity and use
title ^a	Title of this tile matrix set, normally used for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented

Names	Definition	Data type and values	Multiplicity and use
description ^a	Brief narrative description of this tile matrix set, normally available for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented
keywords ^a	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this layer	MD_Keywords class in ISO 19115 See Table 1	Zero or more (optional) One for each keyword authority used
identifier	Unique identifier of the Layer	CodeType, as adaptation of MD_Identifier class ISO 19115	One (mandatory)
dataType	Type of data represented in the layer	DataTypeCode code list	one (mandatory)
geometryType	The geometry type of the features shown in this layer	GeometryTypeCode code list	Zero or more (optional)
featureType	Feature type identifier	CharacterString	Zero or one (optional) ⁱ
pointOfContact	Useful information to contact the authors or custodians for the Layer	CharacterString (e.g. e-mail address, a physical address, phone numbers, etc)	Zero or one (optional)
publisher	Organization or individual responsible for making the layer available	CharacterString	Zero or one (optional)
theme	Category where the layer can be grouped	CharacterString	Zero or more (optional)
crs	Coordinate Reference System (CRS)	CRSType type, see Table 4	Zero or more (optional)
minScaleDenominator	Minimum scale denominator for usage of the layer	double ^c	Zero or one (optional) ^e
minCellSize	Minimum cell size for usage of the layer	double ^c	Zero or one (optional) ^e

Names	Definition	Data type and values	Multiplicity and use
maxScaleDenominator	Maximum cell size for usage of the layer	double ^c	Zero or one (optional) ^f
maxCellSize	Maximum scale denominator for usage of the layer	double ^c	Zero or one (optional) ^f
maxTileMatrix	TileMatrix identifier associated with the minScaleDenominator	CharacterString ^d	Zero or one (optional) ^e
minTileMatrix	TileMatrix identifier associated with the maxScaleDenominator	CharacterString ^d	Zero or one (optional) ^f
boundingBox	Minimum bounding rectangle surrounding the layer ^g	BoundingBox data structure, see Table 3	Zero or one (optional)
created	Timestamp indicating when the layer was first produced	DateTime	Zero or one (optional)
updated	Timestamp of the last layer change/revision	DateTime	Zero or one (optional)
style	Style applied to this layer to generate the tiles in the tileset	Style data structure, see Table 16	Zero or one (optional) ^b
geoDataClass	URI identifying a class of data contained in this layer (useful to determine compatibility with styles or processes)	CodeType, as adaptation of MD_Identifier class ISO 19115	Zero or more (optional)
propertySchema	Properties represented by the features in this layer. Can be the attributes of a feature dataset (datatype=geometries) or the rangeType of a coverage (datatype=coverage)	FeatureAttribute data structure. See Table 15	Zero or more (optional)
link	Links to related resources	WebLink data structure, see [parts-of-atom-link-data-structure] . ^h	Zero or more (optional)

Names	Definition	Data type and values	Multiplicity and use
^a	The multilingual scoping rules in Multilingual text encoding >> apply		
^b	If the tileSetMetadata style property mentions a style applied to all layers, this should be omitted		
^c	SHALL be an scaleDenominator defined in one of the TileMatrix of the TileMatrixSet		
^d	SHALL be an identifier to a tileMatrix element of this tileMatrixSet		
^e	If minCellSize, minScaleDenominator and maxTileMatrix are provided they SHALL be related to the same TileMatrix		
^f	If maxCellSize, maxScaleDenominator and minTileMatrix are provided they SHALL be related to the same TileMatrix		
^g	In the same CRS than the TileMatrixSet		
^h	Possible link 'rel' values are: 'collection' for a URL pointing to the collection represented in this layer		
ⁱ	Only applicable to layers of datatype='geometries'		

Table 14. Parts of GeospatialData data structure

A FeatureAttribute element contains attributes that can be found in at least one feature belonging to the layer the FeatureAttribute element belongs to. Its elements are defined in [Table 15](#).

Names	Definition	Data type and values	Multiplicity and use
title ^a	Title of this tile matrix set, normally used for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented ^f
description ^a	Brief narrative description of this tile matrix set, normally available for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented

Names	Definition	Data type and values	Multiplicity and use
keywords ^a	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this attribute	MD_Keywords class in ISO 19115. See Table 1	Zero or more (optional) One for each keyword authority used
identifier	Identifier of the attribute	CodeType, as adaptation of MD_Identifier class ISO 19115	One (mandatory)
type	The data type of the attribute	CharacterString	One (mandatory)
pattern	Regular expression to validate the values of the attribute	CharacterString	Zero or one (optional) ^b
mediaType	Encodings of a complex attribute (e.g. image/png)	CharacterString restricted by RFC6838 section 4.2	Zero or one (optional) ^b
acceptedValues	Valid values of the attribute	CharacterString	Zero or more (optional) ^b
range	Range of valid values expressed as an array of two items	CharacterString	Zero or two (optional) ^b
lowerMultiplicity	Lower multiplicity of the attribute	Non negative integer	Zero or one (optional) ^c
upperMultiplicity	Upper Multiplicity of the property	Non negative integer. Use '*' for 'unbounded'	Zero or one (optional) ^d
observedProperty	Measured phenomenon (variable) label, commonly a descriptive name	CharacterString	Zero or one (optional)
observedPropertyURI	URI pointing to a representation of the definition of the measured phenomenon (variable)	URI	Zero or one (optional)
uom	Units of measure characterizing the values of the attribute	CharacterString	Zero or one (optional)

Names	Definition	Data type and values	Multiplicity and use
uomURI	URI pointing to a representation of the definition of the units of measure characterizing the values of the attribute	CharacterString	Zero or one (optional)

^a The multilingual scoping rules in [Multilingual text encoding](#)>> apply.

^b If missing all values compatible with the other restrictions are accepted

^c If missing, 0 (optional) is assumed

^d If missing, many (unbounded) is assumed

Table 15. Parts of FeatureAttribute data structure

The style structure applicable to the layer is defined in [Table 16](#).

Names	Definition	Data type and values	Multiplicity and use
title ^a	Title of this style, normally used for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented ^f
description ^a	Brief narrative description of this style, normally available for display to a human	LanguageString data structure. See Table 1	Zero or more (optional) Include when available and useful Include one for each language represented
keywords ^a	Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this style	MD_Keywords class in ISO 19115. See Table 1	Zero or more (optional) One for each keyword authority used
identifier	Identifier of a style	CodeType, as adaptation of MD_Identifier class ISO 19115	One (mandatory)

Names	Definition	Data type and values	Multiplicity and use
link	Links to style related resources	WebLink data structure, see [parts-of-atom-link-data-structure]. ^b	Zero or more (optional)

^a The multilingual scoping rules in [Multilingual text encoding](#)>> apply

Table 16. Parts of Style data structure

The levels of classification applicable to the tile set is defined in [Table 17](#).

Names	Definition
unclassified	Available for general disclosure
restricted	Not for general disclosure
confidential	Available for someone who can be entrusted with information
secret	Kept or meant to be kept private, unknown, or hidden from all but a select group of people
topSecret	Of the highest secrecy

Table 17. Parts of ClassificationCode code list

The data type applicable to the tileset or the layer is defined in [Table 18](#).

Names	Definition
map	Images representing colors for pictorial representation on the screen
vector	Vector based elements. E.g. Features composed by geometries and properties
coverage	Coverage rangeset. E.g. Arrays of values representing physical quantities that are function of the position in a regular grid.

Table 18. Parts of DataTypeCode code list

The geometry type applicable to the geometries of the layer is defined in [Table 19](#).

Names	Definition
points	0D geometries
lines	1D geometries
polygons	2D geometries

Table 19. Parts of GeometryTypeCode enumeration

A center point is a place and scale that results in a tile that is representative of the tileset. A tile that contains some variety of objects, is visually appealing and easy to understand will be selected. The center point data structure applicable to the tile set is defined in [Table 20](#).

Names	Definition	Data type and values	Multiplicity and use
coordinates	Location of the center point in the tile set	GM_Point data structure	one (mandatory)
tileMatrixId	Tile matrix identifier of the tile	CodeType, as adaptation of MD_Identifier class ISO 19115 ^a	One (mandatory)
scaleDenominator	Scale denominator of the tile	double ^c	Zero or one (optional) ^e
cellSize	Cell size of the tile	double ^c	Zero or one (optional) ^e ^a SHALL be an identifier to a tileMatrix in relevant tileMatrixSet. ^b If cellSize and scaleDenominator are provided they SHALL be related to TileMatrix with the indicated tileMatrixId

Table 20. Parts of TilePoint data structure

Chapter 9. TileSetMetadata encodings

9.1. JSON encoding

The JSON encoding has been done respecting the naming and types of the original classes.

9.1.1. JSON TileMatrixSetLimits requirements class

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/json-tilematrixsetlimits	
Target type	tile set metadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixsetlimits

Requirement 49	/req/json-tilematrixsetlimits/model
A TileMatrixSetLimits encoded in JSON <i>shall</i> implement the class TileMatrixSetLimits	
Dependency:	/req/tilematrixsetlimits

Requirement 50	/req/json-tilematrixsetlimits/ietf
A TileMatrixSetLimits encoded in JSON <i>shall</i> conform to IETF RFC7159	

Requirement 51	/req/json-tilematrixsetlimits/schema
A TileMatrixSetLimits encoded in JSON shall validate using the JSON schemas for a tile matrix set limits.	

NOTE

A TileMatrixSetLimits description can be embedded within the TileSetMetadata, as for the OGC API - Tiles tileset conformance class.

An annex provides an [Example JSON encoding of TileSetMetadata](#) including TileMatrixSetLimits.

9.1.2. JSON TileSetMetadata requirements class

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/json-tilesetmetadata	
Target type	tile set metadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilesetmetadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/json-tilematrixsetlimits

Requirement 52	/req/json-tilesetmetadata/model

A TileSetMetadata encoded in JSON *shall* implement the class TileSetMetadata

Requirement 53 **/req/json-tilesetmetadata/ietf**

A TileSetMetadata encoded in JSON *shall* conform to IETF RFC7159

Requirement 54 **/req/json-tilesetmetadata/schema**

A TileSetMetadata encoded in JSON *shall* validate using the JSON schema for a TileSetMetadata (<http://schemas.opengis.net/tms/2.0/json/tileSet.json>).

Requirement 55 **/req/json-tilesetmetadata/media-type**

A TileSetMetadata encoded in an independent JSON document *shall* use the media type application/json.

NOTE

A TileSetMetadata description can be embedded in other file formats, such as a Service Metadata document of a WMTS service. In this case, the media type of the containing document prevails.

An annex provides an [Example JSON encoding of TileSetMetadata](#).

The encoding of PropertySchema adopts the JSON Schema form starting by and element propertiesSchema that defines the GeoJSON element "properties" for this layer (the composition of "properties" is undefined by GeoJSON but it can be define for a layer instance if its features follow a data model). [Table 21](#) describes the mapping between the FeatureAttribute class attributes (in [Table 15](#)) and the JSON schema properties.

.propertiesSchema attributes	JSON Schema properties
title	title
keywords	N/A
identifier	name of the additional property
type	combination of 'type' and 'format'
pattern	pattern
mediaType	contentMediaType
acceptedValues	enum
range	a combination of 'maximum', 'exclusiveMaximum', 'minimum' and 'exclusiveMinimum'
lowerMultiplicity	minItems
upperMultiplicity	maxItems
observedProperty	N/A

.propertiesSchema attributes	JSON Schema properties
observedPropertyURI	N/A
uom	N/A
uomURI	N/A

Table 21. *propertiesSchema attributes and JSON Schema properties equivalences*

9.2. XML encoding

The XML encoding has been done respecting the naming and types of the original classes. However all attribute names have been transformed to UpperCamelCase.

9.2.1. XML TileMatrixSetLimits requirements class

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/xml-tilematrixsetlimits	
Target type	tile set metadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilematrixsetlimits

Requirement 56	/req/xml-tilematrixsetlimits/model
A TileMatrixSetLimits encoded in XML SHALL implement the class TileMatrixSetLimits	
Dependency:	/req/tilematrixsetlimits

Requirement 57	/req/xml-tilematrixsetlimits/schema
A TileMatrixSetLimits encoded in XML shall validate using the XML schemas for a tile matrix set limits.	

NOTE

A TileMatrixSetLimits is normally used as embedded in other XML documents. That is the reason an associated media type is not provided.

An annex provides an [Example XML encoding of TileSetMetadata](#) including TileMatrixSetLimits.

9.2.2. XML TileSetMetadata requirements class

Requirements Class	
http://www.opengis.net/spec/tms/2.0/req/xml-tilessetmetadata	
Target type	tile set metadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/tilessetmetadata
Dependency	http://www.opengis.net/spec/tms/2.0/req/xml-tilematrixsetlimits

Requirement 58**/req/xml-tilesetmetadata/model**

A TileSetMetadata encoded in XML shall implement the class TileSetMetadata

Requirement 59**/req/xml-tilesetmetadata/schema**

A TileSetMetadata encoded in XML shall validate using the XML schema for a TileSetMetadata.

Requirement 60**/req/xml-tilesetmetadata/media-type**

A TileSetMetadata encoded in an independent XML document shall use the media type application/xml.

NOTE

A TileSetMetadata does not currently have an associated media type.

An annex provides an [Example XML encoding of TileSetMetadata](#).

Annex A: Conformance Class Abstract Test Suite (Normative)

A implementation of this standard must satisfy the following system characteristics to be conformant with this specification.

A.1. Conformance Class TileMatrixSet

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/tilematrixset	
Target type	tile matrix sets
Requirements class	Requirements Class "TileMatrixSet"

A.1.1. Model

Abstract Test 31	/conf/tilematrixset/model
Test Purpose	Validated that a tile matrix set follows the UML model and the model description in tables
Requirement	[req_tilematrixset_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSet instances point to the TileMatrixSet data type definition and follow the data model specified in the Tables related to the requirement and its dependencies.</p>

A.2. Conformance Class VariableMatrixWidth

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/variablematrixwidth	
Target type	tile matrix sets
Requirements class	Requirements Class "VariableMatrixWidth"

A.2.1. Model

Abstract Test 32	/conf/variablematrixwidth/model
-------------------------	---------------------------------

Test Purpose	Validate that a tiled resource with variable width tiles, the resource defines the variable matrix width in a tile matrix set using a variablematrixwidth data structure.
Requirement	[ats_variablematrixwidth_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if VariableMatrixWidth instances point to the VariableMatrixWidth data type definition and follow the data model specified in the requirement tables and its dependencies.</p>

A.2.2. Coalescence

Abstract Test 33	/conf/variablematrixwidth/coalescence1
Test Purpose	Validate that the tile rows with a coalescence factors
Requirement	[req_variablematrixwidth_coalescence1]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if all VariableMatrixWidth elements in the instance has a coalescence factor bigger than 1.</p>

A.2.3. Conformance Class JSON TileMatrixSet

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/json-tilematrixset	
Target type	tile matrix sets
Requirements class	JSON TileMatrixSet"

A.2.4. Model

Abstract Test 34	/conf/json-tilematrixset/model
Test Purpose	Validate that a TileMatrixSet encoded in JSON implements the class TileMatrixSet
Requirement	[req_json_tilematrixset_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSet instances follow the data model specified in the requirement and associated tables and its dependencies.</p>

A.2.5. IETF

Abstract Test 35	/conf/json-tilematrixset/ietf
Test Purpose	Validate that a TileMatrixSet encoded in JSON conforms to IETF RFC7159
Requirement	[req_json_tilematrixset_ietf]
Test method	<p>Validate the requirements of the IETF</p> <p>Test passes if TileMatrixSet JSON instances pass format validation against the IETF rules.</p>

A.2.6. Schema

Abstract Test 36	/conf/json-tilematrixset/schema
Test Purpose	Validate that a TileMatrixSet encoded in JSON using the JSON schema for a tile matrix set.
Requirement	[req_json_tilematrixset_schema]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSet JSON instances pass validation against the tile matrix set JSON Schemas.</p>

A.2.7. Media type

Abstract Test 37	/conf/json-tilematrixset/media-type
Test Purpose	Validate that a TileMatrixSet encoded in an independent JSON document uses the media type application/json.
Requirement	[req_json_tilematrixset_media-type]
Test method	<p>Validate the requirements of the media type</p> <p>Test passes if the independent instances of TileMatrixSet are exposed as application/json media type.</p>

A.2.8. Conformance Class JSON VariableMatrixWidth

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/json-variablematrixwidth	
Target type	tile matrix sets
Requirements class	Requirements Class "JSON VariableMatrixWidth"

A.2.9. Model

Abstract Test 38	/conf/json-variablematrixwidth/model
Test Purpose	Validate if VariableMatrixWidth encoded in JSON implements the class VariableMatrixWidth
Requirement	[req_json_variablematrixwidth_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if VariableMatrixWidth instances follow the data model specified in the requirement and associated tables and its dependencies.</p>

A.2.10. IETF

Abstract Test 39	/conf/json-variablematrixwidth/ietf
Test Purpose	Validate if a VariableMatrixWidth encoded in JSON conforms to IETF RFC7159
Requirement	[req_json_variablematrixwidth_ietf]
Test method	<p>Validate the requirements of the IETF rules</p> <p>Test passes if VariableMatrixWidth JSON instances pass format validation against the IETF rules.</p>

A.2.11. Schema

Abstract Test 40	/conf/json-variablematrixwidth/schema
Test Purpose	Validate if a VariableMatrixWidth encoded in JSON validates using the JSON schema for a variable matrix width.
Requirement	[req_json_variablematrixwidth_schema]
Test method	<p>Validate the requirements of the schema</p> <p>Test passes if VariableMatrixWidth JSON instances pass validation against the variable matrix width JSON Schemas.</p>

A.2.12. Conformance Class XML TileMatrixSet

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/xml-tilematrixset	
Target type	tile matrix sets
Requirements class	Requirements Class "VariableMatrixWidth"

A.2.13. Model

Abstract Test 41	/conf/xml-tilematrixset/model
Test Purpose	Validate that aa TileMatrixSet encoded in XML implements the class TileMatrixSet
Requirement	[req_xml_tilematrixset_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSet instances use a TileMatrixSet XML data type definition that follows the data model specified in the requirement and associated tables and its dependencies.</p>

A.2.14. Schema

Abstract Test 42	/conf/xml-tilematrixset/schema
Test Purpose	Validates if a TileMatrixSet encoded in XML validates using the XML schema for a tile matrix set.
Requirement	[req_xml_tilematrixset_schema]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSet XML instances pass validation against the tile matrix set XML Schemas.</p>

A.2.15. Media Type

Abstract Test 43	/conf/xml-tilematrixset/media-type
Test Purpose	Validate that a TileMatrixSet encoded in an independent XML document uses the media type application/xml.
Requirement	[req_xml_tilematrixset_media-type]
Test method	<p>Validate the requirements of the media type</p> <p>Test passes if the independent instances of TileMatrixSet are exposed as application/xml MIME type.</p>

A.2.16. Conformance Class XML VariableMatrixWidth

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/xml-variablematrixwidth	
Target type	tile matrix sets

Requirements class	Requirements Class "XML VariableMatrixWidth"
--------------------	--

A.2.17. Model

Abstract Test 44	/conf/xml-variablematrixwidth/model
Test Purpose	Validate if a TileSetMetadata encoded in XML implements the class TileSetMetadata
Requirement	[req_xml_variablematrixwidth_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if VariableMatrixWidth instances use a VariableMatrixWidth XML data type definition that follows the data model specified in the requirement and associated tables its dependencies.</p>

A.2.18. Schema

Abstract Test 45	/conf/xml-variablematrixwidth/schema
Test Purpose	Validate if a TileSetMetadata encoded in XML validates using the XML schema for a tile matrix set link.
Requirement	[req_xml_variablematrixwidth_schema]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if VariableMatrixWidth XML instances pass validation against the variable matrix width XML Schemas.</p>

A.2.19. Conformance Class TileMatrixSetLimits

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/tilematrixsetlimits	
Target type	tile matrix sets
Requirements class	Requirements Class "TileMatrixSetLimit"

A.2.20. Model

Abstract Test 46	/conf/tilematrixsetlimits/model
Test Purpose	Validate if a tile matrix set limits follows the UML model and model description. Validate if each tileMatrix identifier is mentioned only once in the TileMatrixSetLimits

Requirement	[req_tilematrixsetlimits_model]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSetLimits instances point to the TileMatrixSetLimits data type definition and follow the data model specified in the requirements and associated tables and its dependencies.</p>
	Test passes if all tileMatrix identifier are different in this TileMatrixSetLimits

A.2.21. Conformance Class TileSetMetadata

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/tilesetmetadata	
Target type	tile set metadata
Requirements class	Requirements Class "TileSetMetadata"

A.2.22. Identifier

Abstract Test 47	/conf/tilematrixsetmetadata/identifier
Test Purpose	Validates if a tiled resource or dataset declares support to a tile matrix set by one of the following two methods: a link to a tile matrix set definition (including the TileMatrixSet definition from Common TileMatrixSet definitions (Informative) and Example encodings for Variable Matrix Width TileMatrixSets (Informative)) as one of the links in the links list or by explicitly including a TileMatrixSet definition (as an object in the <code>tileMatrixSet</code> property).
Requirement	[req_tilesetmetadata_identifier]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if a tiled resource or dataset declares support to a tile matrix set by one of the following two methods: a link to a tile matrix set definition (including the TileMatrixSet definition from Common TileMatrixSet definitions (Informative) and Example encodings for Variable Matrix Width TileMatrixSets (Informative)) as one of the links in the links list or by explicitly including a TileMatrixSet definition (as an object in the <code>tileMatrixSet</code> property).</p>

A.2.23. Model

Abstract Test 48	/conf/tilematrixsetmetadata/model
-------------------------	-----------------------------------

Test Purpose	Validate it when a tiled resource or dataset has tiles available, the metadata describing the tiles is described using the data model in UML model shown in Figure 13 and model description in Table 13 .
Requirement	[req_tilesetmetadata_model]
Test method	Test passes if TileSetMetadata instances point to the TileSetMetadata data type definition and follow the data model specified in the requirement and associated tables and its dependencies.

A.2.24. Conformance Class JSON TileMatrixSetLimits

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/json-tilematrixsetlimits	
Target type	tile set metadata
Requirements class	Requirements Class "JSON TileMatrixSetLimits"

A.2.25. Model

Abstract Test 49 /conf/json-tilematrixsetlimits/model	
Test Purpose	Validate that a TileMatrixSetLimits encoded in JSON implements the class TileMatrixSetLimits
Requirement	[req_json_tilematrixsetlimits_model]
Test method	Validate the requirements of the model Test passes if TileMatrixSetLimits instances follow the data model specified in the requirement and associated tables and its dependencies.

A.2.26. IETF

Abstract Test 50 /conf/json-tilematrixsetlimits/ietf	
Test Purpose	Validate that a TileMatrixSetLimits encoded in JSON conforms to IETF RFC7159
Requirement	[req_json_tilematrixsetlimits_ietf]
Test method	Validate the requirements of the IETF Test passes if TileMatrixSetLimits JSON instances pass format validation against the IETF rules.

A.2.27. Schema

Abstract Test 51	/conf/json-tilematrixsetlimits/schema
Test Purpose	Validate that a TileMatrixSetLimits encoded in JSON using the JSON schema for a tile matrix set.
Requirement	[req_json_tilematrixsetlimits_schema]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileMatrixSetLimits JSON instances pass validation against the tile matrix set limits JSON Schemas.</p>

A.2.28. Conformance Class JSON TileSetMetadata

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/json-tilesetmetadata	
Target type	tile set metadata
Requirements class	Requirements Class "JSON TileSetMetadata"

A.2.29. Model

Abstract Test 52	/conf/json-tilesetmetadata/model
Test Purpose	Validate is a TileSetMetadata encoded in JSON implements the class TileSetMetadata
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileSetMetadata instances follow the data model specified in the requirement and associated tables and its dependencies.</p>

A.2.30. IETF

Abstract Test 53	/conf/json-tilematrixsetlimits/ietf
Test Purpose	Validate if a TileSetMetadata encoded in JSON conforms to IETF RFC7159
Requirement	[req_json_tilesetmetadata_ietf]
Test method	<p>Validate the requirements of the IETF</p> <p>Test passes if TileSetMetadata JSON instances pass format validation against the IETF rules.</p>

A.2.31. Schema

Abstract Test 54	/conf/json-tilematrixsetlimits/schema
Test Purpose	Validate if a TileSetMetadata encoded in JSON validates using the JSON schema for a tile set metadata.
Requirement	[req_json_tilesetmetadata_media-type]
Test method:	<p>Validate the requirements of the schema</p> <p>Test passes if TileSetMetadata JSON instances pass validation against the tile set metadata JSON Schemas.</p>

A.2.32. Media type

Abstract Test 55	/conf/json-tilesetmetadata/media-type
Test Purpose	Validate that a TileSetMetadata encoded in an independent JSON document uses the media type application/json.
Requirement	[req_json_tilesetmetadata_media-type]
Test method	<p>Validate the requirements of the media type</p> <p>Test passes if the independent instances of TileSetMetadata are exposed as application/json media type.</p>

A.2.33. Conformance Class XML TileMatrixSetLimits

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/xml-tilematrixsetlimits	
Target type	tile set metadata
Requirements class	Requirements Class "XML TileMatrixSetLimits"

A.2.34. Model

Abstract Test 56	/conf/xml-tilematrixsetlimits/model
Test Purpose	A TileMatrixSetLimits encoded in XML <i>shall</i> implement the class TileMatrixSetLimits
Requirement	[req_xml_tilematrixsetlimits2d_schema]

Test method	Validate the requirements of the model Test passes if TileMatrixSetLimits instances point to the TileMatrixSetLimits data type definition and follow the data model specified in the requirement and in the associated tables and its dependencies.
--------------------	--

Abstract Test 57	/conf/xml-tilematrixsetlimits/schema
Test Purpose	Validate if a TileMatrixSetLimits encoded in XML validates using the XML schemas for a tile matrix set limits.
Requirement	[req__xml_tilematrixsetlimits2d_schema]
Test method	Validate the requirements of the schema Test passes if TileMatrixSetLimits XML instances pass validation against the tile matrix set limits XML Schemas.

A.2.35. Conformance Class XML TileSetMetadata

Conformance Class	
http://www.opengis.net/spec/tms/2.0/conf/xml-tilesetmetadata	
Target type	tile set metadata
Requirements class	Requirements Class "XML TileSetMetadata"

A.2.36. Model

Abstract Test 58	/conf/xml-tilematrixsetmetadata/model
Test Purpose:	Validate if a TileSetMetadata encoded in XML implements the class TileSetMetadata
Requirement	[req__xml_tilsetmetadata_model]
Test method:	Validate the requirements of the model Test passes if TileSetMetadata instances use a TileSetMetadata XML data type definition that follows the data model specified in the requirement and associated tables and its dependencies.

A.2.37. Schema

Abstract Test 59	/conf/xml-tilematrixsetmetadata/schema
Test Purpose	Validate if a TileSetMetadata encoded in XML validates using the XML schema for a tile set metadata.

Requirement	[req_xml_tilesetmetadata_schema]
Test method	<p>Validate the requirements of the model</p> <p>Test passes if TileSetMetadata XML instances pass validation against the tile set metadata XML Schemas.</p>

A.2.38. Media Type

Abstract Test 60	/conf/xml-tilematrixset/media-type
Test Purpose	Validate that a TileSetMetadata encoded in an independent XML document uses the media type application/xml.
Requirement	[req_xml_tilesetmetadata_media-type]
Test method	<p>Validate the requirements of the media type</p> <p>Test passes if the independent instances of TileSetMetadata are exposed as application/xml MIME type.</p>

Annex B: Schema Documents (Normative)

In addition to this document, this standard includes several normative Schema Documents. These Schema Documents are posted online at the URL <http://schemas.opengis.net/tms/2.0>.

The schemas for both JSON and XML encodings of TileMatrixSets are reproduced in the [Example encodings for Common TileMatrixSets \(Informative\)](#) section of this document, along with examples. Additional examples are found in the [Example encodings for Variable Matrix Width TileMatrixSets \(Informative\)](#) section.

The schemas for both JSON and XML encodings of TileSet Metadata are reproduced in the [Example encodings for TileSetMetadata \(Informative\)](#) sections of this document, along with examples.

B.1. JSON Schema

The JSON Schema Documents are situated in the json subfolder and are named:

- json/tileMatrixSet.json
- json/tileMatrixLimits.json
- json/tileSet.json

They include the schemas necessary to JSON-validate the classes JSONTileMatrixSet, JSONTileMatrixSetLimits and JSONTileSetMetadata, respectively.

In addition, the directory named [json/examples/tilematrixset](#) contains examples of the JSON TileMatrixSets encodings, while the directory named [json/examples/tileset](#) contains examples of the JSON tileset metadata encodings.

B.2. XML Schema

The XML Schema Documents are situated in the xml subfolder and are named:

- xml/tilematrixset.xsd
- xml/tilematrixlimits.xsd
- xml/tileset.xsd

They includes the schemas necessary to XML-validate the classes XMLTileMatrixSet, XMLTileMatrixSetLimits and XMLTileSetMetadata, respectively.

In addition, the directory named [xml/examples/tilematrixset](#) contains examples of the XML TileMatrixSets encodings, while the directory named [xml/examples/tileset](#) contains examples of XML tileset metadata encodings.

Annex C: Well-known scale sets (Informative)

The following well-known scale sets (WKSS) are defined in this standard. To be conformant to a WKSS, the tile matrices of a tile matrix set should only include a consecutive subset of the scales defined in one of the following tables (or their implied extensions). Note that the correspondence between the numeric identifiers of a TileMatrixSet and those of a WKSS might be offset by a fixed number of scales. Cell sizes (in terrain units) are calculated assuming 0.28 mm pixel size and the WGS84 equatorial Earth diameter.

The WKSS concept was introduced in WMTS to improve interoperability, but experience has demonstrated that the use of Common TileMatrixSet ([Common TileMatrixSet definitions \(Informative\)](#)) is even better. The use of WKSS is no longer encouraged by this standard.

C.1. GlobalCRS84Scale

URI: <http://www.opengis.net/def/wkss/OGC/1.0/GlobalCRS84Scale>

This WKSS has been defined for global cartographic products. Rounded scales have been chosen for intuitive cartographic representation of vector data. The scale denominator is only accurate near the Equator.

CRS	Scale Denominator	Cell Size (degrees)
http://www.opengis.net/def/crs/OGC/1.3/CRS84	500,000,000	1.25764139776733
	250,000,000	0.628820698883665
	100,000,000	0.251528279553466
	50,000,000	0.125764139776733
	25,000,000	$6.28820698883665 \times 10^{-2}$
	10,000,000	$2.51528279553466 \times 10^{-2}$
	5,000,000	$1.25764139776733 \times 10^{-2}$
	2,500,000	$6.28820698883665 \times 10^{-3}$
	1,000,000	$2.51528279553466 \times 10^{-3}$
	500,000	$1.25764139776733 \times 10^{-3}$
	250,000	$6.28820698883665 \times 10^{-4}$
	100,000	$2.51528279553466 \times 10^{-4}$
	50,000	$1.25764139776733 \times 10^{-4}$
	25,000	$6.28820698883665 \times 10^{-5}$
	10,000	$2.51528279553466 \times 10^{-5}$
	5000	$1.25764139776733 \times 10^{-5}$
	2500	$6.28820698883665 \times 10^{-6}$
	1000	$2.51528279553466 \times 10^{-6}$
	500	$1.25764139776733 \times 10^{-6}$
	250	$6.28820698883665 \times 10^{-7}$
	100	$2.51528279553466 \times 10^{-7}$

Table 22. Definition of Well-known scale set GlobalCRS84Scale

C.2. GlobalCRS84Pixel

URI: <http://www.opengis.net/def/wkss/OGC/1.0/GlobalCRS84Pixel>

This WKSS has been defined for global cartographic products. Rounded cell sizes have been chosen for intuitive cartographic representation of raster data. Some values have been chosen to coincide with original cell size of commonly used global products like SRTM (1" and 3"), GTOPO (30") or ETOPO (2' and 5'). The scale denominator and approximated cell size in meters are only accurate near the Equator.

CRS	Scale Denominator	Cell Size (degrees)	Approx. Cell Size (m)
http://www.opengis.net/def/crs/OGC/1.3/CRS84	795,139,219.9519541	2	240,000
	397,569,609.9759771	1	120,000
	198,784,804.9879885	0.5 (30')	60,000
	132,523,203.3253257	0.333333333333333 (20')	40,000
	66,261,601.66266284	0.166666666666667 (10')	20,000
	33,130,800.83133142	$8.3333333333333 \times 10^{-2}$ (5')	10,000
	13,252,320.33253257	$3.3333333333333 \times 10^{-2}$ (2')	4000
	6,626,160.166266284	$1.66666666666667 \times 10^{-2}$ (1')	2000
	3,313,080.083133142	$8.3333333333333 \times 10^{-3}$ (30'")	1000
	1,656,540.041566571	$4.16666666666667 \times 10^{-3}$ (15'")	500
	552,180.0138555236	$1.38888888888889 \times 10^{-3}$ (5'")	166
	331,308.0083133142	$8.3333333333333 \times 10^{-4}$ (3'")	100
	110,436.0027711047	$2.777777777777778 \times 10^{-4}$ (1'")	33
	55,218.00138555237	$1.38888888888889 \times 10^{-4}$ (0.5'")	16
	33,130.80083133142	$8.3333333333333 \times 10^{-5}$ (0.3'")	10
	11,043.60027711047	$2.777777777777778 \times 10^{-5}$ (0.1'")	3
	3313.080083133142	$8.3333333333333 \times 10^{-6}$ (0.03'")	1
	1104.360027711047	$2.777777777777778 \times 10^{-6}$ (0.01'")	0.33

Table 23. Definition of Well-known scale set GlobalCRS84Pixel

C.3. GoogleCRS84Quad

URI: <http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad>

This WKSS has been defined to allow quadtree pyramids in CRS84. The scale denominator is only accurate near the equator.

CRS	Scale Denominator	Cell Size (m)
http://www.opengis.net/def/crs/OGC/1.3/CRS84	559,082,264.0287178	1.406250000000000
	279,541,132.0143589	0.703125000000000
	139,770,566.0071794	0.351562500000000
	69,885,283.00358972	0.175781250000000
	34,942,641.50179486	8.78906250000000 × 10 ⁻²
	17,471,320.75089743	4.39453125000000 × 10 ⁻²
	8,735,660.375448715	2.19726562500000 × 10 ⁻²
	4,367,830.187724357	1.09863281250000 × 10 ⁻²
	2,183,915.093862179	5.49316406250000 × 10 ⁻³
	1,091,957.546931089	2.74658203125000 × 10 ⁻³
	545,978.7734655447	1.37329101562500 × 10 ⁻³
	272,989.3867327723	6.86645507812500 × 10 ⁻⁴
	136,494.6933663862	3.43322753906250 × 10 ⁻⁴
	68,247.34668319309	1.71661376953125 × 10 ⁻⁴
	34,123.67334159654	8.58306884765625 × 10 ⁻⁵
	17,061.83667079827	4.29153442382812 × 10 ⁻⁵
	8530.918335399136	2.14576721191406 × 10 ⁻⁵
	4265.459167699568	1.07288360595703 × 10 ⁻⁵
	2132.729583849784	5.36441802978516 × 10 ⁻⁶

Table 24. Definition of Well-known scale set GoogleCRS84Quad

NOTE

The first scale denominator allows representation of the whole world in a single tile of 256x256 cells, where 128 lines of the tile are left blank. The latter is the reason why in the [Common TileMatrixSet definitions \(Informative\)](#) "World CRS84 Quad TileMatrixSet definition" [<http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldCRS84Quad>] this level is not used. The next level allows representation of the whole world in 2x1 tiles of 256x256 cells and so on in powers of 2.

NOTE

Selecting the word "Google" for this WKSS id is maintained for backwards compatibility even if the authors recognize that it was an unfortunate selection and might result in confusion since the "Google-like" tiles do not use CRS84.

C.4. GoogleMapsCompatible

URI: <http://www.opengis.net/def/wkss/OGC/1.0/GoogleMapsCompatible>

This well-known scale set has been defined to be compatible with many mass marked implementations such as Google Maps, Microsoft Bing Maps (formerly Microsoft Live Maps) and Open Street Map tiles. The scale denominator and cell size are only accurate near the equator.

CRS	Zoom level name	Scale Denominator	Cell Size (m)
http://www.opengis.net/def/crs/EPSG/0/3857 WGS 84 / Pseudo-Mercator	0	559,082,264.0287178	156,543.0339280410
	1	279,541,132.0143589	78,271.51696402048
	2	139,770,566.0071794	39,135.75848201023
	3	69,885,283.00358972	19,567.87924100512
	4	34,942,641.50179486	9783.939620502561
	5	17,471,320.75089743	4891.969810251280
	6	8,735,660.375448715	2445.984905125640
	7	4,367,830.187724357	1222.992452562820
	8	2,183,915.093862179	611.4962262814100
	9	1,091,957.546931089	305.7481131407048
	10	545,978.7734655447	152.8740565703525
	11	272,989.3867327723	76.43702828517624
	12	136,494.6933663862	38.21851414258813
	13	68,247.34668319309	19.10925707129406
	14	34,123.67334159654	9.554628535647032
	15	17,061.83667079827	4.777314267823516
	16	8530.918335399136	2.388657133911758
	17	4265.459167699568	1.194328566955879
	18	2132.729583849784	0.5971642834779395
	19	1066.364791924892	0.2985821417389697
	20	533.1823959624460	0.1492910708694849
	21	266.5911979812230	0.07464553543474244
	22	133.2955989906115	0.03732276771737122
	23	66.64779949530575	0.01866138385868561
	24	33.32389974765287	0.009330691929342805

Table 25. Definition of Well-known scale set GoogleMapsCompatible

NOTE

Level 0 allows representing most of the world (limited to latitudes between approximately ± 85 degrees) in a single tile of 256x256 cells (Mercator projection cannot cover the whole world because mathematically the poles are at infinity). The next level represents most of the world in 2x2 tiles of 256x256 cells and so on in powers of 2.

C.5. WorldMercatorWGS84

URI: <http://www.opengis.net/def/wkss/OGC/1.0/WorldMercatorWGS84>

This well-known scale set has been defined as similar to Google Maps and Microsoft Bing Maps but using the WGS84 ellipsoid. The scale denominator and cell size are only accurate near the equator.

CRS	Zoom level name	Scale Denominator	Cell Size (m)
http://www.opengis.net/def/crs/EPSG/0/3395	0	559,082,264.02871774	156,543.033928040
WGS 84 / World Mercator	1	279,541,132.01435887	78,271.5169640205
	2	139,770,566.00717943	39,135.7584820102
	3	69,885,283.003589718	19,567.8792410051
	4	34,942,641.501794859	9783.93962050256
	5	17,471,320.750897429	4891.96988102512
	6	8,735,660.3754487147	2445.98490512564
	7	4,367,830.1877243573	1222.99245256282
	8	2,183,915.0938621786	611.496226281410
	9	1,091,957.5469310893	305.748113140705
	10	545,978.77346554467	152.874056570352
	11	272,989.38673277233	76.4370282851762
	12	136,494.69336638616	38.2185141425881
	13	68,247.346683193084	19.1092570712940
	14	34,123.673341596542	9.55462853564703
	15	17,061.836670798271	4.77731426782351
	16	8530.9183353991355	2.38865713391175
	17	4265.4591676995677	1.19432856695587
	18	2132.7295838497838	0.59716428347793
	19	1066.3647919248919	0.29858214173896
	20	533.18239596244597	0.14929107086948
	21	266.59119798122298	0.07464553543474
	22	133.29559899061149	0.03732276771737
	23	66.647799495305746	0.01866138385868
	24	33.323899747652873	0.00933069192934

Table 26. Definition of Well-known scale set WorldMercatorWGS84

NOTE

Level 0 allows representing most of the world (limited to latitudes between approximately ± 85 degrees) in a single tile of 256x256 cells (Mercator projection cannot cover the whole world because mathematically the poles are at infinity). The next level represents most of the world in 2x2 tiles of 256x256 cells and so on in powers of 2.

NOTE

Mercator projection distorts the cell size closer to the poles. The cell sizes provided here are only valid next to the equator.

NOTE

The scales and cell sizes of *WorldMercatorWGS84* and *GoogleMapsCompatible* are identical, but the two WKSS reference a different CRS. This *WorldMercatorWGS84* WKSS was introduced in the first version of this standard and not present in the WMTS 1.0.0 specifications Annex E. However, WKSS are obsolete and not required to define a TileMatrixSet, so the introduction of this new WKSS was not necessary to define the *WorldMercatorWGS84Quad* TileMatrixSet.

Annex D: Common TileMatrixSet definitions (Informative)

This Annex includes some definitions for TileMatrixSets that are commonly used.

D.1. WebMercatorQuad

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/WebMercatorQuad>

CRS: <http://www.opengis.net/def/crs/EPSG/0/3857>, WGS 84 / Pseudo-Mercator

BBOX LowerLeft: -20,037,508.3427892, -20,037,508.3427892 (lat/long: -85.0511287798, -180)

BBOX UpperRight: 20,037,508.3427892, 20,037,508.3427892 (lat/long: 85.0511287798, 180)

WellKnownScaleSet: <http://www.opengis.net/def/wkss/OGC/1.0/GoogleMapsCompatible>

PointOfOrigin: -20,037,508.3427892, 20,037,508.3427892

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (m)	Matrix Width	Matrix Height
0	559,082,264.0287178	156,543.0339280410	1	1
1	279,541,132.0143589	78,271.51696402048	2	2
2	139,770,566.0071794	39,135.75848201023	4	4
3	69,885,283.00358972	19,567.87924100512	8	8
4	34,942,641.50179486	9783.939620502561	16	16
5	17,471,320.75089743	4891.969810251280	32	32
6	8,735,660.375448715	2445.984905125640	64	64
7	4,367,830.187724357	1222.992452562820	128	128
8	2,183,915.093862179	611.4962262814100	256	256
9	1,091,957.546931089	305.7481131407048	512	512
10	545,978.7734655447	152.8740565703525	1024	1024
11	272,989.3867327723	76.43702828517624	2048	2048
12	136,494.6933663862	38.21851414258813	4096	4096
13	68,247.34668319309	19.10925707129406	8192	8192

14	34,123.67334159654	9.554628535647032	16,384	16,384
15	17,061.83667079827	4.777314267823516	32,768	32,768
16	8530.918335399136	2.388657133911758	65,536	65,536
17	4265.459167699568	1.194328566955879	131,072	131,072
18	2132.729583849784	0.5971642834779395	262,144	262,144
19	1066.36479192489	0.2985821417389700	524,288	524,288
20	533.182395962445	0.1492910708694850	1,048,576	1,048,576
21	266.591197981222	0.0746455354347424	2,097,152	2,097,152
22	133.295598990611	0.0373227677173712	4,194,304	4,194,304
23	66.6477994953056	0.0186613838586856	8,388,608	8,388,608
24	33.3238997476528	0.0093306919293428	16,777,216	16,777,216

Table 27. Definition of the WebMercatorQuad TileMatrixSet

One can define an arbitrary number of zoom levels and do not need to include all the zoom levels defined here. Here, 25 zoom levels are illustrated.

- NOTE** Mercator projection distorts the cell size the closer to the poles. The cell sizes provided here are only valid next to the equator in the direction E-W.
- NOTE** The CRS code 3857 is the official code for Web Mercator. An unofficial code “900913” (GOOGLE spelled with numbers) was initially assigned and is sometimes still used.

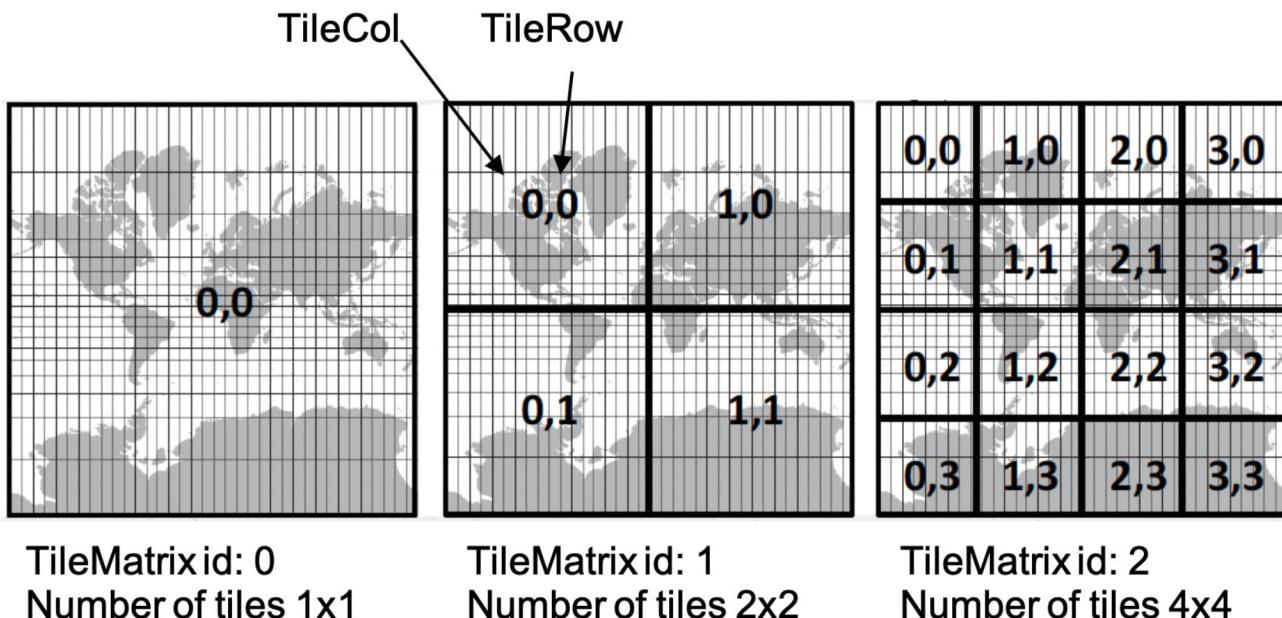


Figure 14. The 3 first Tile Matrix of the WebMercatorQuad TileMatrixSet (Source CCA)

This tile matrix set is the most used tile matrix set in the mass market: for example, by Google

Maps, Microsoft Bing Maps and Open Street Map tiles. Nevertheless, it has been long criticized because it is based on a spherical Mercator instead of an ellipsoid. The use of WebMercatorQuad should be limited to visualization. Any additional use (including distance measurements, routing etc.) needs to use the Mercator spherical expressions to transform the coordinate to an appropriate CRS first.

NOTE

For example, the EPSG database version 8.9 says about the 3857: “Uses spherical development of ellipsoidal coordinates. Relative to WGS 84 / World Mercator (CRS code 3395) errors of 0.7 percent in scale and differences in northing of up to 43km in the map (equivalent to 21km on the ground) may arise.”

NOTE

The use of Web Mercator can generate erroneous geospatial positioning information poses an unacceptable risk to global safety of navigation activities, and department of defense, intelligence community, and allied partner systems, missions, and operations that require accurate and precise positioning and navigation information.” The use of WorldMercatorWGS84Quad is recommended.

D.2. WorldCRS84Quad

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldCRS84Quad>

D.2.1. Variant 1: World CRS84 Quad (recommended)

This Tile Matrix Set defines tiles in the Equirectangular Plate Carrée projection in the CRS84 CRS for the whole world.

CRS: <http://www.opengis.net/def/crs/OGC/1.3/CRS84>, CRS84

BBOX LowerLeft: -180, -90

BBOX UpperRight: 180, 90

WellKnownScaleSet: <http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad>

PointOfOrigin: -180, 90

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (degrees) (true at the equator)	Matrix Width	Matrix Height

0	279,541,132.0143589	0.703125000000000	2	1
1	139,770,566.0071794	0.351562500000000	4	2
2	69,885,283.00358972	0.175781250000000	8	4
3	34,942,641.50179486	$8.7890625000000 \times 10^{-2}$	16	8
4	17,471,320.75089743	$4.3945312500000 \times 10^{-2}$	32	16
5	8,735,660.375448715	$2.1972656250000 \times 10^{-2}$	64	32
6	4,367,830.187724357	$1.0986328125000 \times 10^{-2}$	128	64
7	2,183,915.093862179	$5.4931640625000 \times 10^{-3}$	256	128
8	1,091,957.546931089	$2.7465820312500 \times 10^{-3}$	512	256
9	545,978.7734655447	$1.37329101562500 \times 10^{-3}$	1024	512
10	272,989.3867327723	$6.86645507812500 \times 10^{-4}$	2048	1024
11	136,494.6933663862	$3.43322753906250 \times 10^{-4}$	4096	2048
12	68,247.34668319309	$1.71661376953125 \times 10^{-4}$	8192	4096
13	34,123.67334159654	$8.58306884765625 \times 10^{-5}$	16,384	8192
14	17,061.83667079827	$4.29153442382812 \times 10^{-5}$	32,768	16,384
15	8530.918335399136	$2.14576721191406 \times 10^{-5}$	65,536	32,768
16	4265.459167699568	$1.07288360595703 \times 10^{-5}$	131,072	65,536
17	2132.729583849784	$5.36441802978516 \times 10^{-6}$	262,144	131,072
18	1066.36479192489	$2.68220901489258 \times 10^{-6}$	524,288	262,144
19	533.182395962445	$1.34110450744629 \times 10^{-6}$	1,048,576	524,288
20	266.591197981222	$6.70552253723144 \times 10^{-7}$	2,097,152	1,048,576
21	133.295598990611	$3.35276126861572 \times 10^{-7}$	4,194,304	2,097,152
22	66.6477994953056	$1.67638063430786 \times 10^{-7}$	8,388,608	4,194,304
23	33.3238997476528	$8.3819031715393 \times 10^{-8}$	16,777,216	8,388,608

Table 28. Definition of the WorldCRS84Quad TileMatrixSet

One can define an arbitrary number of zoom levels and do not need to include all the ones defined here. Here, 18 zoom levels are illustrated.

Tile Matrix Id 1 (2x1 tiles) of the WorldCRS84Quad TileMatrixSet

(Source INSPIRE technical guidance) image::figures/11.png[Tile Matrix Id 1 (2x1 tiles) of the WorldCRS84Quad TileMatrixSet
 (Source INSPIRE technical guidance)]

NOTE

The zoom level identifiers in this TileMatrixSet do not correspond to the same scale values in Annex E.3 of WMTS 1.0. In this TileMatrixSet, the TileMatrix with identifier “-1” has only one tile with 128 lines left blank. For that reason, many implementers do not want to offer this level (including the INSPIRE technical guidance) and prefer to start with a TileMatrix that represents the world with just 2 tiles (one for the negative longitudes and one for the positive longitudes).

NOTE

The scale denominators for this TileMatrixSet and WorldMercatorWGS84Quad and WebMercatorQuad are the same but the identifiers are displaced by one. This might generate confusion.

NOTE

for INSPIRE: The Technical Guidance for the implementation of INSPIRE View Services defines a TileMatrixSet called InspireCRS84Quad that is identical to this one. Note that the current version of the INSPIRE Technical Guidance cited in the Bibliography Annex could accidentally generate confusion because it is comparing a GoogleCRS84Quad (that is a well-known scale set name) with the InspireCRS84Quad (that is a TileMatrixSet definition that does not link to any WKSS).

D.2.2. Variant 2: World EPSG:4326 Quad

Despite what is stated in [TileMatrixSet CRS Compatibility](#), some implementers prefer to define the previous TileMatrixSet using the CRS <http://www.opengis.net/def/crs/EPSC/0/4326>. The definition is the same as the variant defined using <http://www.opengis.net/def/crs/OGC/1.3/CRS84> except that CRS coordinates are expressed in latitude, longitude order, affecting the *PointOfOrigin* and the *BoundingBox* encoding only. For most practical purposes, both variations are equivalent because a TileMatrixSet primarily defines the tiling structure as well as the scale/resolution at each tile matrix, rather than how the data within each tile is stored. For many raster and vector tiles formats, CRS84 and EPSG:4326 are equivalent as a particular axis order is enforced. Additional parameters to an API for example could also override the default axis order by specifying the CRS as either CRS84 or EPSG:4326.

If possible, defining it in terms of <http://www.opengis.net/def/crs/OGC/1.3/CRS84> is recommended instead of this variation, because it uses the CRS consistent with the TileMatrixSet URI. However, we introduce it here to clarify how an implementation based on EPSG:4326 should look like and avoid confusion.

This Tile Matrix Set defines tiles in the Equiangular Plate Carrée projection in the EPSG:4326 CRS for the whole world.

CRS: <http://www.opengis.net/def/crs/EPSG/0/4326>, EPSG:4326

BBOX LowerLeft: -90, -180

BBOX UpperRight: 90, 180

WellKnownScaleSet: <http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad>

PointOfOrigin: 90, -180

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (degrees) (true at the equator)	Matrix Width	Matrix Height
0	279,541,132.0143589	0.703125000000000	2	1
1	139,770,566.0071794	0.351562500000000	4	2
2	69,885,283.00358972	0.175781250000000	8	4
3	34,942,641.50179486	$8.78906250000000 \times 10^{-2}$	16	8
4	17,471,320.75089743	$4.39453125000000 \times 10^{-2}$	32	16
5	8,735,660.375448715	$2.19726562500000 \times 10^{-2}$	64	32
6	4,367,830.187724357	$1.09863281250000 \times 10^{-2}$	128	64
7	2,183,915.093862179	$5.49316406250000 \times 10^{-3}$	256	128
8	1,091,957.546931089	$2.74658203125000 \times 10^{-3}$	512	256
9	545,978.7734655447	$1.37329101562500 \times 10^{-3}$	1024	512
10	272,989.3867327723	$6.86645507812500 \times 10^{-4}$	2048	1024
11	136,494.6933663862	$3.43322753906250 \times 10^{-4}$	4096	2048
12	68,247.34668319309	$1.71661376953125 \times 10^{-4}$	8192	4096
13	34,123.67334159654	$8.58306884765625 \times 10^{-5}$	16,384	8192
14	17,061.83667079827	$4.29153442382812 \times 10^{-5}$	32,768	16,384
15	8530.918335399136	$2.14576721191406 \times 10^{-5}$	65,536	32,768
16	4265.459167699568	$1.07288360595703 \times 10^{-5}$	131,072	65,536
17	2132.729583849784	$5.36441802978516 \times 10^{-6}$	262,144	131,072
18	1066.36479192489	$2.68220901489258 \times 10^{-6}$	524,288	262,144
19	533.182395962445	$1.34110450744629 \times 10^{-6}$	1,048,576	524,288
20	266.591197981222	$6.70552253723144 \times 10^{-7}$	2,097,152	1,048,576
21	133.295598990611	$3.35276126861572 \times 10^{-7}$	4,194,304	2,097,152

22	66.6477994953056	$1.67638063430786 \times 10^{-7}$	8,388,608	4,194,304
23	33.3238997476528	$8.3819031715393 \times 10^{-8}$	16,777,216	8,388,608

Table 29. Definition of the WorldCRS84Quad TileMatrixSet (EPSG:4326)

D.3. WorldMercatorWGS84Quad

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldMercatorWGS84Quad>

CRS: <http://www.opengis.net/def/crs/EPSG/0/3395>, WGS 84 / World-Mercator

BBOX LowerLeft: -20,037,508.3427892, -20,037,508.3427892 (lat, long: -85.08405903, -180)

BBOX UpperRight: 20,037,508.3427892, 20,037,508.3427892 (lat, long: 85.08405903, 180)

WellKnownScaleSet: <http://www.opengis.net/def/wkss/OGC/1.0/WorldMercatorWGS84>

PointOfOrigin: -20,037,508.3427892, 20,037,508.3427892

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (m) (true at the equator)	Cell Size (m) at latitude ±31.0606963703645 (informative)	Matrix Width	Matrix Height
0	559,082,264.02871774	156,543.033928040	134,217.728	1	1
1	279,541,132.01435887	78,271.5169640205	67,108.864	2	2
2	139,770,566.00717943	39,135.7584820102	33,554.432	4	4
3	69,885,283.003589718	19,567.8792410051	16,777.216	8	8
4	34,942,641.501794859	9783.93962050256	8388.608	16	16
5	17,471,320.750897429	4891.96988102512	4194.304	32	32
6	8,735,660.3754487147	2445.98490512564	2097.152	64	64
7	4,367,830.1877243573	1222.99245256282	1048.576	128	128
8	2,183,915.0938621786	611.496226281410	524.288	256	256
9	1,091,957.5469310893	305.748113140705	262.144	512	512
10	545,978.77346554467	152.874056570352	131.072	1024	1024
11	272,989.38673277233	76.4370282851762	65.536	2048	2048
12	136,494.69336638616	38.2185141425881	32.768	4096	4096

13	68,247.346683193084	19.1092570712940	16.384	8192	8192
14	34,123.673341596542	9.55462853564703	8.192	16,384	16,384
15	17,061.836670798271	4.77731426782351	4.096	32,768	32,768
16	8530.9183353991355	2.38865713391175	2.048	65,536	65,536
17	4265.4591676995677	1.19432856695587	1.024	131.072	131.072
18	2132.7295838497838	0.59716428347793	0.512	262,144	262,144
19	1066.3647919248919	0.29858214173896	0.256	524,288	524,288
20	533.18239596244597	0.14929107086948	0.128	1,048,576	1,048,576
21	266.59119798122298	0.07464553543474	0.064	2,097,152	2,097,152
22	133.29559899061149	0.03732276771737	0.032	4,194,304	4,194,304
23	66.647799495305746	0.01866138385868	0.016	8,388,608	8,388,608
24	33.323899747652873	0.00933069192934	0.008	16,777,216	16,777,216

Table 30. Definition of the WorldMercatorWGS84Quad TileMatrixSet

One can define an arbitrary number of zoom levels and do not need to include all the zoom levels defined here. Here, 25 zoom levels are illustrated.

This Tile Matrix Set looks similar to the previous one (Web Mercator Quad) but this one is based on an ellipsoidal Mercator. Please note that the most northern latitude cover by this one is 85.08405903 (different from Web Mercator).

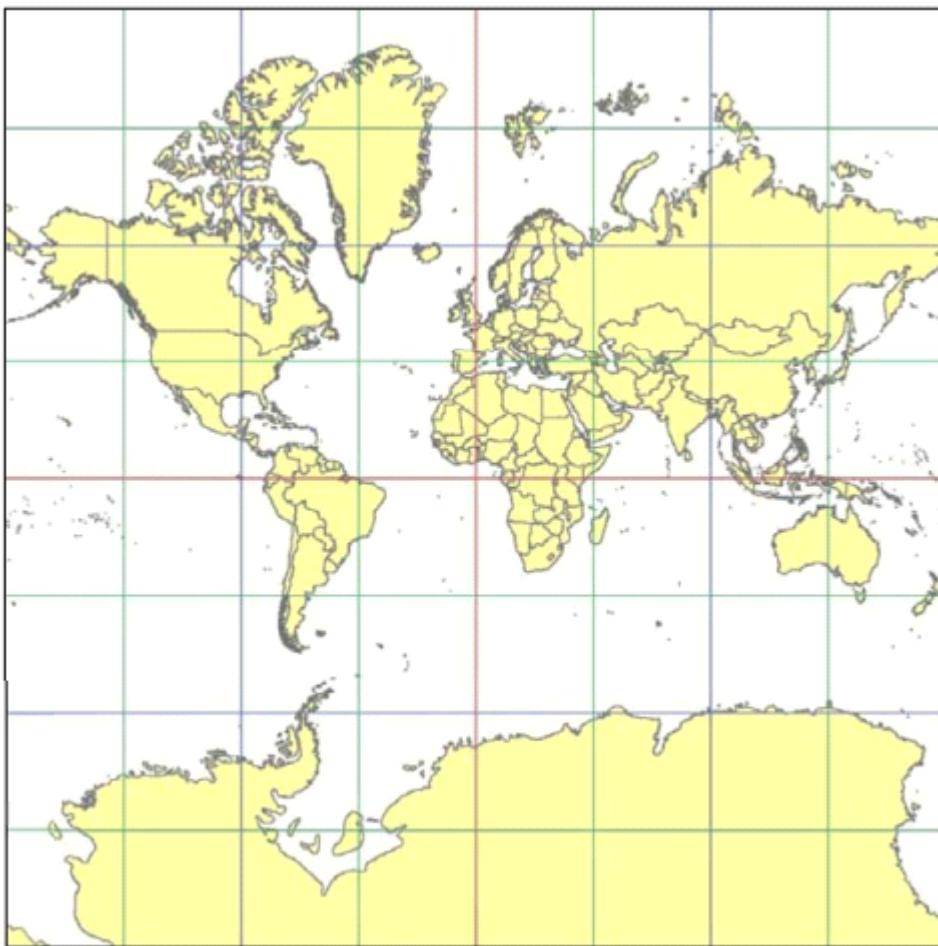


Figure 15. Tile Matrix Id 1 (red lines; 2x2 tiles) and 3 (blue lines; 8x8 tiles) of the WorldMercatorWGS84Quad TileMatrixSet (Source NGA)

NOTE

The NGA Geomatics Office reminds the community to use DoD approved World Geodetic System 1984 (WGS 84) applications for all mission critical activities and encourages the use of WGS84 based tile matrix sets like this one and discourages the use of Web Mercator tiles based on Web Mercator such as WebMercatorQuad.

NOTE

The NGA Geomatics Office recommends the use of Universal zoom-levels scale set that are defined as true cell size at ± 31.0606963703645 degrees of latitude that implies a scale reduction at the equator of 0.857385503731176 . This standard recommends the use of scale denominators at the equator for convenience.

D.4. Universal Transverse Mercator WGS84 Quad family (UTM##WGS84Quad)

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/UTM##WGS84Quad>

The Universal Transversal Mercator (a special case of transverse Mercator), divides the world into 60 zones by longitude. No single zone would make a global or near-global map. Therefore, this

definition is a family of 60 TileMatrixSets in a single table.

##: it is a number that goes from 01 to 60

CRSs: [<http://www.opengis.net/def/crs/EPSG/0/32601>, <http://www.opengis.net/def/crs/EPSG/0/32660>] WGS 84 / UTM

BBOX LowerLeft: -9,501,965.72931276, -20,003,931.4586255; lat, long: -180, -62+(##-31)*6

BBOX UpperRight: 10,501,965.7293128, 20,003,931.4586255 (lat, long: 180, 68+(##-31)*6)

WellKnownScaleSet: N/A

PointOfOrigin: -9,501,965.72931276, 20,003,931.4586255

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (m) along the central meridian	Cell Size (m) at a point on the Equator and at longitude ±30.7005243 32812+3+(##-31)*6	Matrix Width	Matrix Height
1	279,072,704.500914	78,140.3572602559	67,108.864	1	2
2	139,536,352.250457	39,070.178630128	33,554.432	2	4
3	69,768,176.1252285	19,535.089315064	16,777.216	4	8
4	34,884,088.0626143	9767.5446575319	8388.608	8	16
5	17,442,044.0313071	4883.772328766	4194.304	16	32
6	8,721,022.01565356	2441.886164383	2097.152	32	64
7	4,360,511.00782678	1220.9430821915	1048.576	64	128
8	2,180,255.50391339	610.471541095749	524.288	128	256
9	1,090,127.75195670	305.235770547875	262.144	256	512
10	545,063.875978348	152.617885273937	131.072	512	1024
11	272,531.937989174	76.3089426369687	65.536	1024	2048
12	136,265.968994587	38.1544713184843	32.768	2048	4096
13	68,132.9844972935	19.0772356592422	16.384	4096	8192
14	34,066.4922486467	9.53861782962109	8.192	8192	16,384

15	17,033.2461243234	4.76930891481054	4.096	16,384	32,768
16	8516.62306216168	2.38465445740527	2.048	32,768	65,536
17	4258.31153108084	1.19232722870264	1.024	65,536	131.072
18	2129.15576554042	0.596163614351318	0.512	131.072	262,144
19	1064.57788277021	0.298081807175659	0.256	262,144	524,288
20	532.288941385105	0.149040903587829	0.128	524,288	1,048,576
21	266.144470692553	0.0745204517939147	0.064	1,048,576	2,097,152
22	133.072235346276	0.0372602258969574	0.032	2,097,152	4,194,304
23	66.5361176731382	0.0186301129484787	0.016	4,194,304	8,388,608
24	33.2680588365691	0.0093150564742393	0.008	8,388,608	16,777,216
		4			

Table 31. Definition of the UTM##WGS84Quad TileMatrixSets

One can define an arbitrary number of zoom levels and do not need to include all the zoom levels defined here. Here, 24 zoom levels are illustrated.

NOTE

The southern hemisphere ([<http://www.opengis.net/def/crs/EPSG/0/32701>, <http://www.opengis.net/def/crs/EPSG/0/32760>]) is covered by extending the UTM northern CRSs to the south in a way that the southern hemisphere CRSs are neither used nor needed.

NOTE

The UTM projection is supposed to be used in zones that are only 3 degrees apart from the central meridian forming 6-degree wide zones. In some parts of the world, it is useful to relax this limitation to cover a wider object (for example, Spain can be fully represented in UTM30 zone if the 3-degree limit is relaxed). The farther one goes from the central meridian, the more deformations are experienced in the projection. The top left corner of this tile matrix set has been defined 65 degrees apart of the central meridian to allow much more extreme cases, but it is highly recommended that applications limit themselves to the 6-degree wide central area and use TileMatrixSetLimits to define actual boundaries of the tile indices used in this area.

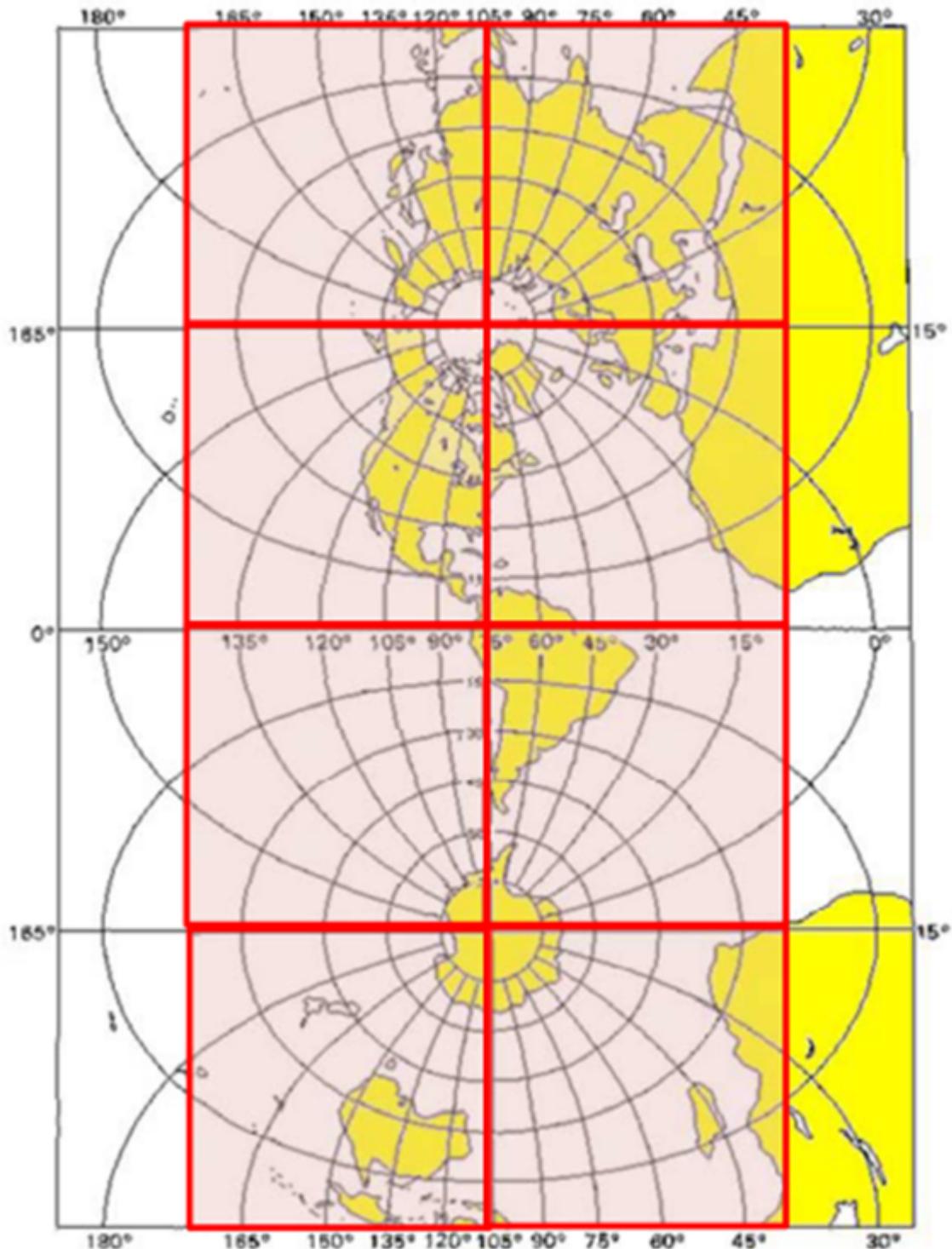


Figure 16. Tile Matrix Id 1 (dashed blue lines; 1x2 tiles) and 2 (red lines; 2x4 tiles) of the UTM18WGS84Quad TileMatrixSet (Source NGA)

NOTE

The NGA Geomatics Office recommends the use of Universal zoom-levels scale set that are defined as true cell size at ± 30.700524332812 degrees of longitude at both sides of the central meridian that implies a scale reduction at the central meridian of 0.85882463752355 . The tiles considering this scale are exactly the same than considering true cell size at the equator and no scale reduction. This standard recommends the use of scale denominators at the equator for convenience.

D.5. Arctic Universal Polar Stereographic WGS 84 Quad (UPSArcticWGS84Quad)

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/UPSArcticWGS84Quad>

It is difficult to find consensus in the geospatial community for what constitutes the “best” tile matrix set for the polar areas. Even if everyone agrees on using a polar stereographic, the election of the PointOfOrigin and scale denominators is almost arbitrary. This document presents the NGA recommendation for polar stereographic that allows for the representation of more than one hemisphere to 15 degrees into the opposite hemisphere and shares a common set of cell sizes with the WorldMercatorWGS84Quad and the UTM##WGS84Quad. The selection of a CRS for a polar stereographic is difficult and this document follows several criteria (see NOTE 2). In the end, the UPS North that is commonly used in conjunction with UTM was selected.

NOTE

In Mercator or Transversal Mercator projections the first scale denominator and top left corner are selected in a way that a single tile can cover all ranges of longitudes or latitudes respectively. Due to the nature of this projection, these criteria cannot be applied. The top left corner selection deeply depends on the application. The very distant top left corner was selected here to include as many applications as possible.

CRS: <http://www.opengis.net/def/crs/EPSG/0/5041>, WGS 84 Universal Polar Stereographic North

BBOX LowerLeft: -14,440,759.350252, -14,440,759.350252

BBOX UpperRight: 18,440,759.350252, 18,440,759.350252

WellKnownScaleSet: N/A

PointOfOrigin: -14,440,759.350252, 18,440,759.350252

TileWidth: 256

TileHeight: 256

Tile Matrix id	Scale Denominator	Cell Size (m) (true at latitude ~81)	True Cell Size (m) at the pole (informative)	Matrix Width	Matrix Height
0	458,726,544.4	128,443.4324	129,218.7449	1	1
1	229,363,272.2	64,221.71621	64,609.37245	2	2
2	114,681,636.1	32,110.85811	32,304.68622	4	4
3	57,340,818.05	16,055.42905	16,152.34311	8	8

4	28,670,409.02	8027.714526	8076.171556	16	16
5	14,335,204.51	4013.857263	4038.085778	32	32
6	7,167,602.256	2006.928632	2019.042889	64	64
7	3,583,801.128	1003.464316	1009.521444	128	128
8	1,791,900.564	501.7321579	504.7607222	256	256
9	895,950.282	250.866079	252.3803611	512	512
10	447,975.141	125.4330395	126.1901806	1024	1024
11	223,987.5705	62.71651974	63.09509028	2048	2048
12	111,993.7852	31.35825987	31.54754514	4096	4096
13	55,996.89262	15.67912993	15.77377257	8192	8192
14	27,998.44631	7.839564967	7.886886285	16,384	16,384
15	13,999.22316	3.919782484	3.943443142	32,768	32,768
16	6999.611578	1.959891242	1.971721571	65,536	65,536
17	3499.805789	0.979945621	0.985860786	131.072	131.072
18	1749.902894	0.48997281	0.492930393	262,144	262,144
19	874.9514472	0.244986405	0.246465196	524,288	524,288
20	437.4757236	0.122493203	0.123232598	1,048,576	1,048,576
21	218.7378618	0.061246601	0.061616299	2,097,152	2,097,152
22	109.3689309	0.030623301	0.03080815	4,194,304	4,194,304
23	54.68446545	0.01531165	0.015404075	8,388,608	8,388,608
24	27.34223273	0.007655825	0.007702037	16,777,216	16,777,216

Table 32. Definition of the UPSArcticWGS84Quad TileMatrixSet

One can define an arbitrary number of zoom levels and does not need to include all the zoom levels defined here. Here, 25 zoom levels are illustrated.

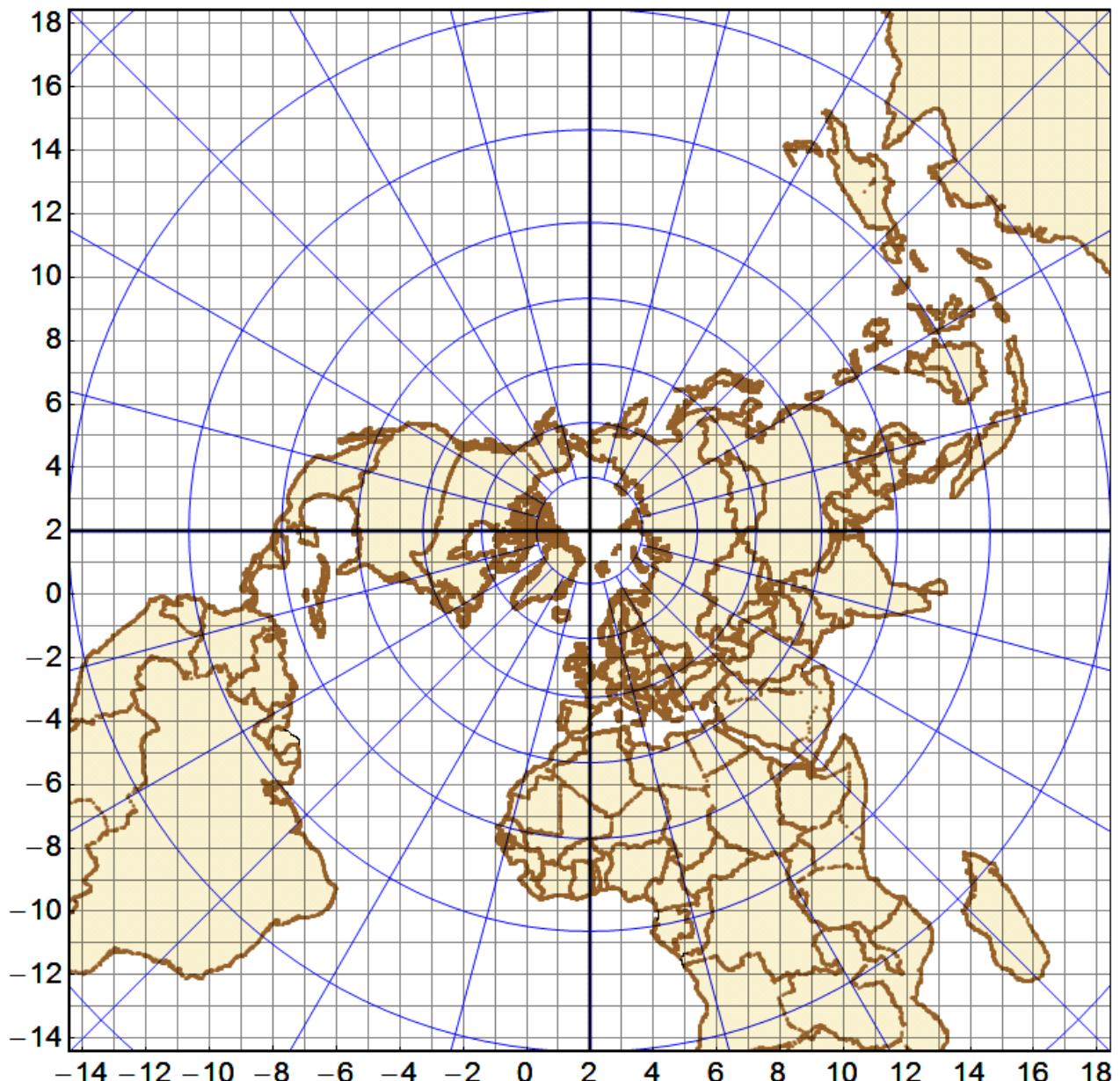


Figure 17. Tile Matrix Id 0 (exterior line; 1x1 tile) and 1 (blank lines; 2x2 tiles) of the UPSArcticWGS84Quad TileMatrixSet (Source NGA)

NOTE

In practice, there are many polar stereographic CRS and it is difficult to prioritize one. In many occasions, the meridian of origin is selected in order to rotate the projection and allow a region to become more prominent. Examples are rotations emphasizing Greenland or Alaska. To be general, the 0 meridian of origin was selected. In addition, polar projections have a variety of standard parallels and here is selected 90N. In the UPS system, the North Pole is assigned the coordinates x=2,000,000, y=2,000,000.

NOTE

NGA has suggested also another TileMatrixSet called *sixteenth tile* that covers less area by having the *PointOfOrigin* at 2,110,189.837563, 6,110,189.837563, BBOX LowerLeft: 2,110,189.837562966, -2,110,189.837562966 and BBOX UpperRight: 6,110,189.837562966, 6,110,189.837562966. The same result can be achieved by using the TileMatrixSetLimits mechanism without having to define an entirely new TileMatrixSet.

Figure 18. Rotation of the meridian of origin to allow a region to become more prominent (a. EPSG:3995, Arctic WGS 84 Polar Stereographic, b. EPSG:3413 WGS 84 / NSIDC Polar Stereographic North, c. EPSG:5936 / Alaska Polar Stereographic). This TileMatrixSet has 0 rotation angle for the meridian of origin.

D.6. Antarctic Universal Polar Stereographic WGS84 Quad (UPSAntarcticWGS84Quad)

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/UPSAntarcticWGS84Quad>

There is no consensus in the geospatial community for what constitutes the “best” tile matrix set for the polar stereographic projection. One reason for this is that the election of the PointOfOrigin and scale denominators is almost completely arbitrary. In this document, a TileMatrixSet identical to the Arctic one but with an Antarctic CRS was selected.

CRS: <http://www.opengis.net/def/crs/EPSC/0/5042>, WGS 84 Universal Polar Stereographic South

BBOX LowerLeft: -14,440,759.350252, -14,440,759.350252

BBOX UpperRight: 18,440,759.350252, 18,440,759.350252

WellKnownScaleSet: N/A

PointOfOrigin: -14,440,759.350252, 18,440,759.350252

TileWidth: 256

TileHeight: 256

Tile Matrix id	Scale Denominator	Cell Size (m) (true at latitude ~-81)	True Cell Size (m) at the pole (informative)	Matrix Width	Matrix Height
0	458,726,544.4	128,443.4324	129,218.7449	1	1
1	229,363,272.2	64,221.71621	64,609.37245	2	2

2	114,681,636.1	32,110.85811	32,304.68622	4	4
3	57,340,818.05	16,055.42905	16,152.34311	8	8
4	28,670,409.02	8027.714526	8076.171556	16	16
5	14,335,204.51	4013.857263	4038.085778	32	32
6	7,167,602.256	2006.928632	2019.042889	64	64
7	3,583,801.128	1003.464316	1009.521444	128	128
8	1,791,900.564	501.7321579	504.7607222	256	256
9	895,950.282	250.866079	252.3803611	512	512
10	447,975.141	125.4330395	126.1901806	1024	1024
11	223,987.5705	62.71651974	63.09509028	2048	2048
12	111,993.7852	31.35825987	31.54754514	4096	4096
13	55,996.89262	15.67912993	15.77377257	8192	8192
14	27,998.44631	7.839564967	7.886886285	16,384	16,384
15	13,999.22316	3.919782484	3.943443142	32,768	32,768
16	6999.611578	1.959891242	1.971721571	65,536	65,536
17	3499.805789	0.979945621	0.985860786	131.072	131.072
18	1749.902894	0.48997281	0.492930393	262,144	262,144
19	874.9514472	0.244986405	0.246465196	524,288	524,288
20	437.4757236	0.122493203	0.123232598	1,048,576	1,048,576
21	218.7378618	0.061246601	0.061616299	2,097,152	2,097,152
22	109.3689309	0.030623301	0.03080815	4,194,304	4,194,304
23	54.68446545	0.01531165	0.015404075	8,388,608	8,388,608
24	27.34223273	0.007655825	0.007702037	16,777,216	16,777,216

Table 33. Definition of the UPSAntarcticWGS84Quad TileMatrixSet

One can define an arbitrary number of zoom levels and does not need to include all the zoom levels defined here. Here, 25 zoom levels are illustrated.

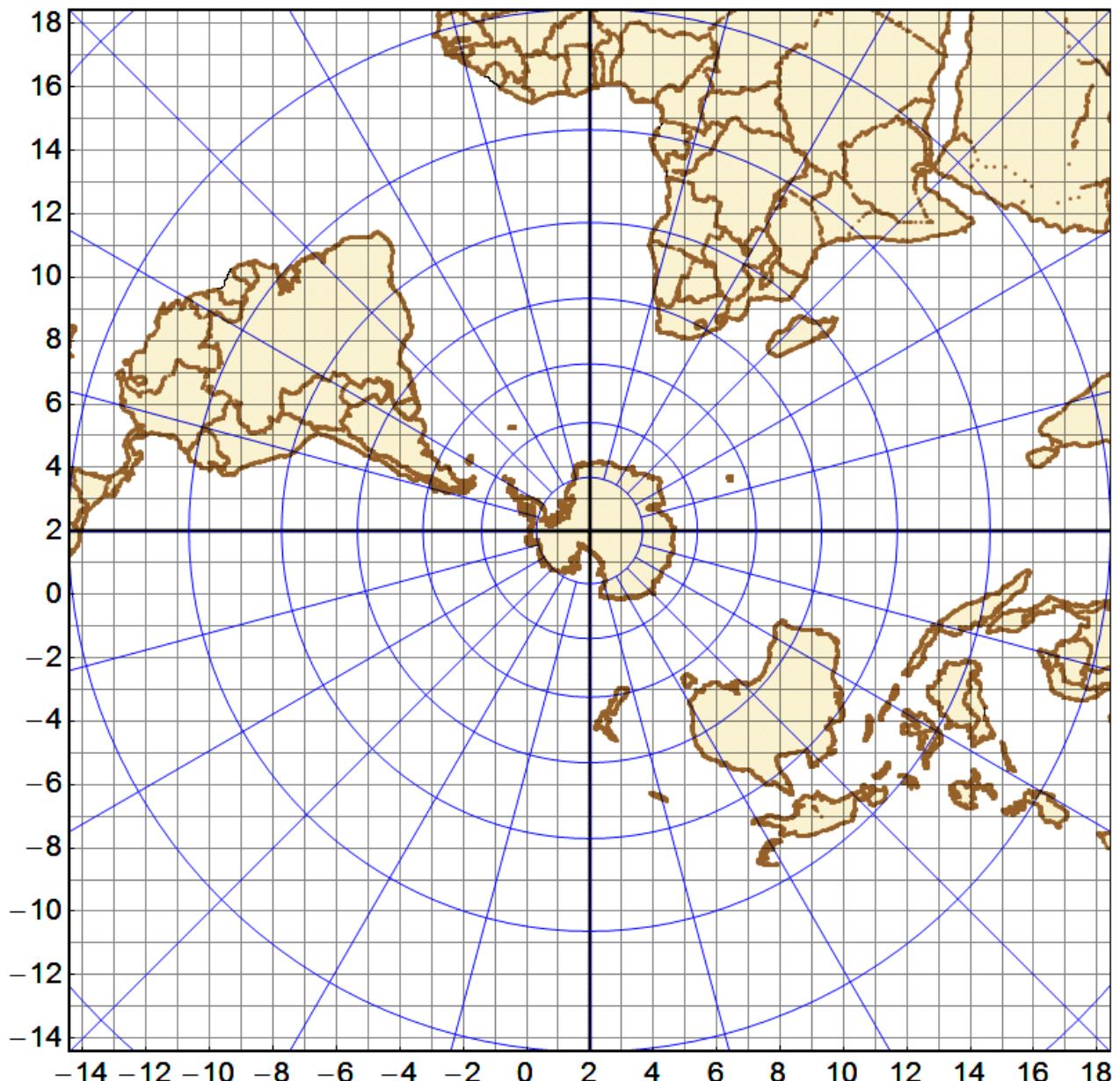


Figure 19. Tile Matrix Id 0 (exterior line; 1x1 tile) and 1 (black lines; 2x2 tiles) of the UPSAntarcticWGS84Quad TileMatrixSet (Source NGA)

D.7. European ETRS89 Lambert azimuthal equal-area Quad (EuropeanETRS89_LAEAQuad)

URI: http://www.opengis.net/def/tilematrixset/OGC/1.0/EuropeanETRS89_LAEAQuad

CRS: <http://www.opengis.net/def/crs/EPSC/0/3035>, ETRS89

BBOX LowerLeft: 1,000,000.0, 2,000,000.0

BBOX UpperRight: 5,500,000.0, 6,500,000.0

WellKnownScaleSet: N/A

PointOfOrigin: 5,500,000.0, 2,000,000.0

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (m)	Matrix Width	Matrix Height
0	62,779,017.857142866	17,578.1250000000	1	1
1	31,389,508.928571433	8789.0625000000	2	2
2	15,694,754.464285716	4394.5312500000	4	4
3	7,847,377.232142858	2197.2656250000	8	8
4	3,923,688.616071429	1098.6328125000	16	16
5	1,961,844.3080357146	549.3164062500	32	32
6	980,922.1540178573	274.6582031250	64	64
7	490,461.07700892864	137.3291015625	128	128
8	245,230.53850446432	68.6645507812	256	256
9	122,615.26925223216	34.3322753906	512	512
10	61,307.63462611608	17.1661376953	1024	1024
11	30,653.81731305804	8.5830688477	2048	2048
12	15,326.90865652902	4.2915344238	4096	4096
13	7663.45432826451	2.1457672119	8192	8192
14	3831.727164132255	1.0728836060	16,384	16,384
15	1915.8635820661275	0.5364418030	32,768	32,768

Table 34. Definition of the EuropeanETRS89_LAEAQuad TileMatrixSet

One can define an arbitrary number of zoom levels and does not need to include all the zoom levels defined here. Here, 16 zoom levels are illustrated.

NOTE

Please note that the EPSG:3035 is defined as: "Cartesian CS. Axes: northing, easting (Y,X). Orientations: north, east. UoM: m". Be aware the axis order is not (X,Y).

D.8. Canadian NAD83 Lambert Conformal Conic (CanadianNAD83_LCC)

URI: http://www.opengis.net/def/tilematrixset/OGC/1.0/CanadianNAD83_LCC

CRS: <http://www.opengis.net/def/crs/EPSG/0/3978>, NAD83

BBOX LowerLeft: -7,786,476.885838887, -5,153,821.09213678

BBOX UpperRight: 7,148,753.233541353, 7,928,343.534071138

WellKnownScaleSet: N/A

PointOfOrigin: -34,655,800, 39,310,000

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Scale Denominator at latitudes 37.897505 and 83.3032475 (informative)	Cartographic product (informative)	Cell Size (m) (true at the equator)	Matrix Width	Matrix Height
0	137,016,643.1	145,000,000	World	38,364.66006	5	5
1	80,320,101.12	85,000,000	World	22,489.62831	8	8
2	47,247,118.3	50,000,000	Atlas SM	13,229.19313	13	14
3	28,348,270.98	30,000,000	Atlas SM	7937.515875	21	22
4	16,536,491.41	17,500,000	Atlas SM	4630.217594	36	38
5	9,449,423.661	10,000,000	Atlas MID	2645.838625	62	66
6	5,669,654.196	6,000,000	Atlas MID	1587.503175	103	110
7	3,307,298.281	3,500,000	Atlas MID	926.0435188	177	188

8	1,889,884.732	2,000,000	Atlas LG	529.167725	309	329
9	1,133,930.839	1,200,000	Atlas LG	317.500635	515	548
10	661,459.6563	700,000	Atlas LG	185.2087038	882	938
11	396,875.7938	420,000	NTDB 250K	111.1252223	1470	1563
12	236,235.5915	250,000	NTDB 250K	66.14596563	2469	2626
13	137,016.6431	145,000	NTDB 250K	38.36466006	4257	4528
14	80,320.10112	85,000	NTDB 50K	22.48962831	7262	7723
15	47,247.1183	50,000	NTDB 50K	13.22919313	12,344	13,130
16	28,348.27098	30,000	NTDB 50K	7.937515875	20,574	21,882
17	16,536.49141	17,500	Geobase	4.630217594	35,269	37,512
18	9449.423661	10,000	Geobase	2.645838625	61,720	65,646
19	5669.654196	6000	Geobase	1.587503175	102,866	109,409
20	3307.298281	3500	Geobase	0.926043519	176,341	187,558
21	1889.884732	2000	Geobase	0.529167725	308,596	328,227
22	1133.930839	1200	Geobase	0.317500635	514,327	547,044
23	661.4596563	700		0.185208704	881,703	937,790
24	396.8757938	420		0.111125222	1,469,505	1,562,983
25	236.2355915	250		0.066145966	2,468,768	2,625,811

Table 35. Definition of the CanadianNAD83_LCC TileMatrixSet

One can define an arbitrary number of zoom levels and does not need to include all the zoom levels defined here. Here, 26 zoom levels are illustrated.

Annex E: Variable width TileMatrixSet definitions (Informative)

This annex includes definitions for TileMatrixSets utilizing the variable width capability. This capability allows for the coalescence of tiles in specific rows so that the overall width of the tile matrix is divided among fewer tiles. This is particularly well suited to define global grids in an Equirectangular Plate Carrée projection, where tiles closer to the pole cover a smaller physical area than tiles at the equator.

Such TileMatrixSets may therefore attempt to approximate equal area tiles. Because the resulting tiles would likely only tend towards equal area but still be off by a significant margin, they cannot be considered Discrete Global Grid Systems (DGGS) per [OGC Abstract Specification Topic 21 - Part 1](#) [<http://www.opengis.net/doc/AS/dggs/1.0>] which specifies a maximum threshold of 1% difference. However, they may qualify as an *Axis-Aligned DGGS* per [OGC Abstract Specification Topic 21 - Part 4](#) introduced in version 2.

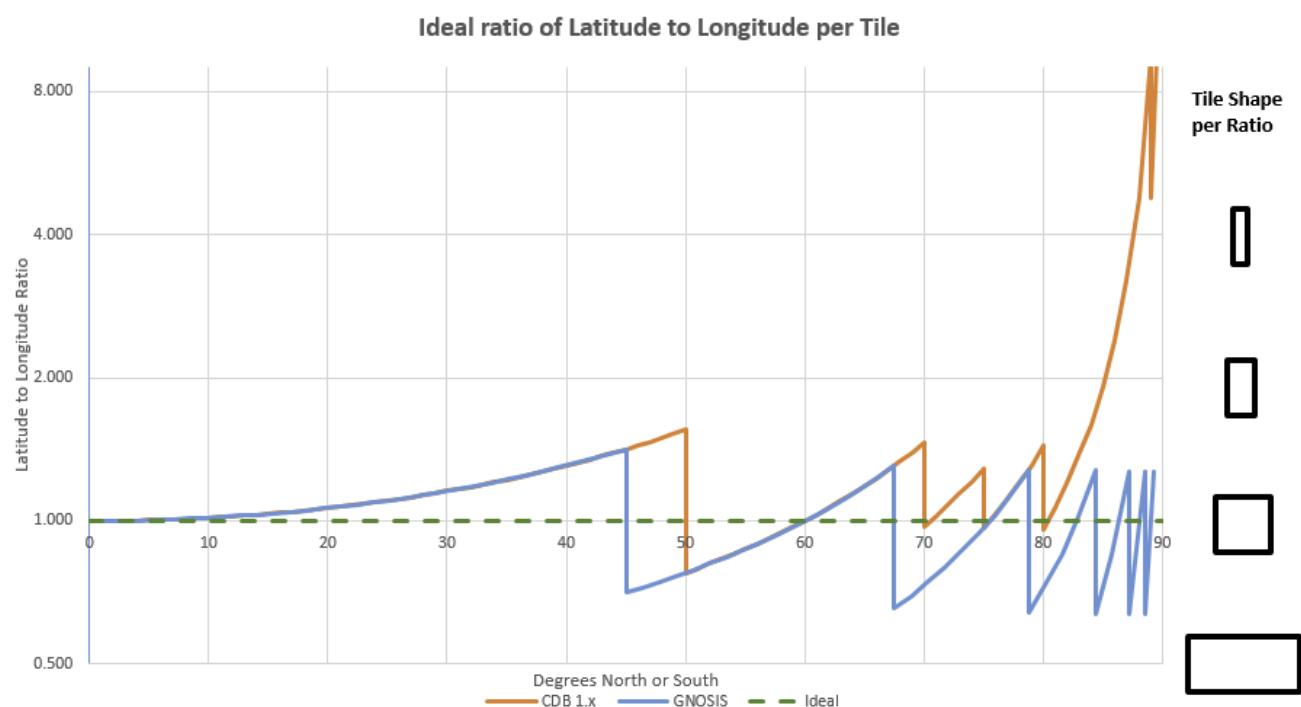


Figure 20. A comparison of variable width tile matrix sets against the ideal tile shape (source: CDB X sprint)

E.1. GNOSISGlobalGrid

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/GNOSISGlobalGrid>

The [GNOSIS](https://ecere.ca/gnosis/) [<https://ecere.ca/gnosis/>] Global Grid Tile Matrix Set defines tiles in the Equirectangular Plate Carrée projection (EPSG:4326 CRS) for the whole world while still attempting to approximate equal area tiles, even near the poles.

Starting at the tile matrix with identifier 1, a variable matrix coalescence factor is applied for

polar tiles to tend towards equal area.

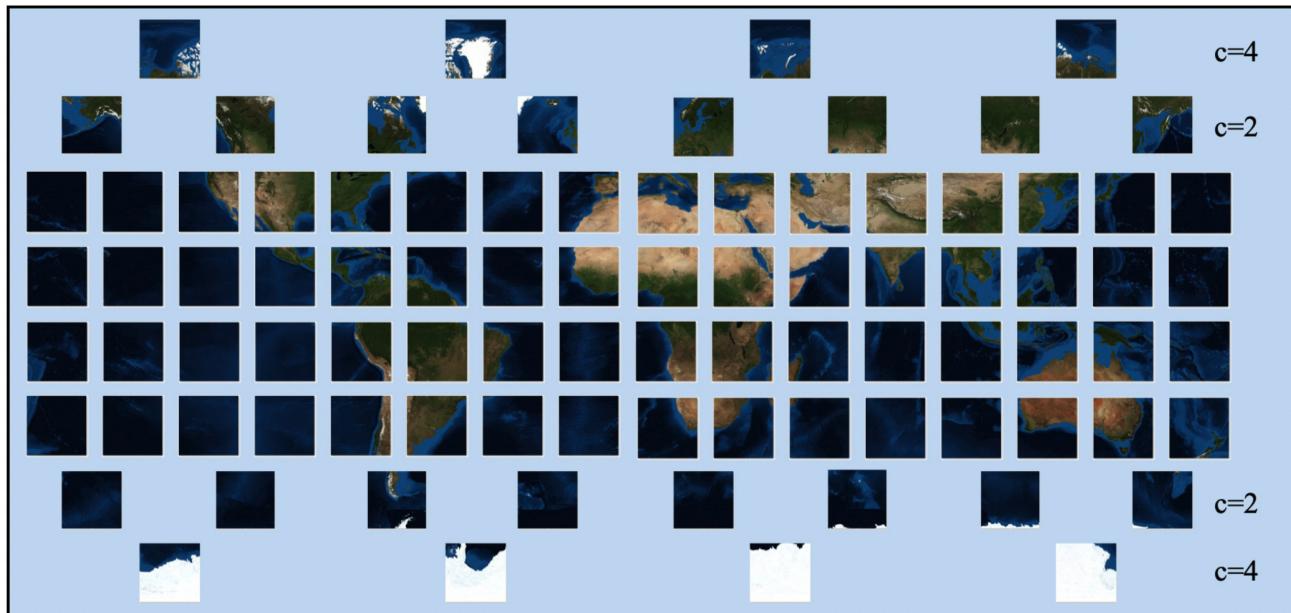


Figure 21. GNOSIS Global Grid (TileMatrix Id 2)

CRS: <http://www.opengis.net/def/crs/EPSC/0/4326>, EPSG:4326

BBOX LowerLeft: -90, -180

BBOX UpperRight: 90, 180

WellKnownScaleSet: <http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad>

PointOfOrigin: 90, -180

TileWidth: 256

TileHeight: 256

TileMatrix id	Scale Denominator	Cell Size (degrees) (true at the equator)	Matrix Width (c=1)	Matrix Height	Coalescing Coefficient (c)
0	139,770,566.007179 4	0.3515625	4	2	
1	69,885,283.0035897 2	0.17578125	8	4	Row 0: 2
					Row 3: 2
2	34,942,641.5017948 6	8.7890625×10^{-2}	16	8	Row 0: 4
					Row 1: 2
					Row 6: 2
					Row 7: 4

3	17,471,320.7508974 3	$4.39453125 \times 10^{-2}$	32	16	<i>Row 0: 8</i>
					<i>Row 1: 4</i>
					<i>Rows 2..3: 2</i>
					<i>Rows 12..13: 2</i>
					<i>Row 14: 4</i>
					<i>Row 15: 8</i>
4	8,735,660.37544871 5	$2.197265625 \times 10^{-2}$	64	32	<i>Row 0: 16</i>
					<i>Row 1: 8</i>
					<i>Rows 2..3: 4</i>
					<i>Rows 4..7: 2</i>
					<i>Rows 24..27: 2</i>
					<i>Rows 28..29: 4</i>
					<i>Row 30: 8</i>
					<i>Row 31: 16</i>
5	4,367,830.18772435 7	$1.0986328125 \times 10^{-2}$	128	64	<i>Row 0: 32</i>
					<i>Row 1: 16</i>
					<i>Rows 2..3: 8</i>
					<i>Rows 4..7: 4</i>
					<i>Rows 8..15: 2</i>
					<i>Rows 48..55: 2</i>
					<i>Rows 56..59: 4</i>
					<i>Rows 60..61: 8</i>
					<i>Row 62: 16</i>
					<i>Row 63: 32</i>

Table 36. Definition of the GNOSISGlobalGrid TileMatrixSet

One can define an arbitrary number of zoom levels and do not need to include all the ones defined here. Here, 6 zoom levels are illustrated. Levels 0 to 28 are officially defined, allowing to reach 2:1 scale while the matrix, row and column identifiers can still fit together within a single 64-bit key.

It is easy to implement procedurally: it is a quad-tree starting with eight 90 x 90 degree tiles at level 0, with the exception that tiles touching a pole are split into 3 tiles rather than 4 — the polar portion is not split longitude-wise. Therefore, there will always be only 4 tiles touching each pole.

NOTE

The GNOSIS Global Grid scales match the GoogleCRS84Quad Well Known Scale Set, but starts at the third scale of the set.

NOTE

Aside from the variable width considerations, the only other difference with the [WorldCRS84Quad TileMatrixSet](#) is the fact that the latter starts at the second scale of the WKSS—its TileMatrix identifier 1 is equivalent to the GNOSISGlobalGrid TileMatrix identifier 0.



Figure 22. GNOSIS Global Grid (TileMatrices Id 0-3)

E.2. CDBGlobalGrid

URI: <http://www.opengis.net/def/tilematrixset/OGC/1.0/CDBGlobalGrid>

This CDB Global Grid Tile Matrix Set defines tiles in the Equirectangular Plate Carrée projection (EPSG:4326 CRS) for the whole world, based on the level of details and zones defined by the [OGC CDB 1.x](#) [<https://docs.ogc.org/is/15-113r6/15-113r6.html>] specifications, which can be encoded following this standard with the use of variable widths.

CRS: <http://www.opengis.net/def/crs/EPSC/0/4326>, EPSG:4326

BBOX LowerLeft: -90, -180

BBOX UpperRight: 90, 180

PointOfOrigin: 90, -180

TileWidth & TileHeight: 1024 (for tile matrices id 0 and up)

TileMatrix id	Scale Denominator	Cell Size (degrees) (true at the equator)	Tile Width & Height	Matrix Width (c=1)	Matrix Height	Coalescing Coefficient (c)
-1	776,503.144	0.001953125	512	360	180	<i>Row 0: 12</i> <i>Rows 1..9: 6</i> <i>Rows 10..14: 4</i> <i>Rows 15..19: 3</i> <i>Rows 20..39: 2</i> <i>Rows 140..159: 2</i> <i>Rows 160..164: 3</i> <i>Rows 165..169: 4</i> <i>Rows 170..178: 6</i> <i>Row 179: 12</i>
0	388,251.572	0.009765625	1024	360	180	<i>Row 0: 12</i> <i>Rows 1..9: 6</i> <i>Rows 10..14: 4</i> <i>Rows 15..19: 3</i> <i>Rows 20..39: 2</i> <i>Rows 140..159: 2</i> <i>Rows 160..164: 3</i> <i>Rows 165..169: 4</i> <i>Rows 170..178: 6</i> <i>Row 179: 12</i>

1	194,125.786	0.004882813	1024	720	360	Row 0: 12
						Rows 1..9: 6
						Rows 10..14: 4
						Rows 15..19: 3
						Rows 20..39: 2
						Rows 140..159: 2
						Rows 160..164: 3
						Rows 165..169: 4
						Rows 170..178: 6
						Row 179: 12

Table 37. Definition of the CDBGlobalGrid TileMatrixSet

For the tile matrices with negative identifiers of the CDB Global Grid, the tiles' geographic extents remain the same as those of tile matrix 0, but the tile size in cells is reduced. The levels -1 to 1 are shown here. For the CDB Global Grid, the polar adjustment zones corresponding to coalescence factors are the same for all tile matrices of the set.

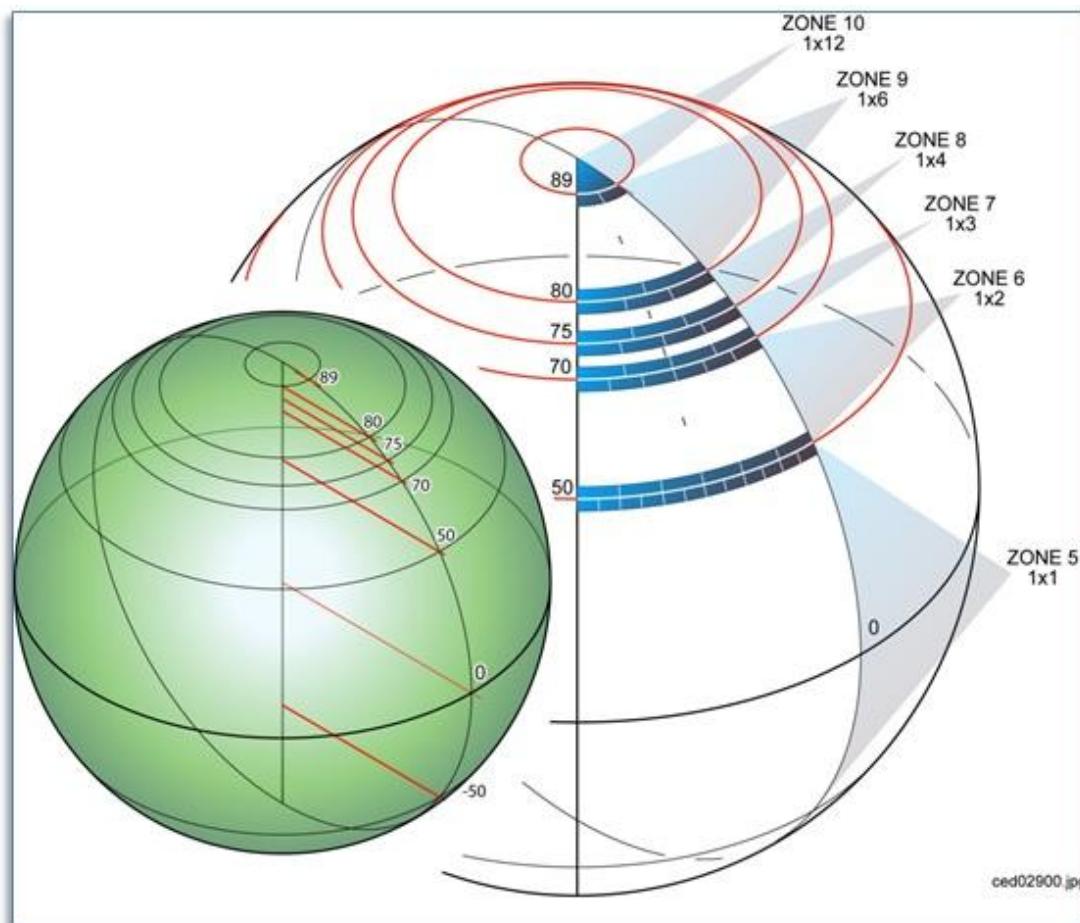


Figure 23. CDB Zones (from OGC CDB Volume 1)

One can define an arbitrary number of zoom levels and do not need to include all the ones

defined here. Here, 3 zoom levels are illustrated.

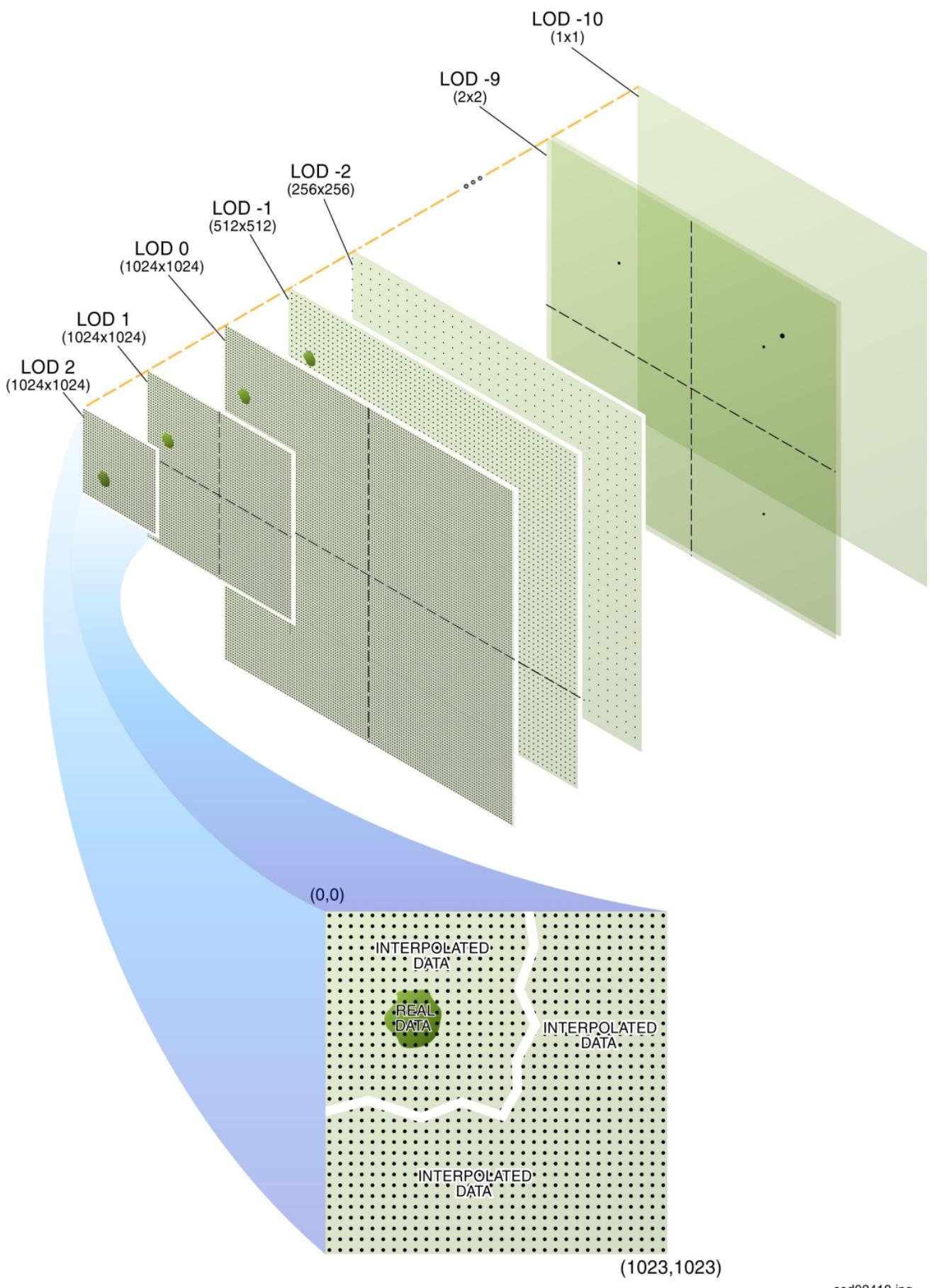


Figure 24. CDB Level of Details (from OGC CDB Volume 1)

Annex F: Example encodings for Common TileMatrixSets (Informative)

This Annex provides examples of JSON and XML encodings than can be used to define tile matrix sets.

F.1. JSON schema for TileMatrixSet

The TileMatrixSet data structure has been transformed into JSON schemas as described in section [TileMatrixSet encodings](#) and provided with this standard as indicated in [Schema Documents \(Normative\)](#). JSON document instances describing TileMatrixSets (including the ones reproduced in this annex) should validate against these schemas. The following JSON Schema fragment reproduces the main module of the TileMatrixSet schema.

```
{  
  "$schema": "https://json-schema.org/draft/2019-09/schema",  
  "title": "Tile Matrix Set Definition",  
  "description": "A definition of a tile matrix set following the Tile Matrix Set standard. For tileset metadata, this property is only required for offline use, as an alternative to tileMatrixSetURI and tileMatrixSetDefinition.",  
  "type": "object",  
  "required": ["id", "crs", "tileMatrices"],  
  "properties": {  
    "title": {  
      "description": "Title of this tile matrix set, normally used for display to a human",  
      "type": "string",  
      "examples": ["Google Maps Compatible for the World"]  
    },  
    "description": {  
      "description": "Brief narrative description of this tile matrix set, normally available for display to a human",  
      "type": "string",  
      "examples": ["The most common TileMatrixSet, used in most of the main IT map browsers. It was initially popularized by Google Maps"]  
    },  
    "keywords": {  
      "description": "Unordered list of one or more commonly used or formalized word(s) or phrase(s) used to describe this tile matrix set",  
      "type": "array",  
      "items": {  
        "type": "string"  
      }  
    }  
  }  
}
```

```

},
"id": {
    "description": "Tile matrix set identifier. Implementation of
'tileMatrixSet'",
    "type": "string"
},
"uri": {
    "description": "Reference to an official source for this
tileMatrixSet",
    "type": "string",
    "format": "uri",
},
"orderedAxes": {
    "type": "array",
    "minItems": 1,
    "items": {
        "type": "string"
    }
},
"crs": {
    "description": "Coordinate Reference System (CRS)",
    "$ref": "crs.json"
},
"wellKnownScaleSet": {
    "description": "Reference to a well-known scale set",
    "type": "string",
    "format": "uri"
},
"boundingBox": {
    "description": "Minimum bounding rectangle surrounding the tile
matrix set, in the supported CRS",
    "$ref": "2DBoundingBox.json"
},
"tileMatrices": {
    "type": "array",
    "description": "Describes scale levels and its tile matrices",
    "items": {
        {
            "$ref": "#/definitions/tileMatrix"
        }
    }
},
"definitions":{
    "tileMatrix": {

```

```

    "description": "",
    "type": "object",
    "required": ["id", "scaleDenominator", "cellSize",
"pointOfOrigin", "tileWidth", "tileHeight", "matrixWidth", "matrixHeight"],
    "properties":
    {
        "title": {
            "description": "Title of this tile matrix set, normally
used for display to a human",
            "type": "string",
            "examples": ["Google Maps Compatible for the World zoom
level 3"]
        },
        "description": {
            "description": "Brief narrative description of this tile
matrix set, normally available for display to a human",
            "type": "string",
            "examples": ["Google Maps Compatible zoom level 3 that is
equivalent to a scale of 1:69885283.00358972 and has 19567.87924100512 meters
of pixel size in the equator"]
        },
        "keywords": {
            "description": "Unordered list of one or more commonly
used or formalized word(s) or phrase(s) used to describe this dataset",
            "type": "array",
            "items": {
                "type": "string"
            }
        },
        "id": {
            "description": "Identifier selecting one of the scales
defined in the TileMatrixSet and representing the scaleDenominator the tile.
Implementation of 'identifier'",
            "type": "string",
            "examples": ["3"]
        },
        "scaleDenominator": {
            "description": "Scale denominator of this tile matrix",
            "type": "number",
            "examples": [69885283]
        },
        "cellSize": {
            "description": "Cell size of this tile matrix",
            "type": "number",
            "examples": [19567.87924]
        },
        "cornerOfOrigin": {

```

```

        "description": "The corner of the tile matrix (_topLeft_ or _bottomLeft_) used as the origin for numbering tile rows and columns. This corner is also a corner of the (0, 0) tile.",
        "type": "string",
        "enum": ["topLeft", "bottomLeft"],
        "default": "topLeft"
    },
    "pointOfOrigin": {
        "description": "Precise position in CRS coordinates of the corner of origin (e.g. the top-left corner) for this tile matrix. This position is also a corner of the (0, 0) tile. In previous version, this was 'topLeftCorner' and 'cornerOfOrigin' did not exist.",
        "$ref": "2DPoint.json",
        "examples": [[-20037508.3427892, 20037508.342789]]
    },
    "tileWidth": {
        "type": "number",
        "description": "Width of each tile of this tile matrix in pixels",
        "format": "integer",
        "minimum": 1,
        "multipleOf": 1,
        "examples": [256]
    },
    "tileHeight": {
        "type": "number",
        "description": "Height of each tile of this tile matrix in pixels",
        "format": "integer",
        "minimum": 1,
        "multipleOf": 1,
        "examples": [256]
    },
    "matrixHeight": {
        "type": "number",
        "description": "Width of the matrix (number of tiles in width)",
        "format": "integer",
        "minimum": 1,
        "multipleOf": 1,
        "examples": [8]
    },
    "matrixWidth": {

```

```

        "type": "number",
        "description": "Height of the matrix (number of tiles in
height)",
        "format": "integer",
        "minimum": 1,
        "multipleOf": 1,
        "examples": [8]
    },
    "variableMatrixWidths": {
        "description": "Describes the rows that has variable
matrix width",
        "type": "array",
        "items": {
            "$ref": "#/definitions/variableMatrixWidth"
        }
    }
},
"variableMatrixWidth": {
    "description": "Variable Matrix Width data structure",
    "type": "object",
    "required": ["coalesce", "minTileRow", "maxTileRow"],
    "properties": {
        "coalesce": {
            "description": "Number of tiles in width that coalesce in
a single tile for these rows",
            "type": "number",
            "format": "integer",
            "minimum": 2,
            "multipleOf": 1
        },
        "minTileRow": {
            "description": "First tile row where the coalescence
factor applies for this tilematrix",
            "type": "number",
            "format": "integer",
            "minimum": 0,
            "multipleOf": 1
        },
        "maxTileRow": {
            "description": "Last tile row where the coalescence
factor applies for this tilematrix",
            "type": "number",
            "format": "integer",
            "minimum": 0,
            "multipleOf": 1
        }
    }
}

```

```

        }
    }
}
}
```

F.2. XML schema for TileMatrixSet

The TileMatrixSet data structure has been transformed into XSD schemas as described in section [TileMatrixSet encodings](#) and provided with this standard as indicated in [Schema Documents \(Normative\)](#). XML document instances describing TileMatrixSets (including the ones reproduced in this annex) should validate against these schemas. The following XML Schema fragment reproduces the main module of the TileMatrixSet schema.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited with XMLSpy v2008 sp1 (http://www.altova.com) by Joan Masó
(UAB-CREAF-MiraMon). Based on previous documents of Keith Pomakis. -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tms="http://www.opengis.net/tms/2.0"
  xmlns:tmSc="http://www.opengis.net/tms/2.0/common"
  targetNamespace="http://www.opengis.net/tms/2.0"
  elementFormDefault="qualified" xml:lang="en">
  <annotation>
    <appinfo>TileMatrixSet 2021-02-02</appinfo>
    <documentation>
      This XML Schema Document encodes the TileMatrixSet data
      structures.
      TMS is an OGC Standard Copyright (c) 2021 Open Geospatial
      Consortium, Inc. All Rights Reserved.
      To obtain additional rights of use, visit
      http://www.opengeospatial.org/legal/.
    </documentation>
  </annotation>
  <!--
***** Includes and imports.

*****
-->
  <!--import namespace="http://www.w3.org/1999/xlink"
  schemaLocation="../../xlink/1.0.0/xlinks.xsd"-->
  <import namespace="http://www.opengis.net/tms/2.0/common"
  schemaLocation="common.xsd"/>
  <!--
***** -->
  <!-- ** The TileMatrixSet element.
```

```

** -->
<!--
***** -->

<element name="TileMatrixSet">
  <annotation>
    <documentation>Describes a particular set of tile
matrices.</documentation>
  </annotation>
  <complexType>
    <complexContent>
      <extension base="tmsc:DescriptionType">
        <sequence>
          <element ref="tmsc:Identifier">
            <annotation>
              <documentation>Tile matrix set
identifier</documentation>
            </annotation>
          </element>
          <element name="uri" type="anyURI" minOccurs="0">
            <annotation>
              <documentation>Reference to an official
source for this tileMatrixSet</documentation>
            </annotation>
          </element>
          <element ref="tmsc:BoundingBox" minOccurs="0">
            <annotation>
              <documentation>
                Minimum bounding rectangle surrounding
the visible layer presented by this tile
matrix
                set, in the supported CRS
              </documentation>
            </annotation>
          </element>
          <element ref="tmsc:CRS">
            <annotation>
              <documentation>Reference to one coordinate
reference
                system (CRS).</documentation>
            </annotation>
          </element>
          <element name="OrderedAxes" type="string"
minOccurs="0">
            <annotation>
              <documentation>
                Names of dimensions in this CRS,
separated by a comma.
              </documentation>
            </annotation>
          </element>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>

```

These names are specified by the CRS definition, but can also be specified here to avoid any ambiguity.

```
</documentation>
    </annotation>
</element>
<element name="WellKnownScaleSet" type="anyURI"
minOccurs="0">
    <annotation>
        <documentation>Reference to a well known
scale set.
```

<http://www.opengis.net/def/wkss/OGC/1.0/GlobalCRS84Scale>,

<http://www.opengis.net/def/wkss/OGC/1.0/GlobalCRS84Pixel>,

<http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad>,

<http://www.opengis.net/def/wkss/OGC/1.0/GoogleMapsCompatible> and

<http://www.opengis.net/def/wkss/OGC/1.0/WorldMercatorWGS84> are
possible values that are defined in Annex C.

It has to be consistent with the

crs and with the ScaleDenominators of the
TileMatrix elements.

```
</documentation>
    </annotation>
</element>
<element ref="tms:TileMatrix" maxOccurs="unbounded">
    <annotation>
        <documentation>Describes a scale level and
its tile matrix.</documentation>
    </annotation>
</element>
<any namespace="#other" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
    <annotation>
        <documentation>This is an extension point for
other properties in the TileMatrixSet.</documentation>
    </annotation>
</any>
</sequence>
<attribute name="id" type="string">
    <annotation>
        <documentation/>
    </annotation>
</attribute>
```

```

        </extension>
    </complexContent>
</complexType>
</element>
<!--
*****
-->
<!-- ** The TileMatrix element.
** -->
<!--
*****
-->
<element name="TileMatrix">
    <annotation>
        <documentation>Describes a particular tile
matrix.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="tmsc:DescriptionType">
                <sequence>
                    <element ref="tmsc:Identifier">
                        <annotation>
                            <documentation>Tile matrix identifier.
Typically an abbreviation of
                                the ScaleDenominator value or its equivalent
pixel size</documentation>
                        </annotation>
                    </element>
                    <element name="ScaleDenominator" type="double">
                        <annotation>
                            <documentation>Scale denominator of this tile
matrix</documentation>
                        </annotation>
                    </element>
                    <element name="CellSize" type="double">
                        <annotation>
                            <documentation>Cell size of this tile
matrix</documentation>
                        </annotation>
                    </element>
                    <element name="CornerOfOrigin" minOccurs="0">
                        <annotation>
                            <documentation>
The corner of the tile matrix (_topLeft_
or _bottomLeft_) used as the origin for numbering tile rows and columns. This
corner is also a corner of the (0, 0) tile.
                            </documentation>
                        </annotation>
                    </element>
                </sequence>
            </extension>
        </complexContent>
    </complexType>
</element>

```

```

<simpleType>
    <restriction base="string">
        <enumeration value="topLeft"/>
        <enumeration value="bottomLeft"/>
    </restriction>
</simpleType>
</element>
<element name="PointOfOrigin"
type="tmsc:PositionType">
    <annotation>
        <documentation>
            Precise position in CRS coordinates of
the corner of origin (e.g. the top-left corner) for this tile matrix.
            This position is also a corner of the (0,
0) tile.
        In previous version, this was
        'topLeftCorner' and 'cornerOfOrigin' did not exist.
        </documentation>
    </annotation>
</element>
<element name="TileWidth" type="positiveInteger">
    <annotation>
        <documentation>Width of each tile of this
tile matrix in pixels.</documentation>
    </annotation>
</element>
<element name="TileHeight" type="positiveInteger">
    <annotation>
        <documentation>Height of each tile of this
tile matrix in pixels</documentation>
    </annotation>
</element>
<element name="MatrixWidth" type="positiveInteger">
    <annotation>
        <documentation>Width of the matrix (number of
tiles in width)</documentation>
    </annotation>
</element>
<element name="MatrixHeight" type="positiveInteger">
    <annotation>
        <documentation>Height of the matrix (number
of tiles in height)</documentation>
    </annotation>
</element>
<element name="VariableMatrixWidth" minOccurs="0"
maxOccurs="unbounded">
    <annotation>

```

```

        <documentation>The rows mention in this array
apply the coalesce mechanisms to fuse n contiguous tiles in the width axis into
a single tile</documentation>
        </annotation>
        <complexType>
            <sequence>
                <element name="Coalesce"
type="positiveInteger">
                    <annotation>
                        <documentation>Number of tiles in
width that coalesce in a single tile for these rows</documentation>
                    </annotation>
                </element>
                <element name="MinTileRow"
type="nonNegativeInteger">
                    <annotation>
                        <documentation>First tile row
where the coalescence factor applies for this tilematrix</documentation>
                    </annotation>
                </element>
                <element name="MaxTileRow"
type="nonNegativeInteger">
                    <annotation>
                        <documentation>Last tile row
where the coalescence factor applies for this tilematrix</documentation>
                    </annotation>
                </element>
            </sequence>
        </complexType>
    </element>
    <any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
        <annotation>
            <documentation>This is an extension point for
other properties in the TileMatrix.</documentation>
        </annotation>
    </any>
</sequence>
</extension>
</complexContent>
</complexType>
</element>
</schema>

```

F.3. Web Mercator Quad

These are the JSON and XML definitions of the WebMercatorQuad tile matrix set (see [WebMercatorQuad](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

One can define an arbitrary number of zoom levels. Here, 25 zoom levels are illustrated.

F.3.1. Web Mercator Quad (JSON encoding)

```
{  
  "id": "WebMercatorQuad",  
  "title": "Google Maps Compatible for the World",  
  "uri": "http://www.opengis.net/def/tilematrixset/OGC/1.0/WebMercatorQuad",  
  "crs": "http://www.opengis.net/def/crs/EPSG/0/3857",  
  "orderedAxes": ["X", "Y"],  
  "wellKnownScaleSet":  
    "http://www.opengis.net/def/wkss/OGC/1.0/GoogleMapsCompatible",  
  "tileMatrices":  
    [  
      {  
        "id": "0",  
        "scaleDenominator": 559082264.028717,  
        "cellSize": 156543.033928041,  
        "pointOfOrigin": [-20037508.3427892, 20037508.3427892],  
        "tileWidth": 256,  
        "tileHeight": 256,  
        "matrixWidth": 1,  
        "matrixHeight": 1  
      },  
      {  
        "id": "1",  
        "scaleDenominator": 279541132.014358,  
        "cellSize": 78271.5169640204,  
        "pointOfOrigin": [-20037508.3427892, 20037508.3427892],  
        "tileWidth": 256,  
        "tileHeight": 256,  
        "matrixWidth": 2,  
        "matrixHeight": 2  
      },  
      {  
        "id": "2",  
        "scaleDenominator": 139770566.007179,  
        "cellSize": 39135.7584820102,  
        "pointOfOrigin": [-20037508.3427892, 20037508.3427892],  
      }  
    ]  
}
```

```
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 4,
"matrixHeight": 4
},
{
  "id": "3",
  "scaleDenominator": 69885283.0035897,
  "cellSize": 19567.8792410051,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8,
  "matrixHeight": 8
},
{
  "id": "4",
  "scaleDenominator": 34942641.5017948,
  "cellSize": 9783.93962050256,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16,
  "matrixHeight": 16
},
{
  "id": "5",
  "scaleDenominator": 17471320.7508974,
  "cellSize": 4891.96981025128,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 32,
  "matrixHeight": 32
},
{
  "id": "6",
  "scaleDenominator": 8735660.37544871,
  "cellSize": 2445.98490512564,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 64,
  "matrixHeight": 64
},
{
  "id": "7",
```

```
"scaleDenominator": 4367830.18772435,
"cellSize": 1222.99245256282,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 128,
"matrixHeight": 128
},
{
"id": "8",
"scaleDenominator": 2183915.09386217,
"cellSize": 611.49622628141,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 256,
"matrixHeight": 256
},
{
"id": "9",
"scaleDenominator": 1091957.54693108,
"cellSize": 305.748113140704,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 512,
"matrixHeight": 512
},
{
"id": "10",
"scaleDenominator": 545978.773465544,
"cellSize": 152.874056570352,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 1024,
"matrixHeight": 1024
},
{
"id": "11",
"scaleDenominator": 272989.386732772,
"cellSize": 76.4370282851762,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 2048,
"matrixHeight": 2048
```

```
},
{
  "id": "12",
  "scaleDenominator": 136494.693366386,
  "cellSize": 38.2185141425881,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4096,
  "matrixHeight": 4096
},
{
  "id": "13",
  "scaleDenominator": 68247.346683193,
  "cellSize": 19.109257071294,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8192,
  "matrixHeight": 8192
},
{
  "id": "14",
  "scaleDenominator": 34123.6733415964,
  "cellSize": 9.55462853564703,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16384,
  "matrixHeight": 16384
},
{
  "id": "15",
  "scaleDenominator": 17061.8366707982,
  "cellSize": 4.77731426782351,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 32768,
  "matrixHeight": 32768
},
{
  "id": "16",
  "scaleDenominator": 8530.91833539913,
  "cellSize": 2.38865713391175,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
```

```
"tileHeight": 256,
"matrixWidth": 65536,
"matrixHeight": 65536
},
{
  "id": "17",
  "scaleDenominator": 4265.45916769956,
  "cellSize": 1.19432856695587,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 131072,
  "matrixHeight": 131072
},
{
  "id": "18",
  "scaleDenominator": 2132.72958384978,
  "cellSize": 0.597164283477939,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 262144,
  "matrixHeight": 262144
},
{
  "id": "19",
  "scaleDenominator": 1066.36479192489,
  "cellSize": 0.29858214173897,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 524288,
  "matrixHeight": 524288
},
{
  "id": "20",
  "scaleDenominator": 533.182395962445,
  "cellSize": 0.149291070869485,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 1048576,
  "matrixHeight": 1048576
},
{
  "id": "21",
  "scaleDenominator": 266.591197981222,
```

```

    "cellSize": 0.0746455354347424,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 2097152,
    "matrixHeight": 2097152
},
{
    "id": "22",
    "scaleDenominator": 133.295598990611,
    "cellSize": 0.0373227677173712,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 4194304,
    "matrixHeight": 4194304
},
{
    "id": "23",
    "scaleDenominator": 66.6477994953056,
    "cellSize": 0.0186613838586856,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 8388608,
    "matrixHeight": 8388608
},
{
    "id": "24",
    "scaleDenominator": 33.3238997476528,
    "cellSize": 0.0093306919293428,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 16777216,
    "matrixHeight": 16777216
}
]
}

```

F.3.2. Web Mercator Quad (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="WebMercatorQuad" xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmsc="http://www.opengis.net/tms/2.0/common"

```

```

xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
  <tmsc:Title>Google Maps Compatible for the World</tmsc:Title>
  <tmsc:Identifier>WebMercatorQuad</tmsc:Identifier>

<uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/WebMercatorQuad</uri>
  <tmsc:CRS>
    <tmsc:URI>http://www.opengis.net/def/crs/EPSG/0/3857</tmsc:URI>
  </tmsc:CRS>
  <OrderedAxes>E,N</OrderedAxes>

<WellKnownScaleSet>http://www.opengis.net/def/wkss/OGC/1.0/GoogleMapsCompatible</WellKnownScaleSet>
  <TileMatrix>
    <tmsc:Identifier>0</tmsc:Identifier>
    <ScaleDenominator>559082264.028717</ScaleDenominator>
    <CellSize>559082264.028717</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1</MatrixWidth>
    <MatrixHeight>1</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmsc:Identifier>1</tmsc:Identifier>
    <ScaleDenominator>279541132.014358</ScaleDenominator>
    <CellSize>279541132.014358</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2</MatrixWidth>
    <MatrixHeight>2</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmsc:Identifier>2</tmsc:Identifier>
    <ScaleDenominator>139770566.007179</ScaleDenominator>
    <CellSize>139770566.007179</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>4</MatrixWidth>
    <MatrixHeight>4</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmsc:Identifier>3</tmsc:Identifier>
    <ScaleDenominator>69885283.0035897</ScaleDenominator>
    <CellSize>69885283.0035897</CellSize>
  </TileMatrix>

```

```
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8</MatrixWidth>
<MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>4</tmsc:Identifier>
    <ScaleDenominator>34942641.5017948</ScaleDenominator>
    <CellSize>34942641.5017948</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>16</MatrixWidth>
    <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>5</tmsc:Identifier>
    <ScaleDenominator>17471320.7508974</ScaleDenominator>
    <CellSize>17471320.7508974</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>32</MatrixWidth>
    <MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>6</tmsc:Identifier>
    <ScaleDenominator>8735660.37544871</ScaleDenominator>
    <CellSize>8735660.37544871</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>64</MatrixWidth>
    <MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>7</tmsc:Identifier>
    <ScaleDenominator>4367830.18772435</ScaleDenominator>
    <CellSize>4367830.18772435</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>128</MatrixWidth>
    <MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
```

```
<tmsc:Identifier>8</tmsc:Identifier>
<ScaleDenominator>2183915.09386217</ScaleDenominator>
<CellSize>2183915.09386217</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>256</MatrixWidth>
<MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>9</tmsc:Identifier>
<ScaleDenominator>1091957.54693108</ScaleDenominator>
<CellSize>1091957.54693108</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>512</MatrixWidth>
<MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>10</tmsc:Identifier>
<ScaleDenominator>545978.773465544</ScaleDenominator>
<CellSize>545978.773465544</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>1024</MatrixWidth>
<MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>11</tmsc:Identifier>
<ScaleDenominator>272989.386732772</ScaleDenominator>
<CellSize>272989.386732772</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2048</MatrixWidth>
<MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>12</tmsc:Identifier>
<ScaleDenominator>136494.693366386</ScaleDenominator>
<CellSize>136494.693366386</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4096</MatrixWidth>
```

```
<MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>13</tmsc:Identifier>
  <ScaleDenominator>68247.346683193</ScaleDenominator>
  <CellSize>68247.346683193</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8192</MatrixWidth>
  <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>14</tmsc:Identifier>
  <ScaleDenominator>34123.6733415964</ScaleDenominator>
  <CellSize>34123.6733415964</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16384</MatrixWidth>
  <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>15</tmsc:Identifier>
  <ScaleDenominator>17061.8366707982</ScaleDenominator>
  <CellSize>17061.8366707982</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>32768</MatrixWidth>
  <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>16</tmsc:Identifier>
  <ScaleDenominator>8530.91833539913</ScaleDenominator>
  <CellSize>8530.91833539913</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>65536</MatrixWidth>
  <MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>17</tmsc:Identifier>
  <ScaleDenominator>4265.45916769956</ScaleDenominator>
  <CellSize>4265.45916769956</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
```

```
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>131072</MatrixWidth>
<MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>18</tmsc:Identifier>
  <ScaleDenominator>2132.72958384978</ScaleDenominator>
  <CellSize>2132.72958384978</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>262144</MatrixWidth>
  <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>19</tmsc:Identifier>
  <ScaleDenominator>1066.36479192489</ScaleDenominator>
  <CellSize>1066.36479192489</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>524288</MatrixWidth>
  <MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>20</tmsc:Identifier>
  <ScaleDenominator>533.182395962445</ScaleDenominator>
  <CellSize>533.182395962445</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>1048576</MatrixWidth>
  <MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>21</tmsc:Identifier>
  <ScaleDenominator>266.591197981222</ScaleDenominator>
  <CellSize>266.591197981222</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2097152</MatrixWidth>
  <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>22</tmsc:Identifier>
```

```

<ScaleDenominator>133.295598990611</ScaleDenominator>
<CellSize>133.295598990611</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4194304</MatrixWidth>
<MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>23</tmsc:Identifier>
  <ScaleDenominator>66.6477994953056</ScaleDenominator>
  <CellSize>66.6477994953056</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8388608</MatrixWidth>
  <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>24</tmsc:Identifier>
  <ScaleDenominator>33.3238997476528</ScaleDenominator>
  <CellSize>33.3238997476528</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16777216</MatrixWidth>
  <MatrixHeight>16777216</MatrixHeight>
</TileMatrix>
</TileMatrixSet>

```

F.4. World CRS84 Quad

Variant 1: World CRS84 Quad (recommended)

These are the JSON and XML definitions of the WorldCRS84Quad tile matrix set (see [Variant 1: World CRS84 Quad \(recommended\)](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

One can define an arbitrary number of zoom levels. Here, 23 zoom levels are illustrated.

F.4.1. World CRS84 Quad (recommended, JSON encoding)

```
{
  "id": "WorldCRS84Quad",
```

```
"title": "CRS84 for the World",
"uri": "http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldCRS84Quad",
"crs": "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
"orderedAxes": [ "Lon", "Lat" ],
"wellKnownScaleSet":  
"http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad",
"tileMatrices": [
{
    "id": "0",
    "scaleDenominator": 279541132.014358,
    "cellSize": 0.703125,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 2,
    "matrixHeight": 1
},
{
    "id": "1",
    "scaleDenominator": 139770566.007179,
    "cellSize": 0.3515625,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 4,
    "matrixHeight": 2
},
{
    "id": "2",
    "scaleDenominator": 69885283.0035897,
    "cellSize": 0.17578125,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 8,
    "matrixHeight": 4
},
{
    "id": "3",
    "scaleDenominator": 34942641.5017948,
    "cellSize": 0.087890625,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 16,
    "matrixHeight": 8
}
],
```

```
{  
    "id": "4",  
    "scaleDenominator": 17471320.7508974,  
    "cellSize": 0.0439453125,  
    "pointOfOrigin": [ -180, 90 ],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 32,  
    "matrixHeight": 16  
},  
{  
    "id": "5",  
    "scaleDenominator": 8735660.37544871,  
    "cellSize": 0.02197265625,  
    "pointOfOrigin": [ -180, 90 ],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 64,  
    "matrixHeight": 32  
},  
{  
    "id": "6",  
    "scaleDenominator": 4367830.18772435,  
    "cellSize": 0.010986328125,  
    "pointOfOrigin": [ -180, 90 ],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 128,  
    "matrixHeight": 64  
},  
{  
    "id": "7",  
    "scaleDenominator": 2183915.09386217,  
    "cellSize": 0.0054931640625,  
    "pointOfOrigin": [ -180, 90 ],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 256,  
    "matrixHeight": 128  
},  
{  
    "id": "8",  
    "scaleDenominator": 1091957.54693108,  
    "cellSize": 0.00274658203125,  
    "pointOfOrigin": [ -180, 90 ],  
    "tileWidth": 256,  
    "tileHeight": 256,
```

```
    "matrixWidth": 512,
    "matrixHeight": 256
},
{
    "id": "9",
    "scaleDenominator": 545978.773465544,
    "cellSize": 0.001373291015625,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 1024,
    "matrixHeight": 512
},
{
    "id": "10",
    "scaleDenominator": 272989.386732772,
    "cellSize": 0.0006866455078125,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 2048,
    "matrixHeight": 1024
},
{
    "id": "11",
    "scaleDenominator": 136494.693366386,
    "cellSize": 0.0034332275390625,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 4096,
    "matrixHeight": 2048
},
{
    "id": "12",
    "scaleDenominator": 68247.346683193,
    "cellSize": 0.00171661376953125,
    "pointOfOrigin": [ -180, 90 ],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 8192,
    "matrixHeight": 4096
},
{
    "id": "13",
    "scaleDenominator": 34123.6733415964,
    "cellSize": 0.000858306884765625,
```

```
"pointOfOrigin": [ -180, 90 ],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 16384,
"matrixHeight": 8192
},
{
  "id": "14",
  "scaleDenominator": 17061.8366707982,
  "cellSize": 0.0000429153442382812,
  "pointOfOrigin": [ -180, 90 ],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 32768,
  "matrixHeight": 16384
},
{
  "id": "15",
  "scaleDenominator": 8530.91833539913,
  "cellSize": 0.0000214576721191406,
  "pointOfOrigin": [ -180, 90 ],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 65536,
  "matrixHeight": 32768
},
{
  "id": "16",
  "scaleDenominator": 4265.45916769956,
  "cellSize": 0.0000107288360595703,
  "pointOfOrigin": [ -180, 90 ],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 131072,
  "matrixHeight": 65536
},
{
  "id": "17",
  "scaleDenominator": 2132.72958384978,
  "cellSize": 0.00000536441802978515,
  "pointOfOrigin": [ -180, 90 ],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 262144,
  "matrixHeight": 131072
},
{
```

```
        "id": "18",
        "scaleDenominator": 1066.36479192489,
        "cellSize": 0.0000268220901489258,
        "pointOfOrigin": [ -180, 90 ],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 524288,
        "matrixHeight": 262144
    },
    {
        "id": "19",
        "scaleDenominator": 533.182395962445,
        "cellSize": 0.0000134110450744629,
        "pointOfOrigin": [ -180, 90 ],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 1048576,
        "matrixHeight": 524288
    },
    {
        "id": "20",
        "scaleDenominator": 266.591197981222,
        "cellSize": 0.0000067055225372314,
        "pointOfOrigin": [ -180, 90 ],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 2097152,
        "matrixHeight": 1048576
    },
    {
        "id": "21",
        "scaleDenominator": 133.295598990611,
        "cellSize": 0.0000033527612686157,
        "pointOfOrigin": [ -180, 90 ],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 4194304,
        "matrixHeight": 2097152
    },
    {
        "id": "22",
        "scaleDenominator": 66.6477994953056,
        "cellSize": 0.0000016763806343079,
        "pointOfOrigin": [ -180, 90 ],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 8388608,
```

```

    "matrixHeight": 4194304
},
{
  "id": "23",
  "scaleDenominator": 33.3238997476528,
  "cellSize": 0.0000008381903171539,
  "pointOfOrigin": [ -180, 90 ],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16777216,
  "matrixHeight": 8388608
}
]
}

```

F.4.2. World CRS84 Quad (recommended, XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="WorldCRS84Quad" xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ..../tilematrixset.xsd">
  <tmsc:Title>CRS84 for the World</tmsc:Title>
  <tmsc:Identifier>WorldCRS84Quad</tmsc:Identifier>
  <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldCRS84Quad</uri>
  <tmsc:CRS>
    <tmsc:URI>http://www.opengis.net/def/crs/OGC/1.3/CRS84</tmsc:URI>
  </tmsc:CRS>
  <OrderedAxes>Lon,Lat</OrderedAxes>

  <WellKnownScaleSet>http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad</WellKnownScaleSet>
    <TileMatrix>
      <tmsc:Identifier>0</tmsc:Identifier>
      <ScaleDenominator>279541132.014358</ScaleDenominator>
      <CellSize>0.703125</CellSize>
      <PointOfOrigin>-180 90</PointOfOrigin>
      <TileWidth>256</TileWidth>
      <TileHeight>256</TileHeight>
      <MatrixWidth>2</MatrixWidth>
      <MatrixHeight>1</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
      <tmsc:Identifier>1</tmsc:Identifier>
      <ScaleDenominator>139770566.007179</ScaleDenominator>
    </TileMatrix>
  </WellKnownScaleSet>
</TileMatrixSet>

```

```
<CellSize>0.3515625</CellSize>
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4</MatrixWidth>
<MatrixHeight>2</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>2</tmsc:Identifier>
<ScaleDenominator>69885283.0035897</ScaleDenominator>
<CellSize>0.17578125</CellSize>
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8</MatrixWidth>
<MatrixHeight>4</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>3</tmsc:Identifier>
<ScaleDenominator>34942641.5017948</ScaleDenominator>
<CellSize>0.087890625</CellSize>
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16</MatrixWidth>
<MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>4</tmsc:Identifier>
<ScaleDenominator>17471320.7508974</ScaleDenominator>
<CellSize>0.0439453125</CellSize>
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>5</tmsc:Identifier>
<ScaleDenominator>8735660.37544871</ScaleDenominator>
<CellSize>0.02197265625</CellSize>
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>32</MatrixHeight>
</TileMatrix>
```

```
<TileMatrix>
  <tmsc:Identifier>6</tmsc:Identifier>
  <ScaleDenominator>4367830.18772435</ScaleDenominator>
  <CellSize>0.010986328125</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>128</MatrixWidth>
  <MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>7</tmsc:Identifier>
  <ScaleDenominator>2183915.09386217</ScaleDenominator>
  <CellSize>0.0054931640625</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>256</MatrixWidth>
  <MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>8</tmsc:Identifier>
  <ScaleDenominator>1091957.54693108</ScaleDenominator>
  <CellSize>0.00274658203125</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>512</MatrixWidth>
  <MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>9</tmsc:Identifier>
  <ScaleDenominator>545978.773465544</ScaleDenominator>
  <CellSize>0.001373291015625</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>1024</MatrixWidth>
  <MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>10</tmsc:Identifier>
  <ScaleDenominator>272989.386732772</ScaleDenominator>
  <CellSize>0.0006866455078125</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
```

```
<MatrixWidth>2048</MatrixWidth>
<MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>11</tmsc:Identifier>
  <ScaleDenominator>136494.693366386</ScaleDenominator>
  <CellSize>0.00034332275390625</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4096</MatrixWidth>
  <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>12</tmsc:Identifier>
  <ScaleDenominator>68247.346683193</ScaleDenominator>
  <CellSize>0.000171661376953125</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8192</MatrixWidth>
  <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>13</tmsc:Identifier>
  <ScaleDenominator>34123.6733415964</ScaleDenominator>
  <CellSize>0.0000858306884765625</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16384</MatrixWidth>
  <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>14</tmsc:Identifier>
  <ScaleDenominator>17061.8366707982</ScaleDenominator>
  <CellSize>0.0000429153442382812</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>32768</MatrixWidth>
  <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>15</tmsc:Identifier>
  <ScaleDenominator>8530.91833539913</ScaleDenominator>
  <CellSize>0.0000214576721191406</CellSize>
```

```
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>65536</MatrixWidth>
<MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>16</tmsc:Identifier>
    <ScaleDenominator>4265.45916769956</ScaleDenominator>
    <CellSize>0.000107288360595703</CellSize>
    <PointOfOrigin>-180 90</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>131072</MatrixWidth>
    <MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>17</tmsc:Identifier>
    <ScaleDenominator>2132.72958384978</ScaleDenominator>
    <CellSize>0.0000536441802978515</CellSize>
    <PointOfOrigin>-180 90</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>262144</MatrixWidth>
    <MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>18</tmsc:Identifier>
    <ScaleDenominator>1066.36479192489</ScaleDenominator>
    <CellSize>0.0000268220901489258</CellSize>
    <PointOfOrigin>-180 90</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>524288</MatrixWidth>
    <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>19</tmsc:Identifier>
    <ScaleDenominator>533.182395962445</ScaleDenominator>
    <CellSize>0.0000134110450744629</CellSize>
    <PointOfOrigin>-180 90</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1048576</MatrixWidth>
    <MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
```

```

<tmsc:Identifier>20</tmsc:Identifier>
<ScaleDenominator>266.591197981222</ScaleDenominator>
<CellSize>0.00000670552253723144</CellSize>
<PointOfOrigin>-180 90</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2097152</MatrixWidth>
<MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>21</tmsc:Identifier>
  <ScaleDenominator>133.295598990611</ScaleDenominator>
  <CellSize>0.00000335276126861572</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4194304</MatrixWidth>
  <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>22</tmsc:Identifier>
  <ScaleDenominator>66.6477994953056</ScaleDenominator>
  <CellSize>0.00000167638063430786</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8388608</MatrixWidth>
  <MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>23</tmsc:Identifier>
  <ScaleDenominator>33.3238997476528</ScaleDenominator>
  <CellSize>0.00000083819031715393</CellSize>
  <PointOfOrigin>-180 90</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16777216</MatrixWidth>
  <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
</TileMatrixSet>

```

Variant 2: World EPSG:4326 Quad

These are the JSON and XML definitions of the WorldCRS84Quad tile matrix based on EPSG:4326 (see [Variant 2: World EPSG:4326 Quad](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other

TileMatrices for more detailed scales is possible if they follow the same pattern.

One can define an arbitrary number of zoom levels. Here, 23 zoom levels are illustrated.

F.4.3. World EPSG:4326 Quad (JSON encoding)

```
{  
  "id": "WorldCRS84Quad",  
  "title": "EPSG:4326 for the World",  
  "uri": "http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldCRS84Quad",  
  "crs": "http://www.opengis.net/def/crs/EPSC/0/4326",  
  "orderedAxes": [ "Lat", "Lon" ],  
  "wellKnownScaleSet":  
    "http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad",  
  "tileMatrices": [  
    {  
      "id": "0",  
      "scaleDenominator": 279541132.014358,  
      "cellSize": 0.703125,  
      "pointOfOrigin": [ 90, -180],  
      "tileWidth": 256,  
      "tileHeight": 256,  
      "matrixWidth": 2,  
      "matrixHeight": 1  
    },  
    {  
      "id": "1",  
      "scaleDenominator": 139770566.007179,  
      "cellSize": 0.3515625,  
      "pointOfOrigin": [ 90, -180],  
      "tileWidth": 256,  
      "tileHeight": 256,  
      "matrixWidth": 4,  
      "matrixHeight": 2  
    },  
    {  
      "id": "2",  
      "scaleDenominator": 69885283.0035897,  
      "cellSize": 0.17578125,  
      "pointOfOrigin": [ 90, -180],  
      "tileWidth": 256,  
      "tileHeight": 256,  
      "matrixWidth": 8,  
      "matrixHeight": 4  
    },  
    {  
    }]
```

```
        "id": "3",
        "scaleDenominator": 34942641.5017948,
        "cellSize": 0.087890625,
        "pointOfOrigin": [ 90, -180],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 16,
        "matrixHeight": 8
    },
    {
        "id": "4",
        "scaleDenominator": 17471320.7508974,
        "cellSize": 0.0439453125,
        "pointOfOrigin": [ 90, -180],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 32,
        "matrixHeight": 16
    },
    {
        "id": "5",
        "scaleDenominator": 8735660.37544871,
        "cellSize": 0.02197265625,
        "pointOfOrigin": [ 90, -180],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 64,
        "matrixHeight": 32
    },
    {
        "id": "6",
        "scaleDenominator": 4367830.18772435,
        "cellSize": 0.010986328125,
        "pointOfOrigin": [ 90, -180],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 128,
        "matrixHeight": 64
    },
    {
        "id": "7",
        "scaleDenominator": 2183915.09386217,
        "cellSize": 0.0054931640625,
        "pointOfOrigin": [ 90, -180],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 256,
```

```
    "matrixHeight": 128
},
{
  "id": "8",
  "scaleDenominator": 1091957.54693108,
  "cellSize": 0.00274658203125,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 512,
  "matrixHeight": 256
},
{
  "id": "9",
  "scaleDenominator": 545978.773465544,
  "cellSize": 0.001373291015625,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 1024,
  "matrixHeight": 512
},
{
  "id": "10",
  "scaleDenominator": 272989.386732772,
  "cellSize": 0.0006866455078125,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 2048,
  "matrixHeight": 1024
},
{
  "id": "11",
  "scaleDenominator": 136494.693366386,
  "cellSize": 0.00034332275390625,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4096,
  "matrixHeight": 2048
},
{
  "id": "12",
  "scaleDenominator": 68247.346683193,
  "cellSize": 0.000171661376953125,
  "pointOfOrigin": [ 90, -180],
```

```
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 8192,  
"matrixHeight": 4096  
},  
{  
    "id": "13",  
    "scaleDenominator": 34123.6733415964,  
    "cellSize": 0.000858306884765625,  
    "pointOfOrigin": [ 90, -180],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 16384,  
    "matrixHeight": 8192  
},  
{  
    "id": "14",  
    "scaleDenominator": 17061.8366707982,  
    "cellSize": 0.000429153442382812,  
    "pointOfOrigin": [ 90, -180],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 32768,  
    "matrixHeight": 16384  
},  
{  
    "id": "15",  
    "scaleDenominator": 8530.91833539913,  
    "cellSize": 0.000214576721191406,  
    "pointOfOrigin": [ 90, -180],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 65536,  
    "matrixHeight": 32768  
},  
{  
    "id": "16",  
    "scaleDenominator": 4265.45916769956,  
    "cellSize": 0.000107288360595703,  
    "pointOfOrigin": [ 90, -180],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 131072,  
    "matrixHeight": 65536  
},  
{  
    "id": "17",
```

```
"scaleDenominator": 2132.72958384978,
"cellSize": 0.00000536441802978515,
"pointOfOrigin": [ 90, -180],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 262144,
"matrixHeight": 131072
},
{
  "id": "18",
  "scaleDenominator": 1066.36479192489,
  "cellSize": 0.00000268220901489258,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 524288,
  "matrixHeight": 262144
},
{
  "id": "19",
  "scaleDenominator": 533.182395962445,
  "cellSize": 0.00000134110450744629,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 1048576,
  "matrixHeight": 524288
},
{
  "id": "20",
  "scaleDenominator": 266.591197981222,
  "cellSize": 0.00000067055225372314,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 2097152,
  "matrixHeight": 1048576
},
{
  "id": "21",
  "scaleDenominator": 133.295598990611,
  "cellSize": 0.00000033527612686157,
  "pointOfOrigin": [ 90, -180],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4194304,
  "matrixHeight": 2097152
```

```

    },
    {
        "id": "22",
        "scaleDenominator": 66.6477994953056,
        "cellSize": 0.0000016763806343079,
        "pointOfOrigin": [ 90, -180],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 8388608,
        "matrixHeight": 4194304
    },
    {
        "id": "23",
        "scaleDenominator": 33.3238997476528,
        "cellSize": 0.0000008381903171539,
        "pointOfOrigin": [ 90, -180],           "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 16777216,
        "matrixHeight": 8388608
    }
]
}

```

F.4.4. World EPSG:4326 Quad (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="WorldCRS84Quad" xmlns="http://www.opengis.net/tms/2.0"
    xmlns:tmse="http://www.opengis.net/tms/2.0/common"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
    <tmse:Title>EPSG:4326 for the World</tmse:Title>
    <tmse:Identifier>WorldCRS84Quad</tmse:Identifier>
    <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldCRS84Quad</uri>
    <tmse:CRS>
        <tmse:URI>http://www.opengis.net/def/crs/EPSG/0/4326</tmse:URI>
    </tmse:CRS>
    <OrderedAxes>Lat,Lon</OrderedAxes>

    <WellKnownScaleSet>http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad</WellKnownScaleSet>
        <TileMatrix>
            <tmse:Identifier>0</tmse:Identifier>
            <ScaleDenominator>279541132.014358</ScaleDenominator>
            <CellSize>0.703125</CellSize>
            <PointOfOrigin>90 -180</PointOfOrigin>

```

```
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2</MatrixWidth>
<MatrixHeight>1</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>1</tmsc:Identifier>
<ScaleDenominator>139770566.007179</ScaleDenominator>
<CellSize>0.3515625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4</MatrixWidth>
<MatrixHeight>2</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>2</tmsc:Identifier>
<ScaleDenominator>69885283.0035897</ScaleDenominator>
<CellSize>0.17578125</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8</MatrixWidth>
<MatrixHeight>4</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>3</tmsc:Identifier>
<ScaleDenominator>34942641.5017948</ScaleDenominator>
<CellSize>0.087890625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16</MatrixWidth>
<MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>4</tmsc:Identifier>
<ScaleDenominator>17471320.7508974</ScaleDenominator>
<CellSize>0.0439453125</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>5</tmsc:Identifier>
```

```
<ScaleDenominator>8735660.37544871</ScaleDenominator>
<CellSize>0.02197265625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>6</tmsc:Identifier>
<ScaleDenominator>4367830.18772435</ScaleDenominator>
<CellSize>0.010986328125</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>128</MatrixWidth>
<MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>7</tmsc:Identifier>
<ScaleDenominator>2183915.09386217</ScaleDenominator>
<CellSize>0.0054931640625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>256</MatrixWidth>
<MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>8</tmsc:Identifier>
<ScaleDenominator>1091957.54693108</ScaleDenominator>
<CellSize>0.00274658203125</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>512</MatrixWidth>
<MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>9</tmsc:Identifier>
<ScaleDenominator>545978.773465544</ScaleDenominator>
<CellSize>0.001373291015625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>1024</MatrixWidth>
<MatrixHeight>512</MatrixHeight>
```

```
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>10</tmsc:Identifier>
  <ScaleDenominator>272989.386732772</ScaleDenominator>
  <CellSize>0.0006866455078125</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2048</MatrixWidth>
  <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>11</tmsc:Identifier>
  <ScaleDenominator>136494.693366386</ScaleDenominator>
  <CellSize>0.00034332275390625</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4096</MatrixWidth>
  <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>12</tmsc:Identifier>
  <ScaleDenominator>68247.346683193</ScaleDenominator>
  <CellSize>0.000171661376953125</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8192</MatrixWidth>
  <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>13</tmsc:Identifier>
  <ScaleDenominator>34123.6733415964</ScaleDenominator>
  <CellSize>0.0000858306884765625</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16384</MatrixWidth>
  <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>14</tmsc:Identifier>
  <ScaleDenominator>17061.8366707982</ScaleDenominator>
  <CellSize>0.0000429153442382812</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>256</TileWidth>
```

```
<TileHeight>256</TileHeight>
<MatrixWidth>32768</MatrixWidth>
<MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>15</tmsc:Identifier>
    <ScaleDenominator>8530.91833539913</ScaleDenominator>
    <CellSize>0.000214576721191406</CellSize>
    <PointOfOrigin>90 -180</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>65536</MatrixWidth>
    <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>16</tmsc:Identifier>
    <ScaleDenominator>4265.45916769956</ScaleDenominator>
    <CellSize>0.000107288360595703</CellSize>
    <PointOfOrigin>90 -180</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>131072</MatrixWidth>
    <MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>17</tmsc:Identifier>
    <ScaleDenominator>2132.72958384978</ScaleDenominator>
    <CellSize>0.0000536441802978515</CellSize>
    <PointOfOrigin>90 -180</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>262144</MatrixWidth>
    <MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>18</tmsc:Identifier>
    <ScaleDenominator>1066.36479192489</ScaleDenominator>
    <CellSize>0.0000268220901489258</CellSize>
    <PointOfOrigin>90 -180</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>524288</MatrixWidth>
    <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>19</tmsc:Identifier>
    <ScaleDenominator>533.182395962445</ScaleDenominator>
```

```
<CellSize>0.00000134110450744629</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>1048576</MatrixWidth>
<MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>20</tmsc:Identifier>
<ScaleDenominator>266.591197981222</ScaleDenominator>
<CellSize>0.00000670552253723144</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2097152</MatrixWidth>
<MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>21</tmsc:Identifier>
<ScaleDenominator>133.295598990611</ScaleDenominator>
<CellSize>0.00000335276126861572</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4194304</MatrixWidth>
<MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>22</tmsc:Identifier>
<ScaleDenominator>66.6477994953056</ScaleDenominator>
<CellSize>0.00000167638063430786</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8388608</MatrixWidth>
<MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>23</tmsc:Identifier>
<ScaleDenominator>33.3238997476528</ScaleDenominator>
<CellSize>0.00000083819031715393</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16777216</MatrixWidth>
<MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
```

```
</TileMatrixSet>
```

F.5. World Mercator WGS84 Quad

These are the JSON and XML definitions of the WorldMercatorQuad tile matrix set (see [WorldMercatorWGS84Quad](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

One can define an arbitrary number of zoom levels. Here, 19 zoom levels are illustrated but resolutions up to 24 are currently available in some mass market services.

F.5.1. World Mercator WGS84 Quad (JSON encoding)

```
{
  "id": "WorldMercatorWGS84Quad",
  "title": "World Mercator WGS84 (ellipsoid)",
  "uri":
  "http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldMercatorWGS84Quad",
  "crs": "http://www.opengis.net/def/crs/EPSG/0/3395",
  "orderedAxes": ["E", "N"],
  "wellKnownScaleSet":
  "http://www.opengis.net/def/wkss/OGC/1.0/WorldMercatorWGS84",
  "tileMatrices":
  [
    {
      "id": "0",
      "scaleDenominator": 559082264.028717,
      "cellSize": 156543.033928041,
      "pointOfOrigin": [-20037508.3427892,20037508.3427892],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 1,
      "matrixHeight": 1
    },
    {
      "id": "1",
      "scaleDenominator": 279541132.014358,
      "cellSize": 78271.5169640204,
      "pointOfOrigin": [-20037508.3427892,20037508.3427892],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 2,
      "matrixHeight": 2
    },
  ]}
```

```
{  
    "id": "2",  
    "scaleDenominator": 139770566.007179,  
    "cellSize": 39135.7584820102,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 4,  
    "matrixHeight": 4  
},  
{  
    "id": "3",  
    "scaleDenominator": 69885283.0035897,  
    "cellSize": 19567.8792410051,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 8,  
    "matrixHeight": 8  
},  
{  
    "id": "4",  
    "scaleDenominator": 34942641.5017948,  
    "cellSize": 9783.93962050256,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 16,  
    "matrixHeight": 16  
},  
{  
    "id": "5",  
    "scaleDenominator": 17471320.7508974,  
    "cellSize": 4891.96981025128,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 32,  
    "matrixHeight": 32  
},  
{  
    "id": "6",  
    "scaleDenominator": 8735660.37544871,  
    "cellSize": 2445.98490512564,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,
```

```
    "matrixWidth": 64,
    "matrixHeight": 64
  },
  {
    "id": "7",
    "scaleDenominator": 4367830.18772435,
    "cellSize": 1222.99245256282,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 128,
    "matrixHeight": 128
  },
  {
    "id": "8",
    "scaleDenominator": 2183915.09386217,
    "cellSize": 611.49622628141,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 256,
    "matrixHeight": 256
  },
  {
    "id": "9",
    "scaleDenominator": 1091957.54693108,
    "cellSize": 305.748113140704,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 512,
    "matrixHeight": 512
  },
  {
    "id": "10",
    "scaleDenominator": 545978.773465544,
    "cellSize": 152.874056570352,
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 1024,
    "matrixHeight": 1024
  },
  {
    "id": "11",
    "scaleDenominator": 272989.386732772,
    "cellSize": 76.4370282851762,
```

```
"pointOfOrigin": [-20037508.3427892,20037508.3427892],  
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 2048,  
"matrixHeight": 2048  
},  
{  
    "id": "12",  
    "scaleDenominator": 136494.693366386,  
    "cellSize": 38.2185141425881,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 4096,  
    "matrixHeight": 4096  
},  
{  
    "id": "13",  
    "scaleDenominator": 68247.346683193,  
    "cellSize": 19.109257071294,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 8192,  
    "matrixHeight": 8192  
},  
{  
    "id": "14",  
    "scaleDenominator": 34123.6733415964,  
    "cellSize": 9.55462853564703,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 16384,  
    "matrixHeight": 16384  
},  
{  
    "id": "15",  
    "scaleDenominator": 17061.8366707982,  
    "cellSize": 4.77731426782351,  
    "pointOfOrigin": [-20037508.3427892,20037508.3427892],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 32768,  
    "matrixHeight": 32768  
},  
{
```

```
"id": "16",
"scaleDenominator": 8530.91833539913,
"cellSize": 2.38865713391175,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 65536,
"matrixHeight": 65536
},
{
"id": "17",
"scaleDenominator": 4265.45916769956,
"cellSize": 1.19432856695587,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 131072,
"matrixHeight": 131072
},
{
"id": "18",
"scaleDenominator": 2132.72958384978,
"cellSize": 0.597164283477939,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 262144,
"matrixHeight": 262144
},
{
"id": "19",
"scaleDenominator": 1066.36479192489,
"cellSize": 0.29858214173897,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 524288,
"matrixHeight": 524288
},
{
"id": "20",
"scaleDenominator": 533.182395962445,
"cellSize": 0.149291070869485,
"pointOfOrigin": [-20037508.3427892,20037508.3427892],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 1048576,
```

```
"matrixHeight": 1048576
},
{
  "id": "21",
  "scaleDenominator": 266.591197981222,
  "cellSize": 0.0746455354347424,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 2097152,
  "matrixHeight": 2097152
},
{
  "id": "22",
  "scaleDenominator": 133.295598990611,
  "cellSize": 0.0373227677173712,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4194304,
  "matrixHeight": 4194304
},
{
  "id": "23",
  "scaleDenominator": 66.6477994953056,
  "cellSize": 0.0186613838586856,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8388608,
  "matrixHeight": 8388608
},
{
  "id": "24",
  "scaleDenominator": 33.3238997476528,
  "cellSize": 0.0093306919293428,
  "pointOfOrigin": [-20037508.3427892,20037508.3427892],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16777216,
  "matrixHeight": 16777216
}
]
```

F.5.2. World Mercator WGS84 Quad (XML encoding)

```
<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="WorldMercatorWGS84Quad"
  xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmSc="http://www.opengis.net/tms/2.0/common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
  <tmSc:Title>World Mercator WGS84 (ellipsoid)</tmSc:Title>
  <tmSc:Identifier>WorldMercatorWGS84Quad</tmSc:Identifier>

  <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/WorldMercatorWGS84Quad</uri>
  <tmSc:CRS>
    <tmSc:URI>http://www.opengis.net/def/crs/EPSG/0/3395</tmSc:URI>
  </tmSc:CRS>
  <OrderedAxes>E,N</OrderedAxes>

  <WellKnownScaleSet>http://www.opengis.net/def/wkss/OGC/1.0/WorldMercatorWGS84
  </WellKnownScaleSet>
  <TileMatrix>
    <tmSc:Identifier>0</tmSc:Identifier>
    <ScaleDenominator>559082264.028717</ScaleDenominator>
    <CellSize>156543.033928041</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1</MatrixWidth>
    <MatrixHeight>1</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmSc:Identifier>1</tmSc:Identifier>
    <ScaleDenominator>279541132.014358</ScaleDenominator>
    <CellSize>78271.5169640204</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2</MatrixWidth>
    <MatrixHeight>2</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmSc:Identifier>2</tmSc:Identifier>
    <ScaleDenominator>139770566.007179</ScaleDenominator>
    <CellSize>39135.7584820102</CellSize>
    <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
    <TileWidth>256</TileWidth>
```

```
<TileHeight>256</TileHeight>
<MatrixWidth>4</MatrixWidth>
<MatrixHeight>4</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>3</tmsc:Identifier>
  <ScaleDenominator>69885283.0035897</ScaleDenominator>
  <CellSize>19567.8792410051</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8</MatrixWidth>
  <MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>4</tmsc:Identifier>
  <ScaleDenominator>34942641.5017948</ScaleDenominator>
  <CellSize>9783.93962050256</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16</MatrixWidth>
  <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>5</tmsc:Identifier>
  <ScaleDenominator>17471320.7508974</ScaleDenominator>
  <CellSize>4891.96981025128</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>32</MatrixWidth>
  <MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>6</tmsc:Identifier>
  <ScaleDenominator>8735660.37544871</ScaleDenominator>
  <CellSize>2445.98490512564</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>64</MatrixWidth>
  <MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>7</tmsc:Identifier>
  <ScaleDenominator>4367830.18772435</ScaleDenominator>
```

```
<CellSize>1222.99245256282</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>128</MatrixWidth>
<MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>8</tmsc:Identifier>
<ScaleDenominator>2183915.09386217</ScaleDenominator>
<CellSize>611.49622628141</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>256</MatrixWidth>
<MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>9</tmsc:Identifier>
<ScaleDenominator>1091957.54693108</ScaleDenominator>
<CellSize>305.748113140704</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>512</MatrixWidth>
<MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>10</tmsc:Identifier>
<ScaleDenominator>545978.773465544</ScaleDenominator>
<CellSize>152.874056570352</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>1024</MatrixWidth>
<MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>11</tmsc:Identifier>
<ScaleDenominator>272989.386732772</ScaleDenominator>
<CellSize>76.4370282851762</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2048</MatrixWidth>
<MatrixHeight>2048</MatrixHeight>
</TileMatrix>
```

```
<TileMatrix>
  <tmsc:Identifier>12</tmsc:Identifier>
  <ScaleDenominator>136494.693366386</ScaleDenominator>
  <CellSize>38.2185141425881</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4096</MatrixWidth>
  <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>13</tmsc:Identifier>
  <ScaleDenominator>68247.346683193</ScaleDenominator>
  <CellSize>19.109257071294</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8192</MatrixWidth>
  <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>14</tmsc:Identifier>
  <ScaleDenominator>34123.6733415964</ScaleDenominator>
  <CellSize>9.55462853564703</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16384</MatrixWidth>
  <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>15</tmsc:Identifier>
  <ScaleDenominator>17061.8366707982</ScaleDenominator>
  <CellSize>4.77731426782351</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>32768</MatrixWidth>
  <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>16</tmsc:Identifier>
  <ScaleDenominator>8530.91833539913</ScaleDenominator>
  <CellSize>2.38865713391175</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
```

```
<MatrixWidth>65536</MatrixWidth>
<MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>17</tmsc:Identifier>
<ScaleDenominator>4265.45916769956</ScaleDenominator>
<CellSize>1.19432856695587</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>131072</MatrixWidth>
<MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>18</tmsc:Identifier>
<ScaleDenominator>2132.72958384978</ScaleDenominator>
<CellSize>0.597164283477939</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>262144</MatrixWidth>
<MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>19</tmsc:Identifier>
<ScaleDenominator>1066.36479192489</ScaleDenominator>
<CellSize>0.29858214173897</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>524288</MatrixWidth>
<MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>20</tmsc:Identifier>
<ScaleDenominator>533.182395962445</ScaleDenominator>
<CellSize>0.149291070869485</CellSize>
<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>1048576</MatrixWidth>
<MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>21</tmsc:Identifier>
<ScaleDenominator>266.591197981222</ScaleDenominator>
<CellSize>0.0746455354347424</CellSize>
```

```

<PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2097152</MatrixWidth>
<MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>22</tmsc:Identifier>
  <ScaleDenominator>133.295598990611</ScaleDenominator>
  <CellSize>0.0373227677173712</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4194304</MatrixWidth>
  <MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>23</tmsc:Identifier>
  <ScaleDenominator>66.6477994953056</ScaleDenominator>
  <CellSize>0.0186613838586856</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8388608</MatrixWidth>
  <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>24</tmsc:Identifier>
  <ScaleDenominator>33.3238997476528</ScaleDenominator>
  <CellSize>0.0093306919293428</CellSize>
  <PointOfOrigin>-20037508.3427892 20037508.3427892</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16777216</MatrixWidth>
  <MatrixHeight>16777216</MatrixHeight>
</TileMatrix>
</TileMatrixSet>

```

F.6. Universal Transverse Mercator WGS84 Quad for zone 31

These are the JSON and XML definitions of the UTM31WGS84Quad tile matrix set (see [Universal Transverse Mercator WGS84 Quad family \(UTM##WGS84Quad\)](#)) that can be reproduced by other standards needing to define a tile matrix set if the zone 31 is required. The other 59 zones in the 60 UTM partition can be defined similarly. Not all TileMatrix elements need to be included and

including other TileMatrices for more detailed scales is possible if they follow the same pattern.

F.6.1. UTM WGS84 Quad for zone 31 (JSON encoding)

```
{  
  "id": "UTM31WGS84Quad",  
  "title": "Universal Transverse Mercator Zone 31 WGS84 Quad",  
  "uri": "http://www.opengis.net/def/tilematrixset/OGC/1.0/UTM31WGS84Quad",  
  "crs": "http://www.opengis.net/def/crs/EPSG/0/32631",  
  "orderedAxes": ["E", "N"],  
  "tileMatrices":  
  [  
    {  
      "id": "1",  
      "scaleDenominator": 279072704.500914,  
      "cellSize": 78140.3572602559,  
      "pointOfOrigin": [-9501965.72931276, 20003931.4586255],  
      "tileWidth": 256,  
      "tileHeight": 256,  
      "matrixWidth": 1,  
      "matrixHeight": 2  
    },  
    {  
      "id": "2",  
      "scaleDenominator": 139536352.250457,  
      "cellSize": 39070.178630128,  
      "pointOfOrigin": [-9501965.72931276, 20003931.4586255],  
      "tileWidth": 256,  
      "tileHeight": 256,  
      "matrixWidth": 2,  
      "matrixHeight": 4  
    },  
    {  
      "id": "3",  
      "scaleDenominator": 69768176.1252285,  
      "cellSize": 19535.089315064,  
      "pointOfOrigin": [-9501965.72931276, 20003931.4586255],  
      "tileWidth": 256,  
      "tileHeight": 256,  
      "matrixWidth": 4,  
      "matrixHeight": 8  
    },  
    {  
      "id": "4",  
      "scaleDenominator": 34884088.0626143,  
      "cellSize": 9767.5446575319,  
      "matrixWidth": 8,  
      "matrixHeight": 16  
    }  
  ]  
}
```

```
"pointOfOrigin": [-9501965.72931276,20003931.4586255],  
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 8,  
"matrixHeight": 16  
},  
{  
    "id": "5",  
    "scaleDenominator": 17442044.0313071,  
    "cellSize": 4883.772328766,  
    "pointOfOrigin": [-9501965.72931276,20003931.4586255],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 16,  
    "matrixHeight": 32  
},  
{  
    "id": "6",  
    "scaleDenominator": 8721022.01565356,  
    "cellSize": 2441.886164383,  
    "pointOfOrigin": [-9501965.72931276,20003931.4586255],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 32,  
    "matrixHeight": 64  
},  
{  
    "id": "7",  
    "scaleDenominator": 4360511.00782678,  
    "cellSize": 1220.9430821915,  
    "pointOfOrigin": [-9501965.72931276,20003931.4586255],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 64,  
    "matrixHeight": 128  
},  
{  
    "id": "8",  
    "scaleDenominator": 2180255.50391339,  
    "cellSize": 610.471541095749,  
    "pointOfOrigin": [-9501965.72931276,20003931.4586255],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 128,  
    "matrixHeight": 256  
},  
{
```

```
"id": "9",
"scaleDenominator": 1090127.7519567,
"cellSize": 305.235770547875,
"pointOfOrigin": [-9501965.72931276,20003931.4586255],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 256,
"matrixHeight": 512
},
{
"id": "10",
"scaleDenominator": 545063.875978348,
"cellSize": 152.617885273937,
"pointOfOrigin": [-9501965.72931276,20003931.4586255],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 512,
"matrixHeight": 1024
},
{
"id": "11",
"scaleDenominator": 272531.937989174,
"cellSize": 76.3089426369687,
"pointOfOrigin": [-9501965.72931276,20003931.4586255],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 1024,
"matrixHeight": 2048
},
{
"id": "12",
"scaleDenominator": 136265.968994587,
"cellSize": 38.1544713184843,
"pointOfOrigin": [-9501965.72931276,20003931.4586255],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 2048,
"matrixHeight": 4096
},
{
"id": "13",
"scaleDenominator": 68132.9844972935,
"cellSize": 19.0772356592422,
"pointOfOrigin": [-9501965.72931276,20003931.4586255],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 4096,
```

```
    "matrixHeight": 8192
},
{
  "id": "14",
  "scaleDenominator": 34066.4922486467,
  "cellSize": 9.53861782962109,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8192,
  "matrixHeight": 16384
},
{
  "id": "15",
  "scaleDenominator": 17033.2461243234,
  "cellSize": 4.76930891481054,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16384,
  "matrixHeight": 32768
},
{
  "id": "16",
  "scaleDenominator": 8516.62306216168,
  "cellSize": 2.38465445740527,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 32768,
  "matrixHeight": 65536
},
{
  "id": "17",
  "scaleDenominator": 4258.31153108084,
  "cellSize": 1.19232722870264,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 65536,
  "matrixHeight": 131072
},
{
  "id": "18",
  "scaleDenominator": 2129.15576554042,
  "cellSize": 0.596163614351318,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
```

```
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 131072,
    "matrixHeight": 262144
},
{
  "id": "19",
  "scaleDenominator": 1064.57788277021,
  "cellSize": 0.298081807175659,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 262144,
  "matrixHeight": 524288
},
{
  "id": "20",
  "scaleDenominator": 532.288941385105,
  "cellSize": 0.149040903587829,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 524288,
  "matrixHeight": 1048576
},
{
  "id": "21",
  "scaleDenominator": 266.144470692553,
  "cellSize": 0.0745204517939147,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 1048576,
  "matrixHeight": 2097152
},
{
  "id": "22",
  "scaleDenominator": 133.072235346276,
  "cellSize": 0.0372602258969574,
  "pointOfOrigin": [-9501965.72931276,20003931.4586255],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 2097152,
  "matrixHeight": 4194304
},
{
  "id": "23",

```

```

    "scaleDenominator": 66.5361176731382,
    "cellSize": 0.0186301129484787,
    "pointOfOrigin": [-9501965.72931276,20003931.4586255],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 4194304,
    "matrixHeight": 8388608
  },
  {
    "id": "24",
    "scaleDenominator": 33.2680588365691,
    "cellSize": 0.00931505647423934,
    "pointOfOrigin": [-9501965.72931276,20003931.4586255],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 8388608,
    "matrixHeight": 16777216
  }
]
}

```

F.6.2. UTM WGS84 Quad for zone 31 (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="UTM31WGS84Quad" xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
  <tmsc:Title>Universal Transverse Mercator Zone 31 WGS84 Quad</tmsc:Title>
  <tmsc:Identifier>UTM31WGS84Quad</tmsc:Identifier>
  <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/UTM31WGS84Quad</uri>
  <tmsc:CRS>
    <tmsc:URI>http://www.opengis.net/def/crs/EPSG/0/32631</tmsc:URI>
  </tmsc:CRS>
  <OrderedAxes>E,N</OrderedAxes>
  <TileMatrix>
    <tmsc:Identifier>1</tmsc:Identifier>
    <ScaleDenominator>279072704.500914</ScaleDenominator>
    <CellSize>78140.3572602559</CellSize>
    <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1</MatrixWidth>
    <MatrixHeight>2</MatrixHeight>
  </TileMatrix>

```

```
<TileMatrix>
  <tmsc:Identifier>2</tmsc:Identifier>
  <ScaleDenominator>139536352.250457</ScaleDenominator>
  <CellSize>39070.178630128</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2</MatrixWidth>
  <MatrixHeight>4</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>3</tmsc:Identifier>
  <ScaleDenominator>69768176.1252285</ScaleDenominator>
  <CellSize>19535.089315064</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4</MatrixWidth>
  <MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>4</tmsc:Identifier>
  <ScaleDenominator>34884088.0626143</ScaleDenominator>
  <CellSize>9767.5446575319</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8</MatrixWidth>
  <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>5</tmsc:Identifier>
  <ScaleDenominator>17442044.0313071</ScaleDenominator>
  <CellSize>4883.772328766</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16</MatrixWidth>
  <MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>6</tmsc:Identifier>
  <ScaleDenominator>8721022.01565356</ScaleDenominator>
  <CellSize>2441.886164383</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
```

```
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>7</tmsc:Identifier>
<ScaleDenominator>4360511.00782678</ScaleDenominator>
<CellSize>1220.9430821915</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>8</tmsc:Identifier>
<ScaleDenominator>2180255.50391339</ScaleDenominator>
<CellSize>610.471541095749</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>128</MatrixWidth>
<MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>9</tmsc:Identifier>
<ScaleDenominator>1090127.7519567</ScaleDenominator>
<CellSize>305.235770547875</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>256</MatrixWidth>
<MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>10</tmsc:Identifier>
<ScaleDenominator>545063.875978348</ScaleDenominator>
<CellSize>152.617885273937</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>512</MatrixWidth>
<MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>11</tmsc:Identifier>
<ScaleDenominator>272531.937989174</ScaleDenominator>
<CellSize>76.3089426369687</CellSize>
```

```
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>1024</MatrixWidth>
<MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>12</tmsc:Identifier>
    <ScaleDenominator>136265.968994587</ScaleDenominator>
    <CellSize>38.1544713184843</CellSize>
    <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2048</MatrixWidth>
    <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>13</tmsc:Identifier>
    <ScaleDenominator>68132.9844972935</ScaleDenominator>
    <CellSize>19.0772356592422</CellSize>
    <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>4096</MatrixWidth>
    <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>14</tmsc:Identifier>
    <ScaleDenominator>34066.4922486467</ScaleDenominator>
    <CellSize>9.53861782962109</CellSize>
    <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>8192</MatrixWidth>
    <MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>15</tmsc:Identifier>
    <ScaleDenominator>17033.2461243234</ScaleDenominator>
    <CellSize>4.76930891481054</CellSize>
    <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>16384</MatrixWidth>
    <MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
```

```
<tmsc:Identifier>16</tmsc:Identifier>
<ScaleDenominator>8516.62306216168</ScaleDenominator>
<CellSize>2.38465445740527</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32768</MatrixWidth>
<MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>17</tmsc:Identifier>
<ScaleDenominator>4258.31153108084</ScaleDenominator>
<CellSize>1.19232722870264</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>65536</MatrixWidth>
<MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>18</tmsc:Identifier>
<ScaleDenominator>2129.15576554042</ScaleDenominator>
<CellSize>0.596163614351318</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>131072</MatrixWidth>
<MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>19</tmsc:Identifier>
<ScaleDenominator>1064.57788277021</ScaleDenominator>
<CellSize>0.298081807175659</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>262144</MatrixWidth>
<MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>20</tmsc:Identifier>
<ScaleDenominator>532.288941385105</ScaleDenominator>
<CellSize>0.149040903587829</CellSize>
<PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>524288</MatrixWidth>
```

```
<MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>21</tmsc:Identifier>
  <ScaleDenominator>266.144470692553</ScaleDenominator>
  <CellSize>0.0745204517939147</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>1048576</MatrixWidth>
  <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>22</tmsc:Identifier>
  <ScaleDenominator>133.072235346276</ScaleDenominator>
  <CellSize>0.0372602258969574</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2097152</MatrixWidth>
  <MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>23</tmsc:Identifier>
  <ScaleDenominator>66.5361176731382</ScaleDenominator>
  <CellSize>0.0186301129484787</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4194304</MatrixWidth>
  <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>24</tmsc:Identifier>
  <ScaleDenominator>33.2680588365691</ScaleDenominator>
  <CellSize>0.00931505647423934</CellSize>
  <PointOfOrigin>-9501965.72931276 20003931.4586255</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8388608</MatrixWidth>
  <MatrixHeight>16777216</MatrixHeight>
</TileMatrix>
</TileMatrixSet>
```

F.7. Arctic Universal Polar Stereographic WGS 84 Quad

These are the JSON and XML definition of the UPSArcticWGS84Quad tile matrix set (see [Arctic Universal Polar Stereographic WGS 84 Quad \(UPSArcticWGS84Quad\)](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

F.7.1. Arctic UPS WGS 84 Quad (JSON encoding)

```
{
  "id": "UPSArcticWGS84Quad",
  "title": "Universal Polar Stereographic WGS 84 Quad for Arctic",
  "uri": "http://www.opengis.net/def/tilematrixset/OGC/1.0/UPSArcticWGS84Quad",
  "crs": "http://www.opengis.net/def/crs/EPSG/0/5041",
  "orderedAxes": ["E", "N"],
  "tileMatrices": [
    {
      "id": "0",
      "scaleDenominator": 458726544.4,
      "cellSize": 128443.4324,
      "pointOfOrigin": [-14440759.350252, 18440759.350252],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 1,
      "matrixHeight": 1
    },
    {
      "id": "1",
      "scaleDenominator": 229363272.2,
      "cellSize": 64221.71621,
      "pointOfOrigin": [-14440759.350252, 18440759.350252],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 2,
      "matrixHeight": 2
    },
    {
      "id": "2",
      "scaleDenominator": 114681636.1,
      "cellSize": 32110.85811,
      "pointOfOrigin": [-14440759.350252, 18440759.350252],
      "tileWidth": 256,
      "tileHeight": 256,
```

```
    "matrixWidth": 4,
    "matrixHeight": 4
},
{
  "id": "3",
  "scaleDenominator": 57340818.05,
  "cellSize": 16055.42905,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8,
  "matrixHeight": 8
},
{
  "id": "4",
  "scaleDenominator": 28670409.02,
  "cellSize": 8027.714526,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16,
  "matrixHeight": 16
},
{
  "id": "5",
  "scaleDenominator": 14335204.51,
  "cellSize": 4013.857263,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 32,
  "matrixHeight": 32
},
{
  "id": "6",
  "scaleDenominator": 7167602.256,
  "cellSize": 2006.928632,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 64,
  "matrixHeight": 64
},
{
  "id": "7",
  "scaleDenominator": 3583801.128,
  "cellSize": 1003.464316,
```

```
"pointOfOrigin": [-14440759.350252,18440759.350252],  
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 128,  
"matrixHeight": 128  
},  
{  
    "id": "8",  
    "scaleDenominator": 1791900.564,  
    "cellSize": 501.7321579,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 256,  
    "matrixHeight": 256  
},  
{  
    "id": "9",  
    "scaleDenominator": 895950.282,  
    "cellSize": 250.866079,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 512,  
    "matrixHeight": 512  
},  
{  
    "id": "10",  
    "scaleDenominator": 447975.141,  
    "cellSize": 125.4330395,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 1024,  
    "matrixHeight": 1024  
},  
{  
    "id": "11",  
    "scaleDenominator": 223987.5705,  
    "cellSize": 62.71651974,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 2048,  
    "matrixHeight": 2048  
},  
{
```

```
"id": "12",
"scaleDenominator": 111993.7852,
"cellSize": 31.35825987,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 4096,
"matrixHeight": 4096
},
{
"id": "13",
"scaleDenominator": 55996.89262,
"cellSize": 15.67912993,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 8192,
"matrixHeight": 8192
},
{
"id": "14",
"scaleDenominator": 27998.44631,
"cellSize": 7.839564967,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 16384,
"matrixHeight": 16384
},
{
"id": "15",
"scaleDenominator": 13999.22316,
"cellSize": 3.919782484,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 32768,
"matrixHeight": 32768
},
{
"id": "16",
"scaleDenominator": 6999.611578,
"cellSize": 1.959891242,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 65536,
```

```
    "matrixHeight": 65536
},
{
  "id": "17",
  "scaleDenominator": 3499.805789,
  "cellSize": 0.979945621,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 131072,
  "matrixHeight": 131072
},
{
  "id": "18",
  "scaleDenominator": 1749.902894,
  "cellSize": 0.48997281,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 262144,
  "matrixHeight": 262144
},
{
  "id": "19",
  "scaleDenominator": 874.9514472,
  "cellSize": 0.244986405,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 524288,
  "matrixHeight": 524288
},
{
  "id": "20",
  "scaleDenominator": 437.4757236,
  "cellSize": 0.122493203,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 1048576,
  "matrixHeight": 1048576
},
{
  "id": "21",
  "scaleDenominator": 218.7378618,
  "cellSize": 0.061246601,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
```

```

    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 2097152,
    "matrixHeight": 2097152
},
{
  "id": "22",
  "scaleDenominator": 109.3689309,
  "cellSize": 0.030623301,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4194304,
  "matrixHeight": 4194304
},
{
  "id": "23",
  "scaleDenominator": 54.68446545,
  "cellSize": 0.01531165,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8388608,
  "matrixHeight": 8388608
},
{
  "id": "24",
  "scaleDenominator": 27.34223273,
  "cellSize": 0.007655825,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16777216,
  "matrixHeight": 16777216
}
]
}

```

F.7.2. Arctic UPS WGS 84 Quad (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="UPSArcticWGS84Quad" xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">

```

```
<tmsc:Title>Universal Polar Stereographic WGS 84 Quad for  
Arctic</tmsc:Title>  
<tmsc:Identifier>UPSArcticWGS84Quad</tmsc:Identifier>  
  
<uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/UPSArcticWGS84Quad</uri>  
<tmsc:CRS>  
  <tmsc:URI>http://www.opengis.net/def/crs/EPSG/0/5041</tmsc:URI>  
</tmsc:CRS>  
<OrderedAxes>E,N</OrderedAxes>  
<TileMatrix>  
  <tmsc:Identifier>0</tmsc:Identifier>  
  <ScaleDenominator>458726544.4</ScaleDenominator>  
  <CellSize>128443.4324</CellSize>  
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>  
  <TileWidth>256</TileWidth>  
  <TileHeight>256</TileHeight>  
  <MatrixWidth>1</MatrixWidth>  
  <MatrixHeight>1</MatrixHeight>  
</TileMatrix>  
<TileMatrix>  
  <tmsc:Identifier>1</tmsc:Identifier>  
  <ScaleDenominator>229363272.2</ScaleDenominator>  
  <CellSize>64221.71621</CellSize>  
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>  
  <TileWidth>256</TileWidth>  
  <TileHeight>256</TileHeight>  
  <MatrixWidth>2</MatrixWidth>  
  <MatrixHeight>2</MatrixHeight>  
</TileMatrix>  
<TileMatrix>  
  <tmsc:Identifier>2</tmsc:Identifier>  
  <ScaleDenominator>114681636.1</ScaleDenominator>  
  <CellSize>32110.85811</CellSize>  
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>  
  <TileWidth>256</TileWidth>  
  <TileHeight>256</TileHeight>  
  <MatrixWidth>4</MatrixWidth>  
  <MatrixHeight>4</MatrixHeight>  
</TileMatrix>  
<TileMatrix>  
  <tmsc:Identifier>3</tmsc:Identifier>  
  <ScaleDenominator>57340818.05</ScaleDenominator>  
  <CellSize>16055.42905</CellSize>  
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>  
  <TileWidth>256</TileWidth>  
  <TileHeight>256</TileHeight>
```

```
<MatrixWidth>8</MatrixWidth>
<MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>4</tmsc:Identifier>
<ScaleDenominator>28670409.02</ScaleDenominator>
<CellSize>8027.714526</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16</MatrixWidth>
<MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>5</tmsc:Identifier>
<ScaleDenominator>14335204.51</ScaleDenominator>
<CellSize>4013.857263</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>6</tmsc:Identifier>
<ScaleDenominator>7167602.256</ScaleDenominator>
<CellSize>2006.928632</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>7</tmsc:Identifier>
<ScaleDenominator>3583801.128</ScaleDenominator>
<CellSize>1003.464316</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>128</MatrixWidth>
<MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>8</tmsc:Identifier>
<ScaleDenominator>1791900.564</ScaleDenominator>
<CellSize>501.7321579</CellSize>
```

```
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>256</MatrixWidth>
<MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>9</tmsc:Identifier>
    <ScaleDenominator>895950.282</ScaleDenominator>
    <CellSize>250.866079</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>512</MatrixWidth>
    <MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>10</tmsc:Identifier>
    <ScaleDenominator>447975.141</ScaleDenominator>
    <CellSize>125.4330395</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1024</MatrixWidth>
    <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>11</tmsc:Identifier>
    <ScaleDenominator>223987.5705</ScaleDenominator>
    <CellSize>62.71651974</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2048</MatrixWidth>
    <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>12</tmsc:Identifier>
    <ScaleDenominator>111993.7852</ScaleDenominator>
    <CellSize>31.35825987</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>4096</MatrixWidth>
    <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
```

```
<tmsc:Identifier>13</tmsc:Identifier>
<ScaleDenominator>55996.89262</ScaleDenominator>
<CellSize>15.67912993</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8192</MatrixWidth>
<MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>14</tmsc:Identifier>
<ScaleDenominator>27998.44631</ScaleDenominator>
<CellSize>7.839564967</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16384</MatrixWidth>
<MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>15</tmsc:Identifier>
<ScaleDenominator>13999.22316</ScaleDenominator>
<CellSize>3.919782484</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32768</MatrixWidth>
<MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>16</tmsc:Identifier>
<ScaleDenominator>6999.611578</ScaleDenominator>
<CellSize>1.959891242</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>65536</MatrixWidth>
<MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>17</tmsc:Identifier>
<ScaleDenominator>3499.805789</ScaleDenominator>
<CellSize>0.979945621</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>131072</MatrixWidth>
```

```
<MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>18</tmsc:Identifier>
  <ScaleDenominator>1749.902894</ScaleDenominator>
  <CellSize>0.48997281</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>262144</MatrixWidth>
  <MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>19</tmsc:Identifier>
  <ScaleDenominator>874.9514472</ScaleDenominator>
  <CellSize>0.244986405</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>524288</MatrixWidth>
  <MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>20</tmsc:Identifier>
  <ScaleDenominator>437.4757236</ScaleDenominator>
  <CellSize>0.122493203</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>1048576</MatrixWidth>
  <MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>21</tmsc:Identifier>
  <ScaleDenominator>218.7378618</ScaleDenominator>
  <CellSize>0.061246601</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2097152</MatrixWidth>
  <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>22</tmsc:Identifier>
  <ScaleDenominator>109.3689309</ScaleDenominator>
  <CellSize>0.030623301</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
```

```

<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4194304</MatrixWidth>
<MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>23</tmsc:Identifier>
  <ScaleDenominator>54.68446545</ScaleDenominator>
  <CellSize>0.01531165</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8388608</MatrixWidth>
  <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>24</tmsc:Identifier>
  <ScaleDenominator>27.34223273</ScaleDenominator>
  <CellSize>0.007655825</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16777216</MatrixWidth>
  <MatrixHeight>16777216</MatrixHeight>
</TileMatrix>
</TileMatrixSet>

```

F.8. Antarctic Universal Polar Stereographic WGS84 Quad

These are the JSON and XML definitions of the UPSAntarcticWGS84Quad tile matrix set (see [Antarctic Universal Polar Stereographic WGS84 Quad \(UPSAntarcticWGS84Quad\)](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

F.8.1. Antarctic UPS WGS84 Quad (JSON encoding)

```
{
  "id": "UPSAntarcticWGS84Quad",
  "title": "Universal Polar Stereographic WGS 84 Quad for Antarctic",
  "uri":
  "http://www.opengis.net/def/tilematrixset/OGC/1.0/UPSAntarcticWGS84Quad",
  "crs": "http://www.opengis.net/def/crs/EPSG/0/5042",
```

```
"orderedAxes": ["E", "N"],
"tileMatrices":
[
  {
    "id": "0",
    "scaleDenominator": 458726544.4,
    "cellSize": 128443.4324,
    "pointOfOrigin": [-14440759.350252,18440759.350252],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 1,
    "matrixHeight": 1
  },
  {
    "id": "1",
    "scaleDenominator": 229363272.2,
    "cellSize": 64221.71621,
    "pointOfOrigin": [-14440759.350252,18440759.350252],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 2,
    "matrixHeight": 2
  },
  {
    "id": "2",
    "scaleDenominator": 114681636.1,
    "cellSize": 32110.85811,
    "pointOfOrigin": [-14440759.350252,18440759.350252],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 4,
    "matrixHeight": 4
  },
  {
    "id": "3",
    "scaleDenominator": 57340818.05,
    "cellSize": 16055.42905,
    "pointOfOrigin": [-14440759.350252,18440759.350252],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 8,
    "matrixHeight": 8
  },
  {
    "id": "4",
    "scaleDenominator": 28670409.02,
    "cellSize": 8027.714526,
    "matrixWidth": 16,
    "matrixHeight": 16
  }
]
```

```
"pointOfOrigin": [-14440759.350252,18440759.350252],  
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 16,  
"matrixHeight": 16  
},  
{  
    "id": "5",  
    "scaleDenominator": 14335204.51,  
    "cellSize": 4013.857263,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 32,  
    "matrixHeight": 32  
},  
{  
    "id": "6",  
    "scaleDenominator": 7167602.256,  
    "cellSize": 2006.928632,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 64,  
    "matrixHeight": 64  
},  
{  
    "id": "7",  
    "scaleDenominator": 3583801.128,  
    "cellSize": 1003.464316,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 128,  
    "matrixHeight": 128  
},  
{  
    "id": "8",  
    "scaleDenominator": 1791900.564,  
    "cellSize": 501.7321579,  
    "pointOfOrigin": [-14440759.350252,18440759.350252],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 256,  
    "matrixHeight": 256  
},  
{
```

```
"id": "9",
"scaleDenominator": 895950.282,
"cellSize": 250.866079,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 512,
"matrixHeight": 512
},
{
"id": "10",
"scaleDenominator": 447975.141,
"cellSize": 125.4330395,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 1024,
"matrixHeight": 1024
},
{
"id": "11",
"scaleDenominator": 223987.5705,
"cellSize": 62.71651974,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 2048,
"matrixHeight": 2048
},
{
"id": "12",
"scaleDenominator": 111993.7852,
"cellSize": 31.35825987,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 4096,
"matrixHeight": 4096
},
{
"id": "13",
"scaleDenominator": 55996.89262,
"cellSize": 15.67912993,
"pointOfOrigin": [-14440759.350252,18440759.350252],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 8192,
```

```
    "matrixHeight": 8192
},
{
  "id": "14",
  "scaleDenominator": 27998.44631,
  "cellSize": 7.839564967,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16384,
  "matrixHeight": 16384
},
{
  "id": "15",
  "scaleDenominator": 13999.22316,
  "cellSize": 3.919782484,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 32768,
  "matrixHeight": 32768
},
{
  "id": "16",
  "scaleDenominator": 6999.611578,
  "cellSize": 1.959891242,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 65536,
  "matrixHeight": 65536
},
{
  "id": "17",
  "scaleDenominator": 3499.805789,
  "cellSize": 0.979945621,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 131072,
  "matrixHeight": 131072
},
{
  "id": "18",
  "scaleDenominator": 1749.902894,
  "cellSize": 0.48997281,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
```

```
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 262144,
    "matrixHeight": 262144
},
{
  "id": "19",
  "scaleDenominator": 874.9514472,
  "cellSize": 0.244986405,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 524288,
  "matrixHeight": 524288
},
{
  "id": "20",
  "scaleDenominator": 437.4757236,
  "cellSize": 0.122493203,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 1048576,
  "matrixHeight": 1048576
},
{
  "id": "21",
  "scaleDenominator": 218.7378618,
  "cellSize": 0.061246601,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 2097152,
  "matrixHeight": 2097152
},
{
  "id": "22",
  "scaleDenominator": 109.3689309,
  "cellSize": 0.030623301,
  "pointOfOrigin": [-14440759.350252,18440759.350252],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4194304,
  "matrixHeight": 4194304
},
{
  "id": "23",
```

```

    "scaleDenominator": 54.68446545,
    "cellSize": 0.01531165,
    "pointOfOrigin": [-14440759.350252,18440759.350252],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 8388608,
    "matrixHeight": 8388608
  },
  {
    "id": "24",
    "scaleDenominator": 27.34223273,
    "cellSize": 0.007655825,
    "pointOfOrigin": [-14440759.350252,18440759.350252],
    "tileWidth": 256,
    "tileHeight": 256,
    "matrixWidth": 16777216,
    "matrixHeight": 16777216
  }
]
}

```

F.8.2. Antarctic UPS WGS84 Quad (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="UPSAntarcticWGS84Quad"
 xmlns="http://www.opengis.net/tms/2.0"
 xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
  <tmsc:Title>Universal Polar Stereographic WGS 84 Quad for
Antarctic</tmsc:Title>
  <tmsc:Identifier>UPSAntarcticWGS84Quad</tmsc:Identifier>

  <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/UPSAntarcticWGS84Quad</
uri>
  <tmsc:CRS>
    <tmsc:URI>http://www.opengis.net/def/crs/EPSG/0/5042</tmsc:URI>
  </tmsc:CRS>
  <OrderedAxes>E,N</OrderedAxes>
  <TileMatrix>
    <tmsc:Identifier>0</tmsc:Identifier>
    <ScaleDenominator>458726544.4</ScaleDenominator>
    <CellSize>128443.4324</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
  
```

```
<TileHeight>256</TileHeight>
<MatrixWidth>1</MatrixWidth>
<MatrixHeight>1</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>1</tmsc:Identifier>
  <ScaleDenominator>229363272.2</ScaleDenominator>
  <CellSize>64221.71621</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2</MatrixWidth>
  <MatrixHeight>2</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>2</tmsc:Identifier>
  <ScaleDenominator>114681636.1</ScaleDenominator>
  <CellSize>32110.85811</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4</MatrixWidth>
  <MatrixHeight>4</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>3</tmsc:Identifier>
  <ScaleDenominator>57340818.05</ScaleDenominator>
  <CellSize>16055.42905</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8</MatrixWidth>
  <MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>4</tmsc:Identifier>
  <ScaleDenominator>28670409.02</ScaleDenominator>
  <CellSize>8027.714526</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>16</MatrixWidth>
  <MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>5</tmsc:Identifier>
  <ScaleDenominator>14335204.51</ScaleDenominator>
```

```
<CellSize>4013.857263</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>6</tmsc:Identifier>
<ScaleDenominator>7167602.256</ScaleDenominator>
<CellSize>2006.928632</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>7</tmsc:Identifier>
<ScaleDenominator>3583801.128</ScaleDenominator>
<CellSize>1003.464316</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>128</MatrixWidth>
<MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>8</tmsc:Identifier>
<ScaleDenominator>1791900.564</ScaleDenominator>
<CellSize>501.7321579</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>256</MatrixWidth>
<MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>9</tmsc:Identifier>
<ScaleDenominator>895950.282</ScaleDenominator>
<CellSize>250.866079</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>512</MatrixWidth>
<MatrixHeight>512</MatrixHeight>
</TileMatrix>
```

```
<TileMatrix>
  <tmsc:Identifier>10</tmsc:Identifier>
  <ScaleDenominator>447975.141</ScaleDenominator>
  <CellSize>125.4330395</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>1024</MatrixWidth>
  <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>11</tmsc:Identifier>
  <ScaleDenominator>223987.5705</ScaleDenominator>
  <CellSize>62.71651974</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>2048</MatrixWidth>
  <MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>12</tmsc:Identifier>
  <ScaleDenominator>111993.7852</ScaleDenominator>
  <CellSize>31.35825987</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>4096</MatrixWidth>
  <MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>13</tmsc:Identifier>
  <ScaleDenominator>55996.89262</ScaleDenominator>
  <CellSize>15.67912993</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
  <MatrixWidth>8192</MatrixWidth>
  <MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>14</tmsc:Identifier>
  <ScaleDenominator>27998.44631</ScaleDenominator>
  <CellSize>7.839564967</CellSize>
  <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
  <TileWidth>256</TileWidth>
  <TileHeight>256</TileHeight>
```

```
<MatrixWidth>16384</MatrixWidth>
<MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>15</tmsc:Identifier>
<ScaleDenominator>13999.22316</ScaleDenominator>
<CellSize>3.919782484</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32768</MatrixWidth>
<MatrixHeight>32768</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>16</tmsc:Identifier>
<ScaleDenominator>6999.611578</ScaleDenominator>
<CellSize>1.959891242</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>65536</MatrixWidth>
<MatrixHeight>65536</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>17</tmsc:Identifier>
<ScaleDenominator>3499.805789</ScaleDenominator>
<CellSize>0.979945621</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>131072</MatrixWidth>
<MatrixHeight>131072</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>18</tmsc:Identifier>
<ScaleDenominator>1749.902894</ScaleDenominator>
<CellSize>0.48997281</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>262144</MatrixWidth>
<MatrixHeight>262144</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>19</tmsc:Identifier>
<ScaleDenominator>874.9514472</ScaleDenominator>
<CellSize>0.244986405</CellSize>
```

```
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>524288</MatrixWidth>
<MatrixHeight>524288</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>20</tmsc:Identifier>
    <ScaleDenominator>437.4757236</ScaleDenominator>
    <CellSize>0.122493203</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1048576</MatrixWidth>
    <MatrixHeight>1048576</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>21</tmsc:Identifier>
    <ScaleDenominator>218.7378618</ScaleDenominator>
    <CellSize>0.061246601</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2097152</MatrixWidth>
    <MatrixHeight>2097152</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>22</tmsc:Identifier>
    <ScaleDenominator>109.3689309</ScaleDenominator>
    <CellSize>0.030623301</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>4194304</MatrixWidth>
    <MatrixHeight>4194304</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>23</tmsc:Identifier>
    <ScaleDenominator>54.68446545</ScaleDenominator>
    <CellSize>0.01531165</CellSize>
    <PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>8388608</MatrixWidth>
    <MatrixHeight>8388608</MatrixHeight>
</TileMatrix>
<TileMatrix>
```

```

<tmsc:Identifier>24</tmsc:Identifier>
<ScaleDenominator>27.34223273</ScaleDenominator>
<CellSize>0.007655825</CellSize>
<PointOfOrigin>-14440759.350252 18440759.350252</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16777216</MatrixWidth>
<MatrixHeight>16777216</MatrixHeight>
</TileMatrix>
</TileMatrixSet>

```

F.9. European ETRS89 Lambert Azimuthal Equal-Area Quad

These are the JSON and XML definitions of the EuropeanETRS89_LAEAQuad tile matrix set (see [European ETRS89 Lambert azimuthal equal-area Quad \(EuropeanETRS89_LAEAQuad\)](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

F.9.1. European ETRS89 Lambert azimuthal equal-area Quad (JSON encoding)

```
{
  "id": "EuropeanETRS89_LAEAQuad",
  "title": "Lambert Azimuthal Equal Area ETRS89 for Europe",
  "uri":
    "http://www.opengis.net/def/tilematrixset/OGC/1.0/EuropeanETRS89_LAEAQuad",
  "crs": "http://www.opengis.net/def/crs/EPSG/0/3035",
  "orderedAxes": [ "Y", "X" ],
  "tileMatrices":
    [
      {
        "id": "0",
        "scaleDenominator": 62779017.8571428,
        "cellSize": 17578.125,
        "pointOfOrigin": [ 5500000.0, 2000000.0 ],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 1,
        "matrixHeight": 1
      },
      {
        "id": "1",

```

```
"scaleDenominator": 31389508.9285714,
"cellSize": 8789.0625,
"pointOfOrigin": [5500000.0, 2000000.0],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 2,
"matrixHeight": 2
},
{
"id": "2",
"scaleDenominator": 15694754.4642857,
"cellSize": 4394.53125,
"pointOfOrigin": [5500000.0, 2000000.0],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 4,
"matrixHeight": 4
},
{
"id": "3",
"scaleDenominator": 7847377.23214285,
"cellSize": 2197.265625,
"pointOfOrigin": [5500000.0, 2000000.0],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 8,
"matrixHeight": 8
},
{
"id": "4",
"scaleDenominator": 3923688.61607142,
"cellSize": 1098.6328125,
"pointOfOrigin": [5500000.0, 2000000.0],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 16,
"matrixHeight": 16
},
{
"id": "5",
"scaleDenominator": 1961844.30803571,
"cellSize": 549.31640625,
"pointOfOrigin": [5500000.0, 2000000.0],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 32,
"matrixHeight": 32
```

```
},
{
  "id": "6",
  "scaleDenominator": 980922.154017857,
  "cellSize": 274.658203125,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 64,
  "matrixHeight": 64
},
{
  "id": "7",
  "scaleDenominator": 490461.077008928,
  "cellSize": 137.3291015625,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 128,
  "matrixHeight": 128
},
{
  "id": "8",
  "scaleDenominator": 245230.538504464,
  "cellSize": 68.6645507812,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 256,
  "matrixHeight": 256
},
{
  "id": "9",
  "scaleDenominator": 122615.269252232,
  "cellSize": 34.3322753906,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 512,
  "matrixHeight": 512
},
{
  "id": "10",
  "scaleDenominator": 61307.634626116,
  "cellSize": 17.1661376953,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
```

```
"tileHeight": 256,
"matrixWidth": 1024,
"matrixHeight": 1024
},
{
  "id": "11",
  "scaleDenominator": 30653.817313058,
  "cellSize": 8.5830688477,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 2048,
  "matrixHeight": 2048
},
{
  "id": "12",
  "scaleDenominator": 15326.908656529,
  "cellSize": 4.2915344238,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 4096,
  "matrixHeight": 4096
},
{
  "id": "13",
  "scaleDenominator": 7663.45432826451,
  "cellSize": 2.1457672119,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 8192,
  "matrixHeight": 8192
},
{
  "id": "14",
  "scaleDenominator": 3831.72716413225,
  "cellSize": 1.072883606,
  "pointOfOrigin": [5500000.0, 2000000.0],
  "tileWidth": 256,
  "tileHeight": 256,
  "matrixWidth": 16384,
  "matrixHeight": 16384
},
{
  "id": "15",
  "scaleDenominator": 1915.86358206612,
```

```

        "cellSize": 0.536441803,
        "pointOfOrigin": [5500000.0, 2000000.0],
        "tileWidth": 256,
        "tileHeight": 256,
        "matrixWidth": 32768,
        "matrixHeight": 32768
    }
]
}

```

F.9.2. European ETRS89 Lambert Azimuthal Equal-Area Quad (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="EuropeanETRS89_LAEAQuad"
 xmlns="http://www.opengis.net/tms/2.0"
 xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
 xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
    <tmsc:Title>Lambert Azimuthal Equal Area ETRS89 for Europe</tmsc:Title>
    <tmsc:Identifier>EuropeanETRS89_LAEAQuad</tmsc:Identifier>

    <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/EuropeanETRS89_LAEAQuad
    </uri>
    <tmsc:CRS>
        <tmsc:URI>http://www.opengis.net/def/crs/EPSG/0/3035</tmsc:URI>
    </tmsc:CRS>
    <OrderedAxes>Y,X</OrderedAxes>
    <TileMatrix>
        <tmsc:Identifier>0</tmsc:Identifier>
        <ScaleDenominator>62779017.8571428</ScaleDenominator>
        <CellSize>17578.125</CellSize>
        <PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
        <MatrixWidth>1</MatrixWidth>
        <MatrixHeight>1</MatrixHeight>
    </TileMatrix>
    <TileMatrix>
        <tmsc:Identifier>1</tmsc:Identifier>
        <ScaleDenominator>31389508.9285714</ScaleDenominator>
        <CellSize>8789.0625</CellSize>
        <PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
        <TileWidth>256</TileWidth>
        <TileHeight>256</TileHeight>
    
```

```
<MatrixWidth>2</MatrixWidth>
<MatrixHeight>2</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>2</tmsc:Identifier>
<ScaleDenominator>15694754.4642857</ScaleDenominator>
<CellSize>4394.53125</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4</MatrixWidth>
<MatrixHeight>4</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>3</tmsc:Identifier>
<ScaleDenominator>7847377.23214285</ScaleDenominator>
<CellSize>2197.265625</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8</MatrixWidth>
<MatrixHeight>8</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>4</tmsc:Identifier>
<ScaleDenominator>3923688.61607142</ScaleDenominator>
<CellSize>1098.6328125</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16</MatrixWidth>
<MatrixHeight>16</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>5</tmsc:Identifier>
<ScaleDenominator>1961844.30803571</ScaleDenominator>
<CellSize>549.31640625</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>32</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>6</tmsc:Identifier>
<ScaleDenominator>980922.154017857</ScaleDenominator>
<CellSize>274.658203125</CellSize>
```

```
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>64</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>7</tmsc:Identifier>
    <ScaleDenominator>490461.077008928</ScaleDenominator>
    <CellSize>137.3291015625</CellSize>
    <PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>128</MatrixWidth>
    <MatrixHeight>128</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>8</tmsc:Identifier>
    <ScaleDenominator>245230.538504464</ScaleDenominator>
    <CellSize>68.6645507812</CellSize>
    <PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>256</MatrixWidth>
    <MatrixHeight>256</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>9</tmsc:Identifier>
    <ScaleDenominator>122615.269252232</ScaleDenominator>
    <CellSize>34.3322753906</CellSize>
    <PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>512</MatrixWidth>
    <MatrixHeight>512</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>10</tmsc:Identifier>
    <ScaleDenominator>61307.634626116</ScaleDenominator>
    <CellSize>17.1661376953</CellSize>
    <PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1024</MatrixWidth>
    <MatrixHeight>1024</MatrixHeight>
</TileMatrix>
<TileMatrix>
```

```
<tmsc:Identifier>11</tmsc:Identifier>
<ScaleDenominator>30653.817313058</ScaleDenominator>
<CellSize>8.5830688477</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>2048</MatrixWidth>
<MatrixHeight>2048</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>12</tmsc:Identifier>
<ScaleDenominator>15326.908656529</ScaleDenominator>
<CellSize>4.2915344238</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4096</MatrixWidth>
<MatrixHeight>4096</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>13</tmsc:Identifier>
<ScaleDenominator>7663.45432826451</ScaleDenominator>
<CellSize>2.1457672119</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8192</MatrixWidth>
<MatrixHeight>8192</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>14</tmsc:Identifier>
<ScaleDenominator>3831.72716413225</ScaleDenominator>
<CellSize>1.072883606</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16384</MatrixWidth>
<MatrixHeight>16384</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>15</tmsc:Identifier>
<ScaleDenominator>1915.86358206612</ScaleDenominator>
<CellSize>0.536441803</CellSize>
<PointOfOrigin>5500000.0 2000000.0</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32768</MatrixWidth>
```

```

<MatrixHeight>32768</MatrixHeight>
</TileMatrix>
</TileMatrixSet>

```

F.10. Canadian Lambert Conformal Conic NAD83

These are the JSON and XML definitions of the CanadianNAD83_LCC tile matrix set (see [Canadian NAD83 Lambert Conformal Conic \(CanadianNAD83_LCC\)](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

F.10.1. Canadian Lambert Conformal Conic NAD83 (JSON encoding)

```

{
  "id": "CanadianNAD83_LCC",
  "title": "Lambert conformal conic NAD83 for Canada",
  "uri": "http://www.opengis.net/def/tilematrixset/OGC/1.0/CanadianNAD83_LCC",
  "crs": "http://www.opengis.net/def/crs/EPSG/0/3978",
  "orderedAxes": ["E", "N"],
  "tileMatrices": [
    {
      "id": "0",
      "scaleDenominator": 145000000,
      "cellSize": 38364.6600626534,
      "pointOfOrigin": [-34655800,39310000],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 5,
      "matrixHeight": 5
    },
    {
      "id": "1",
      "scaleDenominator": 85000000,
      "cellSize": 22489.6283125899,
      "pointOfOrigin": [-34655800,39310000],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 8,
      "matrixHeight": 8
    },
    {
      "id": "2",
      "scaleDenominator": 50000000,
      "cellSize": 11244.81256294995,
      "pointOfOrigin": [-34655800,39310000],
      "tileWidth": 256,
      "tileHeight": 256,
      "matrixWidth": 16,
      "matrixHeight": 16
    }
  ]
}

```

```
"cellSize": 13229.1931250529,  
"pointOfOrigin": [-34655800,39310000],  
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 13,  
"matrixHeight": 14  
},  
{  
    "id": "3",  
    "scaleDenominator": 30000000,  
    "cellSize": 7937.51587503175,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 21,  
    "matrixHeight": 22  
},  
{  
    "id": "4",  
    "scaleDenominator": 17500000,  
    "cellSize": 4630.21759376852,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 36,  
    "matrixHeight": 38  
},  
{  
    "id": "5",  
    "scaleDenominator": 10000000,  
    "cellSize": 2645.83862501058,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 62,  
    "matrixHeight": 66  
},  
{  
    "id": "6",  
    "scaleDenominator": 6000000,  
    "cellSize": 1587.50317500635,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 103,  
    "matrixHeight": 110  
},
```

```
{  
    "id": "7",  
    "scaleDenominator": 3500000,  
    "cellSize": 926.043518753704,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 177,  
    "matrixHeight": 188  
},  
{  
    "id": "8",  
    "scaleDenominator": 2000000,  
    "cellSize": 529.167725002116,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 309,  
    "matrixHeight": 329  
},  
{  
    "id": "9",  
    "scaleDenominator": 1200000,  
    "cellSize": 317.50063500127,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 515,  
    "matrixHeight": 548  
},  
{  
    "id": "10",  
    "scaleDenominator": 700000,  
    "cellSize": 185.20870375074,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 882,  
    "matrixHeight": 938  
},  
{  
    "id": "11",  
    "scaleDenominator": 420000,  
    "cellSize": 111.125222250444,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,
```

```
"matrixWidth": 1470,  
"matrixHeight": 1563  
},  
{  
    "id": "12",  
    "scaleDenominator": 250000,  
    "cellSize": 66.1459656252646,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 2469,  
    "matrixHeight": 2626  
},  
{  
    "id": "13",  
    "scaleDenominator": 145000,  
    "cellSize": 38.3646600626534,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 4257,  
    "matrixHeight": 4528  
},  
{  
    "id": "14",  
    "scaleDenominator": 85000,  
    "cellSize": 22.4896283125899,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 7262,  
    "matrixHeight": 7723  
},  
{  
    "id": "15",  
    "scaleDenominator": 50000,  
    "cellSize": 13.2291931250529,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 12344,  
    "matrixHeight": 13130  
},  
{  
    "id": "16",  
    "scaleDenominator": 30000,  
    "cellSize": 7.93751587503175,
```

```
"pointOfOrigin": [-34655800,39310000],  
"tileWidth": 256,  
"tileHeight": 256,  
"matrixWidth": 20574,  
"matrixHeight": 21882  
},  
{  
    "id": "17",  
    "scaleDenominator": 17500,  
    "cellSize": 4.63021759376852,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 35269,  
    "matrixHeight": 37512  
},  
{  
    "id": "18",  
    "scaleDenominator": 10000,  
    "cellSize": 2.64583862501058,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 61720,  
    "matrixHeight": 65646  
},  
{  
    "id": "19",  
    "scaleDenominator": 6000,  
    "cellSize": 1.58750317500635,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 102866,  
    "matrixHeight": 109409  
},  
{  
    "id": "20",  
    "scaleDenominator": 3500,  
    "cellSize": 0.926043518753704,  
    "pointOfOrigin": [-34655800,39310000],  
    "tileWidth": 256,  
    "tileHeight": 256,  
    "matrixWidth": 176341,  
    "matrixHeight": 187558  
},  
{
```

```
"id": "21",
"scaleDenominator": 2000,
"cellSize": 0.529167725002116,
"pointOfOrigin": [-34655800,39310000],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 308596,
"matrixHeight": 328227
},
{
"id": "22",
"scaleDenominator": 1200,
"cellSize": 0.31750063500127,
"pointOfOrigin": [-34655800,39310000],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 514327,
"matrixHeight": 547044
},
{
"id": "23",
"scaleDenominator": 700,
"cellSize": 0.18520870375074,
"pointOfOrigin": [-34655800,39310000],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 881703,
"matrixHeight": 937790
},
{
"id": "24",
"scaleDenominator": 420,
"cellSize": 0.111125222250444,
"pointOfOrigin": [-34655800,39310000],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 1469505,
"matrixHeight": 1562983
},
{
"id": "25",
"scaleDenominator": 250,
"cellSize": 0.0661459656252645,
"pointOfOrigin": [-34655800,39310000],
"tileWidth": 256,
"tileHeight": 256,
"matrixWidth": 2468768,
```

```

        "matrixHeight": 2625811
    }
]
}
```

F.10.2. Canadian Lambert Conformal Conic NAD83 (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="CanadianNAD83_LCC" xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmSc="http://www.opengis.net/tms/2.0/common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
  <tmSc:Title>Lambert Conformal Conic NAD83 for Canada</tmSc:Title>
  <tmSc:Identifier>CanadianNAD83_LCC</tmSc:Identifier>

  <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/CanadianNAD83_LCC</uri>
  <tmSc:CRS>
    <tmSc:URI>http://www.opengis.net/def/crs/EPSG/0/3978</tmSc:URI>
  </tmSc:CRS>
  <OrderedAxes>E,N</OrderedAxes>
  <TileMatrix>
    <tmSc:Identifier>0</tmSc:Identifier>
    <ScaleDenominator>145000000</ScaleDenominator>
    <CellSize>38364.6600626534</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>5</MatrixWidth>
    <MatrixHeight>5</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmSc:Identifier>1</tmSc:Identifier>
    <ScaleDenominator>85000000</ScaleDenominator>
    <CellSize>22489.6283125899</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>8</MatrixWidth>
    <MatrixHeight>8</MatrixHeight>
  </TileMatrix>
  <TileMatrix>
    <tmSc:Identifier>2</tmSc:Identifier>
    <ScaleDenominator>50000000</ScaleDenominator>
    <CellSize>13229.1931250529</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
  </TileMatrix>

```

```
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>13</MatrixWidth>
<MatrixHeight>14</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>3</tmsc:Identifier>
<ScaleDenominator>30000000</ScaleDenominator>
<CellSize>7937.51587503175</CellSize>
<PointOfOrigin>-34655800 39310000</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>21</MatrixWidth>
<MatrixHeight>22</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>4</tmsc:Identifier>
<ScaleDenominator>17500000</ScaleDenominator>
<CellSize>4630.21759376852</CellSize>
<PointOfOrigin>-34655800 39310000</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>36</MatrixWidth>
<MatrixHeight>38</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>5</tmsc:Identifier>
<ScaleDenominator>10000000</ScaleDenominator>
<CellSize>2645.83862501058</CellSize>
<PointOfOrigin>-34655800 39310000</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>62</MatrixWidth>
<MatrixHeight>66</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>6</tmsc:Identifier>
<ScaleDenominator>6000000</ScaleDenominator>
<CellSize>1587.50317500635</CellSize>
<PointOfOrigin>-34655800 39310000</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>103</MatrixWidth>
<MatrixHeight>110</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>7</tmsc:Identifier>
```

```
<ScaleDenominator>3500000</ScaleDenominator>
<CellSize>926.043518753704</CellSize>
<PointOfOrigin>-34655800 39310000</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>177</MatrixWidth>
<MatrixHeight>188</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>8</tmsc:Identifier>
    <ScaleDenominator>2000000</ScaleDenominator>
    <CellSize>529.167725002116</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>309</MatrixWidth>
    <MatrixHeight>329</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>9</tmsc:Identifier>
    <ScaleDenominator>1200000</ScaleDenominator>
    <CellSize>317.50063500127</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>515</MatrixWidth>
    <MatrixHeight>548</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>10</tmsc:Identifier>
    <ScaleDenominator>700000</ScaleDenominator>
    <CellSize>185.20870375074</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>882</MatrixWidth>
    <MatrixHeight>938</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>11</tmsc:Identifier>
    <ScaleDenominator>420000</ScaleDenominator>
    <CellSize>111.125222250444</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1470</MatrixWidth>
    <MatrixHeight>1563</MatrixHeight>
```

```
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>12</tmsc:Identifier>
    <ScaleDenominator>250000</ScaleDenominator>
    <CellSize>66.1459656252646</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2469</MatrixWidth>
    <MatrixHeight>2626</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>13</tmsc:Identifier>
    <ScaleDenominator>145000</ScaleDenominator>
    <CellSize>38.3646600626534</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>4257</MatrixWidth>
    <MatrixHeight>4528</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>14</tmsc:Identifier>
    <ScaleDenominator>85000</ScaleDenominator>
    <CellSize>22.4896283125899</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>7262</MatrixWidth>
    <MatrixHeight>7723</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>15</tmsc:Identifier>
    <ScaleDenominator>50000</ScaleDenominator>
    <CellSize>13.2291931250529</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>12344</MatrixWidth>
    <MatrixHeight>13130</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>16</tmsc:Identifier>
    <ScaleDenominator>30000</ScaleDenominator>
    <CellSize>7.93751587503175</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
```

```
<TileHeight>256</TileHeight>
<MatrixWidth>20574</MatrixWidth>
<MatrixHeight>21882</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>17</tmsc:Identifier>
    <ScaleDenominator>17500</ScaleDenominator>
    <CellSize>4.63021759376852</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>35269</MatrixWidth>
    <MatrixHeight>37512</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>18</tmsc:Identifier>
    <ScaleDenominator>10000</ScaleDenominator>
    <CellSize>2.64583862501058</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>61720</MatrixWidth>
    <MatrixHeight>65646</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>19</tmsc:Identifier>
    <ScaleDenominator>6000</ScaleDenominator>
    <CellSize>1.58750317500635</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>102866</MatrixWidth>
    <MatrixHeight>109409</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>20</tmsc:Identifier>
    <ScaleDenominator>3500</ScaleDenominator>
    <CellSize>0.926043518753704</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>176341</MatrixWidth>
    <MatrixHeight>187558</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>21</tmsc:Identifier>
    <ScaleDenominator>2000</ScaleDenominator>
```

```
<CellSize>0.529167725002116</CellSize>
<PointOfOrigin>-34655800 39310000</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>308596</MatrixWidth>
<MatrixHeight>328227</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>22</tmsc:Identifier>
    <ScaleDenominator>1200</ScaleDenominator>
    <CellSize>0.31750063500127</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>514327</MatrixWidth>
    <MatrixHeight>547044</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>23</tmsc:Identifier>
    <ScaleDenominator>700</ScaleDenominator>
    <CellSize>0.18520870375074</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>881703</MatrixWidth>
    <MatrixHeight>937790</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>24</tmsc:Identifier>
    <ScaleDenominator>420</ScaleDenominator>
    <CellSize>0.111125222250444</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>1469505</MatrixWidth>
    <MatrixHeight>1562983</MatrixHeight>
</TileMatrix>
<TileMatrix>
    <tmsc:Identifier>25</tmsc:Identifier>
    <ScaleDenominator>250</ScaleDenominator>
    <CellSize>0.0661459656252645</CellSize>
    <PointOfOrigin>-34655800 39310000</PointOfOrigin>
    <TileWidth>256</TileWidth>
    <TileHeight>256</TileHeight>
    <MatrixWidth>2468768</MatrixWidth>
    <MatrixHeight>2625811</MatrixHeight>
</TileMatrix>
```

```
</TileMatrixSet>
```

Annex G: Example encodings for Variable Matrix Width TileMatrixSets (Informative)

This informative Annex provides examples of encodings for tile matrix sets using the Variable Matrix Width requirements class.

G.1. GNOSIS Global Grid

These are the JSON and XML definitions of the GNOSISGlobalGrid tile matrix set (see [GNOSISGlobalGrid](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

One can define an arbitrary number of zoom levels. Here, 5 zoom levels are illustrated.

G.1.1. GNOSIS Global Grid (JSON encoding)

The complete JSON example can be found in the supplemental material described in [Schema Documents \(Normative\)](#)

```
{  
  "id" : "GNOSISGlobalGrid",  
  "title" : "GNOSIS Global Grid",  
  "uri" :  
    "http://www.opengis.net/def/tilematrixset/OGC/1.0/GNOSISGlobalGrid",  
  "crs" : "http://www.opengis.net/def/crs/EPSG/0/4326",  
  "orderedAxes" : ["Lat","Lon"],  
  "wellKnownScaleSet" :  
    "http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad",  
  "tileMatrices" : [  
    {  
      "id" : "0",  
      "scaleDenominator" : 139770566.0071794390678,  
      "cellSize" : 0.3515625,  
      "cornerOfOrigin" : "topLeft",  
      "pointOfOrigin" : [ 90, -180 ],  
      "matrixWidth" : 4,  
      "matrixHeight" : 2,  
      "tileWidth" : 256,  
      "tileHeight" : 256  
    },  
    {  
      "id" : "1",  
      "scaleDenominator" : 69885283.0035897195339,
```

```

    "cellSize" : 0.17578125,
    "cornerOfOrigin" : "topLeft",
    "pointOfOrigin" : [ 90, -180 ],
    "matrixWidth" : 8,
    "matrixHeight" : 4,
    "tileWidth" : 256,
    "tileHeight" : 256,
    "variableMatrixWidths" : [
        { "coalesce" : 2, "minTileRow" : 0, "maxTileRow" : 0 },
        { "coalesce" : 2, "minTileRow" : 3, "maxTileRow" : 3 }
    ]
},
{
    "id" : "2",
    "scaleDenominator" : 34942641.501794859767,
    "cellSize" : 0.087890625,
    "cornerOfOrigin" : "topLeft",
    "pointOfOrigin" : [ 90, -180 ],
    "matrixWidth" : 16,
    "matrixHeight" : 8,
    "tileWidth" : 256,
    "tileHeight" : 256,
    "variableMatrixWidths" : [
        { "coalesce" : 4, "minTileRow" : 0, "maxTileRow" : 0 },
        { "coalesce" : 2, "minTileRow" : 1, "maxTileRow" : 1 },
        { "coalesce" : 2, "minTileRow" : 6, "maxTileRow" : 6 },
        { "coalesce" : 4, "minTileRow" : 7, "maxTileRow" : 7 }
    ]
},
{
    "id" : "3",
    "scaleDenominator" : 17471320.7508974298835,
    "cellSize" : 0.0439453125,
    "cornerOfOrigin" : "topLeft",
    "pointOfOrigin" : [ 90, -180 ],
    "matrixWidth" : 32,
    "matrixHeight" : 16,
    "tileWidth" : 256,
    "tileHeight" : 256,
    "variableMatrixWidths" : [
        { "coalesce" : 8, "minTileRow" : 0, "maxTileRow" : 0 },
        { "coalesce" : 4, "minTileRow" : 1, "maxTileRow" : 1 },
        { "coalesce" : 2, "minTileRow" : 2, "maxTileRow" : 3 },
        { "coalesce" : 2, "minTileRow" : 12, "maxTileRow" : 13 },
        { "coalesce" : 4, "minTileRow" : 14, "maxTileRow" : 14 },
        { "coalesce" : 8, "minTileRow" : 15, "maxTileRow" : 15 }
    ]
}

```

```

},
{
  "id" : "4",
  "scaleDenominator" : 8735660.3754487149417,
  "cellSize" : 0.02197265625,
  "cornerOfOrigin" : "topLeft",
  "pointOfOrigin" : [ 90, -180 ],
  "matrixWidth" : 64,
  "matrixHeight" : 32,
  "tileWidth" : 256,
  "tileHeight" : 256,
  "variableMatrixWidths" : [
    { "coalesce" : 16, "minTileRow" : 0, "maxTileRow" : 0 },
    { "coalesce" : 8, "minTileRow" : 1, "maxTileRow" : 1 },
    { "coalesce" : 4, "minTileRow" : 2, "maxTileRow" : 3 },
    { "coalesce" : 2, "minTileRow" : 4, "maxTileRow" : 7 },
    { "coalesce" : 2, "minTileRow" : 24, "maxTileRow" : 27 },
    { "coalesce" : 4, "minTileRow" : 28, "maxTileRow" : 29 },
    { "coalesce" : 8, "minTileRow" : 30, "maxTileRow" : 30 },
    { "coalesce" : 16, "minTileRow" : 31, "maxTileRow" : 31 }
  ]
}
]
}
}

```

G.1.2. GNOSIS Global Grid (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="GNOSISGlobalGrid" xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmSc="http://www.opengis.net/tms/2.0/common"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
  <tmSc:Title>GNOSIS Global Grid</tmSc:Title>
  <tmSc:Identifier>GNOSISGlobalGrid</tmSc:Identifier>

  <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/GNOSISGlobalGrid</uri>
  <tmSc:CRS>
    <tmSc:URI>http://www.opengis.net/def/crs/EPSG/0/4326</tmSc:URI>
  </tmSc:CRS>
  <OrderedAxes>Lat,Lon</OrderedAxes>

  <WellKnownScaleSet>http://www.opengis.net/def/wkss/OGC/1.0/GoogleCRS84Quad</WellKnownScaleSet>
  <TileMatrix>
    <tmSc:Identifier>0</tmSc:Identifier>

```

```

<ScaleDenominator>139770566.0071794</ScaleDenominator>
<CellSize>0.3515625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>4</MatrixWidth>
<MatrixHeight>2</MatrixHeight>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>1</tmsc:Identifier>
<ScaleDenominator>69885283.00358972</ScaleDenominator>
<CellSize>0.17578125</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>8</MatrixWidth>
<MatrixHeight>4</MatrixHeight>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>3</MinTileRow><MaxTileRow>3</MaxTileRow></VariableMatrixWidth>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>2</tmsc:Identifier>
<ScaleDenominator>34942641.50179486</ScaleDenominator>
<CellSize>0.087890625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>16</MatrixWidth>
<MatrixHeight>8</MatrixHeight>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>1</MinTileRow><MaxTileRow>1</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>6</MinTileRow><MaxTileRow>6</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>7</MinTileRow><MaxTileRow>7</MaxTileRow></VariableMatrixWidth>
</TileMatrix>
<TileMatrix>

```

```

<tmsc:Identifier>3</tmsc:Identifier>
<ScaleDenominator>17471320.7508974298835</ScaleDenominator>
<CellSize>0.0439453125</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>32</MatrixWidth>
<MatrixHeight>16</MatrixHeight>

<VariableMatrixWidth><Coalesce>8</Coalesce><MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>1</MinTileRow><MaxTileRow>1</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>2</MinTileRow><MaxTileRow>3</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>12</MinTileRow><MaxTileRow>13</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>14</MinTileRow><MaxTileRow>14</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>8</Coalesce><MinTileRow>15</MinTileRow><MaxTileRow>15</MaxTileRow></VariableMatrixWidth>
</TileMatrix>
<TileMatrix>
<tmsc:Identifier>4</tmsc:Identifier>
<ScaleDenominator>8735660.3754487149417</ScaleDenominator>
<CellSize>0.02197265625</CellSize>
<PointOfOrigin>90 -180</PointOfOrigin>
<TileWidth>256</TileWidth>
<TileHeight>256</TileHeight>
<MatrixWidth>64</MatrixWidth>
<MatrixHeight>32</MatrixHeight>

<VariableMatrixWidth><Coalesce>16</Coalesce><MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>8</Coalesce><MinTileRow>1</MinTileRow><MaxTileRow>1</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>2</MinTileRow><MaxTileRow>3</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>4</MinTileRow><MaxTile

```

```

Row>7</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>24</MinTileRow><MaxTil
eRow>27</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>28</MinTileRow><MaxTil
eRow>29</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>8</Coalesce><MinTileRow>30</MinTileRow><MaxTil
eRow>30</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>16</Coalesce><MinTileRow>31</MinTileRow><MaxTi
leRow>31</MaxTileRow></VariableMatrixWidth>
</TileMatrix>
</TileMatrixSet>
```

G.2. CDB Global Grid

These are the JSON and XML definitions of the CDBGlobalGrid tile matrix set (see [CDBGlobalGrid](#)) that can be reproduced by other standards needing to define a tile matrix set. Not all TileMatrix elements need to be included and including other TileMatrices for more detailed scales is possible if they follow the same pattern.

One can define an arbitrary number of zoom levels. Here, 3 zoom levels are illustrated.

G.2.1. CDB Global Grid (JSON encoding)

```
{
  "id" : "CDBGlobalGrid",
  "title" : "CDB Global Grid",
  "uri" : "http://www.opengis.net/def/tilematrixset/OGC/1.0/CDBGlobalGrid",
  "crs" : "http://www.opengis.net/def/crs/EPSG/0/4326",
  "orderedAxes" : ["Lat", "Lon"],
  "tileMatrices" : [
    {
      "id" : "-1",
      "scaleDenominator" : 776503.1444843303179,
      "cellSize" : 0.001953125,
      "cornerOfOrigin" : "topLeft",
      "pointOfOrigin" : [ 90, -180 ],
      "matrixWidth" : 360,
      "matrixHeight" : 180,
      "tileWidth" : 512,
      "tileHeight" : 512,
      "variableMatrixWidths" : [
        1024, 512, 256, 128, 64, 32, 16, 8, 4, 2, 1
      ]
    }
  ]
}
```

```

        {
          "coalesce" : 12, "minTileRow" : 0, "maxTileRow" : 0 },
        {
          "coalesce" : 6, "minTileRow" : 1, "maxTileRow" : 9 },
        {
          "coalesce" : 4, "minTileRow" : 10, "maxTileRow" : 14 },
        {
          "coalesce" : 3, "minTileRow" : 15, "maxTileRow" : 19 },
        {
          "coalesce" : 2, "minTileRow" : 20, "maxTileRow" : 39 },
        {
          "coalesce" : 2, "minTileRow" : 140, "maxTileRow" : 159 },
        {
          "coalesce" : 3, "minTileRow" : 160, "maxTileRow" : 164 },
        {
          "coalesce" : 4, "minTileRow" : 165, "maxTileRow" : 169 },
        {
          "coalesce" : 6, "minTileRow" : 170, "maxTileRow" : 178 },
        {
          "coalesce" : 12, "minTileRow" : 179, "maxTileRow" : 179 }
      ]
    },
    {
      "id" : "0",
      "scaleDenominator" : 388251.572242165159,
      "cellSize" : 0.0009765625,
      "cornerOfOrigin" : "topLeft",
      "pointOfOrigin" : [ 90, -180 ],
      "matrixWidth" : 360,
      "matrixHeight" : 180,
      "tileWidth" : 1024,
      "tileHeight" : 1024,
      "variableMatrixWidths" : [
        {
          "coalesce" : 12, "minTileRow" : 0, "maxTileRow" : 0 },
        {
          "coalesce" : 6, "minTileRow" : 1, "maxTileRow" : 9 },
        {
          "coalesce" : 4, "minTileRow" : 10, "maxTileRow" : 14 },
        {
          "coalesce" : 3, "minTileRow" : 15, "maxTileRow" : 19 },
        {
          "coalesce" : 2, "minTileRow" : 20, "maxTileRow" : 39 },
        {
          "coalesce" : 2, "minTileRow" : 140, "maxTileRow" : 159 },
        {
          "coalesce" : 3, "minTileRow" : 160, "maxTileRow" : 164 },
        {
          "coalesce" : 4, "minTileRow" : 165, "maxTileRow" : 169 },
        {
          "coalesce" : 6, "minTileRow" : 170, "maxTileRow" : 178 },
        {
          "coalesce" : 12, "minTileRow" : 179, "maxTileRow" : 179 }
      ]
    },
    {
      "id" : "1",
      "scaleDenominator" : 194125.7861210825795,
      "cellSize" : 0.00048828125,
      "cornerOfOrigin" : "topLeft",
      "pointOfOrigin" : [ 90, -180 ],
      "matrixWidth" : 720,
      "matrixHeight" : 360,
      "tileWidth" : 1024,
      "tileHeight" : 1024,
      "variableMatrixWidths" : [
        { "coalesce" : 12, "minTileRow" : 0, "maxTileRow" : 1 },
      ]
    }
  ]
}

```

```

        {
            "coalesce" : 6, "minTileRow" : 2, "maxTileRow" : 19 },
        {
            "coalesce" : 4, "minTileRow" : 20, "maxTileRow" : 29 },
        {
            "coalesce" : 3, "minTileRow" : 30, "maxTileRow" : 39 },
        {
            "coalesce" : 2, "minTileRow" : 40, "maxTileRow" : 79 },
        {
            "coalesce" : 2, "minTileRow" : 280, "maxTileRow" : 319 },
        {
            "coalesce" : 3, "minTileRow" : 320, "maxTileRow" : 329 },
        {
            "coalesce" : 4, "minTileRow" : 330, "maxTileRow" : 339 },
        {
            "coalesce" : 6, "minTileRow" : 340, "maxTileRow" : 357 },
        {
            "coalesce" : 12, "minTileRow" : 358, "maxTileRow" : 359 }
        ]
    }
]
}

```

G.2.2. CDB Global Grid (XML encoding)

```

<?xml version="1.0" encoding="UTF-8"?>
<TileMatrixSet id="CDBGlobalGrid" xmlns="http://www.opengis.net/tms/2.0"
    xmlns:tmSc="http://www.opengis.net/tms/2.0/common"
    xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
    xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tilematrixset.xsd">
    <tmSc:Title>CDB Global Grid</tmSc:Title>
    <tmSc:Identifier>CDBGlobalGrid</tmSc:Identifier>
    <uri>http://www.opengis.net/def/tilematrixset/OGC/1.0/CDBGlobalGrid</uri>
    <tmSc:CRS>
        <tmSc:URI>http://www.opengis.net/def/crs/EPSG/0/4326</tmSc:URI>
    </tmSc:CRS>
    <OrderedAxes>Lat,Lon</OrderedAxes>
    <TileMatrix>
        <tmSc:Identifier>-1</tmSc:Identifier>
        <ScaleDenominator>776503.1444843302015</ScaleDenominator>
        <CellSize>0.001953125</CellSize>
        <PointOfOrigin>90 -180</PointOfOrigin>
        <TileWidth>512</TileWidth>
        <TileHeight>512</TileHeight>
        <MatrixWidth>360</MatrixWidth>
        <MatrixHeight>180</MatrixHeight>

        <VariableMatrixWidth><Coalesce>12</Coalesce><MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow></VariableMatrixWidth>

        <VariableMatrixWidth><Coalesce>6</Coalesce><MinTileRow>1</MinTileRow><MaxTileRow>9</MaxTileRow></VariableMatrixWidth>
    
```

```
<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>10</MinTileRow><MaxTileRow>14</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>3</Coalesce><MinTileRow>15</MinTileRow><MaxTileRow>19</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>20</MinTileRow><MaxTileRow>39</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>140</MinTileRow><MaxTileRow>159</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>3</Coalesce><MinTileRow>160</MinTileRow><MaxTileRow>164</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>165</MinTileRow><MaxTileRow>169</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>6</Coalesce><MinTileRow>170</MinTileRow><MaxTileRow>178</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>12</Coalesce><MinTileRow>179</MinTileRow><MaxTileRow>179</MaxTileRow></VariableMatrixWidth>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>0</tmsc:Identifier>
  <ScaleDenominator>388251.5722421651008</ScaleDenominator>
  <CellSize>0.0009765625</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>1024</TileWidth>
  <TileHeight>1024</TileHeight>
  <MatrixWidth>360</MatrixWidth>
  <MatrixHeight>180</MatrixHeight>

<VariableMatrixWidth><Coalesce>12</Coalesce><MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>6</Coalesce><MinTileRow>1</MinTileRow><MaxTileRow>9</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>10</MinTileRow><MaxTileRow>14</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>3</Coalesce><MinTileRow>15</MinTileRow><MaxTileRow>19</MaxTileRow></VariableMatrixWidth>
```

```

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>20</MinTileRow><MaxTil
eRow>39</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>140</MinTileRow><MaxTi
leRow>159</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>3</Coalesce><MinTileRow>160</MinTileRow><MaxTi
leRow>164</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>165</MinTileRow><MaxTi
leRow>169</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>6</Coalesce><MinTileRow>170</MinTileRow><MaxTi
leRow>178</MaxTileRow></VariableMatrixWidth>
</TileMatrix>
<TileMatrix>
  <tmsc:Identifier>1</tmsc:Identifier>
  <ScaleDenominator>194125.7861210825504</ScaleDenominator>
  <CellSize>0.00048828125</CellSize>
  <PointOfOrigin>90 -180</PointOfOrigin>
  <TileWidth>1024</TileWidth>
  <TileHeight>1024</TileHeight>
  <MatrixWidth>720</MatrixWidth>
  <MatrixHeight>360</MatrixHeight>

<VariableMatrixWidth><Coalesce>12</Coalesce><MinTileRow>0</MinTileRow><MaxTil
eRow>1</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>6</Coalesce><MinTileRow>2</MinTileRow><MaxTil
eRow>19</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>20</MinTileRow><MaxTil
eRow>29</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>3</Coalesce><MinTileRow>30</MinTileRow><MaxTil
eRow>39</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>40</MinTileRow><MaxTil
eRow>79</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>2</Coalesce><MinTileRow>280</MinTileRow><MaxTi
leRow>319</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>3</Coalesce><MinTileRow>320</MinTileRow><MaxTi

```

```
leRow>329</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>4</Coalesce><MinTileRow>330</MinTileRow><MaxTi
leRow>339</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>6</Coalesce><MinTileRow>340</MinTileRow><MaxTi
leRow>357</MaxTileRow></VariableMatrixWidth>

<VariableMatrixWidth><Coalesce>12</Coalesce><MinTileRow>358</MinTileRow><MaxT
ileRow>359</MaxTileRow></VariableMatrixWidth>
  </TileMatrix>
</TileMatrixSet>
```

Annex H: Example encodings for TileSetMetadata (Informative)

This informative Annex provides an example of use for *TileSetMetadata* and *TileMatrixSetLimits* data structures.

The *TileSetMetadata* data structure can be used for a resource to specify the tiling scheme supported by a particular tile set, as well as additional metadata.

The *TileMatrixSetLimits* data structure is populated to indicate that the resource is only available in a fragment of the tiled space.

H.1. Example JSON encoding of TileSetMetadata

This example illustrates a *TileSetMetadata* instance encoded in JSON.

```
{  
  "title" : "Daraa multi-layer vector tiles",  
  "description" : "Vector tiles of Daraa, Syria from NSG OpenStreetMap  
Topographic Data Store",  
  "keywords" : [ "Daraa", "Syria", "OpenStreetMap", "NSG", "TDS" ],  
  "dataType" : "vector",  
  "accessConstraints" : "unclassified",  
  "crs" : "http://www.opengis.net/def/crs/EPSG/0/3857",  
  "epoch" : 2021.33,  
  "links" : [  
    {  
      "rel" : "self",  
      "type" : "application/json",  
      "href" : "/Daraa/ogcapi/tiles/WebMercatorQuad?f=json"  
    },  
    {  
      "rel" : "alternate",  
      "type" : "application/json+vdn.mapbox.tilejson",  
      "href" : "/Daraa/ogcapi/tiles/WebMercatorQuad?f=tilejson"  
    },  
    {  
      "rel" : "tiling-scheme",  
      "type" : "application/json",  
      "href" : "/Daraa/ogcapi/tileMatrixSets/WebMercatorQuad"  
    },  
    {  
      "rel" : "item",  
      "href" : "
```

```

"/Daraa/ogcapi/tiles/WebMercatorQuad/{tileMatrix}/{tileRow}/{tileCol}",
    "templated" : true
},
{
    "rel"  : "item",
    "href" :
"/Daraa/ogcapi/tiles/WebMercatorQuad/{tileMatrix}/{tileRow}/{tileCol}.json",
    "type" : "application/geo+json",
    "templated" : true
},
{
    "rel"  : "item",
    "href" :
"/Daraa/ogcapi/tiles/WebMercatorQuad/{tileMatrix}/{tileRow}/{tileCol}.mvt",
    "type" : "application/vnd.mapbox-vector-tile",
    "templated" : true
},
{
    "rel"  : "dataset",
    "href" : "/Daraa/ogcapi"
}
],
"layers" : [
{
    "id" : "AgricultureSrf",
    "dataType" : "vector",
    "geometryType" : "polygons",
    "maxTileMatrix" : "17",
    "minScaleDenominator" : 4265.4591676995678,
    "minCellSize" : 1.19432856696,
    "boundingBox" :
{
    "crs" : "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
    "lowerLeft": [35.898213, 32.4633913],
    "upperRight": [36.5614696, 32.8370158]
},
    "geoDataClasses" : [
"http://example.com/def/geodataclasses/OSMNSGTDS" ],
    "links" : [
{
        "rel"  : "http://www.opengis.net/def/rel/ogc/1.0/geodata",
        "href" : "/Daraa/ogcapi/collections/AgricultureSrf"
}
],
    "propertiesSchema" :
{
        "type": "object",

```

```

"properties": {
    "F_CODE" :
    {
        "title" : "Feature Code",
        "type" : "string",
        "enum" : [ "EA010", "EA040", "AM020" ]
    },
    "FFN" : { "type" : "integer" },
    "FCSUBTYPE" :
    {
        "title" : "Feature subtype",
        "type" : "integer",
        "enum" : [ 100380, 100384 ]
    },
    "ZI005_FNA" : { "type" : "string" }
}
},
{
    "id" : "TransportationGroundCrv",
    "dataType" : "vector",
    "geometryType" : "lines",
    "maxTileMatrix" : "17",
    "minScaleDenominator" : 4265.4591676995678,
    "minCellSize" : 1.19432856696,
    "boundingBox" :
    {
        "crs" : "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
        "lowerLeft": [35.9028738, 32.4168138],
        "upperRight": [36.5747694, 33.1424348]
    },
    "geoDataClasses" : [
        "http://example.com/def/geodataclasses/OSMNSGTDS"
    ],
    "links" : [
        {
            "rel" : "http://www.opengis.net/def/rel/ogc/1.0/geodata",
            "href" : "/Daraa/ogcapi/collections/TransportationGroundCrv"
        }
    ],
    "propertiesSchema" :
    {
        "type": "object",
        "properties" :
        {
            "F_CODE" :
            {
                "title" : "Feature Code",

```

```

        "type" : "string",
        "enum" : [ "AN010", "AP010", "AP030", "AQ040" ]
    },
    "FFN" : { "type" : "integer" },
    "FCSUBTYPE" :
    {
        "title" : "Feature subtype",
        "type" : "integer",
        "enum" : [ 100143, 100152, 100161 ]
    },
    "ZI005_FNA" : { "type" : "string" }
}
}
],
"tileMatrixSetURI" :
"http://www.opengis.net/def/tilematrixset/OGC/1.0/WebMercatorQuad",
"tileMatrixSetLimits" : [
    { "tileMatrix" : "0", "minTileRow" : 0, "maxTileRow" : 0, "minTileCol" : 0, "maxTileCol" : 0 },
    { "tileMatrix" : "1", "minTileRow" : 0, "maxTileRow" : 0, "minTileCol" : 1, "maxTileCol" : 1 },
    { "tileMatrix" : "2", "minTileRow" : 1, "maxTileRow" : 1, "minTileCol" : 2, "maxTileCol" : 2 },
    { "tileMatrix" : "3", "minTileRow" : 3, "maxTileRow" : 3, "minTileCol" : 4, "maxTileCol" : 4 },
    { "tileMatrix" : "4", "minTileRow" : 6, "maxTileRow" : 6, "minTileCol" : 9, "maxTileCol" : 9 },
    { "tileMatrix" : "5", "minTileRow" : 12, "maxTileRow" : 12, "minTileCol" : 19, "maxTileCol" : 19 },
    { "tileMatrix" : "6", "minTileRow" : 25, "maxTileRow" : 25, "minTileCol" : 38, "maxTileCol" : 38 },
    { "tileMatrix" : "7", "minTileRow" : 51, "maxTileRow" : 51, "minTileCol" : 76, "maxTileCol" : 77 },
    { "tileMatrix" : "8", "minTileRow" : 102, "maxTileRow" : 103, "minTileCol" : 152, "maxTileCol" : 154 },
    { "tileMatrix" : "9", "minTileRow" : 205, "maxTileRow" : 207, "minTileCol" : 305, "maxTileCol" : 308 },
    { "tileMatrix" : "10", "minTileRow" : 411, "maxTileRow" : 415, "minTileCol" : 611, "maxTileCol" : 617 },
    { "tileMatrix" : "11", "minTileRow" : 823, "maxTileRow" : 831, "minTileCol" : 1223, "maxTileCol" : 1234 },
    { "tileMatrix" : "12", "minTileRow" : 1646, "maxTileRow" : 1663, "minTileCol" : 2446, "maxTileCol" : 2468 },
    { "tileMatrix" : "13", "minTileRow" : 3292, "maxTileRow" : 3326, "minTileCol" : 4892, "maxTileCol" : 4937 },
    { "tileMatrix" : "14", "minTileRow" : 6584, "maxTileRow" : 6653,

```

```

    "minTileCol" : 9784, "maxTileCol" : 9875 },
    { "tileMatrix" : "15", "minTileRow" : 13169, "maxTileRow" : 13306,
    "minTileCol" : 19569, "maxTileCol" : 19751 },
    { "tileMatrix" : "16", "minTileRow" : 26339, "maxTileRow" : 26613,
    "minTileCol" : 39139, "maxTileCol" : 39503 },
    { "tileMatrix" : "17", "minTileRow" : 52679, "maxTileRow" : 53227,
    "minTileCol" : 78279, "maxTileCol" : 79007 }
],
"boundingBox" :
{
    "crs" : "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
    "lowerLeft" : [ 35, 32 ],
    "upperRight" : [ 37, 33.2671397 ]
},
"centerPoint" :
{
    "crs" : "http://www.opengis.net/def/crs/OGC/1.3/CRS84",
    "coordinates" : [ 36.2298413, 32.65020355 ],
    "scaleDenominator" : 17061.8366707982,
    "cellSize" : 4.77731426782,
    "tileMatrix" : "15"
}
}

```

H.2. TileSetMetadata JSON Schema

The TileSetMetadata data structure can be included the definition of an API that provides tiles. This JSON Schema can be used in an API definition in JSON format and to validate the previous JSON example.

```
{
    "$schema": "https://json-schema.org/draft/2019-09/schema",
    "title": "Tile Set Metadata",
    "description": "A resource describing a tileset based on the OGC TileSet Metadata Standard. At least one of the 'TileMatrixSet', or a link with 'rel' tiling-scheme",
    "type": "object",
    "required": ["dataType"],
    "properties":
    {
        "title": {
            "description": "A title for this tileset",
            "type": "string"
        },
        "description": {

```

```
        "description": "Brief narrative description of this tile set",
        "type": "string"
    },
    "keywords": {
        "description": "Keywords about this tileset",
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "version": {
        "description": "Version of the Tile Set. Changes if the data behind the tiles has been changed",
        "type": "string"
    },
    "pointOfContact": {
        "description": "Useful information to contact the authors or custodians for the Tile Set",
        "type": "string"
    },
    "accessConstraints": {
        "description": "Restrictions on the availability of the Tile Set that the user needs to be aware of before using or redistributing the Tile Set",
        "type": "string",
        "default": "unclassified",
        "enum": ["unclassified", "restricted", "confidential", "secret",
        "topSecret"]
    },
    "license": {
        "description": "License applicable to the tiles",
        "type": "string"
    },
    "mediaTypes": {
        "description": "Media types available for the tiles",
        "type": "array",
        "items": {
            "type": "string"
        }
    },
    "dataType": {
        "description": "Type of data represented in the tileset",
        "$ref": "#/definitions/dataType"
    },
    "tileMatrixSetLimits": {
```

```

    "description": "Limits for the TileRow and TileCol values for
each TileMatrix in the tileMatrixSet. If missing, there are no limits other
than the ones imposed by the TileMatrixSet. If present the TileMatrices
listed are limited and the rest not available at all",
    "type": "array",
    "items": [
      {
        "$ref": "tilematrixlimits.json"
      }
    ],
  },
  "crs": {
    "description": "Coordinate Reference System (CRS)",
    "$ref": "crs.json"
  },
  "epoch": {
    "description": "Epoch of the Coordinate Reference System (CRS)",
    "type": "number"
  },
  "boundingBox": {
    "description": "Minimum bounding rectangle surrounding the tile
matrix set, in the supported CRS",
    "$ref": "2DBoundingBox.json"
  },
  "created": {
    "description": "When the Tile Set was first produced",
    "$ref": "timeStamp.json"
  },
  "updated": {
    "description": "Last Tile Set change/revision",
    "$ref": "timeStamp.json"
  },
  "layers": {
    "minItems": 1,
    "type": "array",
    "items": [
      {
        "$ref": "#/definitions/geospatialData"
      }
    ],
  },
  "style": {
    "description": "Style involving all layers used to generate the
tileset",
    "$ref": "#/definitions/style"
  },
  "centerPoint": {
    "description": "Location of a tile that nicely represents the
    "
  }
}

```

```
tileset. Implementations may use this center value to set the default
location or to present a representative tile in a user interface",
    "$ref": "#/definitions/tilePoint"
},
"tileMatrixSet":
{
    "description": "Tile matrix set definition",
    "$ref": "tileMatrixSet.json"
},
"tileMatrixSetURI":
{
    "description": "Reference to a Tile Matrix Set on the OGC NA
definition server (http://www.opengis.net/def/tms/). Required if the tile
matrix set is registered on the definition server.",
    "type": "string",
    "format": "uri",
    "examples":
["http://www.opengis.net/def/tilematrixset/OGC/1.0/GNOSISGlobalGrid"]
},
"links":
{
    "description": "Links to related resources. Possible link 'rel'
values are: 'dataset' for a URL pointing to the dataset, 'tiles' for a URL
template to get the tiles; 'alternate' for a URL pointing to another
representation of the TileSetMetadata (e.g a TileJSON file); 'tiling-scheme'
for a definition of the TileMatrixSet",
    "type": "array",
    "items": {
        "$ref": "link.json"
    }
},
"definitions":
{
    "geospatialData":
    {
        "type": "object",
        "properties": {
            "title": {
                "description": "Title of this tile matrix set, normally
used for display to a human",
                "type": "string"
            },
            "description": {
                "description": "Brief narrative description of this tile
matrix set, normally available for display to a human",
                "type": "string"
            }
        }
    }
}
```

```

},
"keywords": {
    "description": "Unordered list of one or more commonly
used or formalized word(s) or phrase(s) used to describe this layer",
    "type": "string"
},
"id": {
    "description": "Unique identifier of the Layer.
Implementation of 'identifier'",
    "type": "string"
},
"dataType": {
    "description": "Type of data represented in the layer",
    "$ref": "#/definitions/dataType"
},
"geometryType": {
    "description": "The geometry type of the features shown
in this layer",
    "type": "string",
    "enum": ["points", "lines", "polygons"]
},
"featureType": {
    "description": "Feature type identifier. Only applicable
to layers of datatype 'geometries'",
    "type": "string"
},
"pointOfContact": {
    "description": "Useful information to contact the authors
or custodians for the layer (e.g. e-mail address, a physical address, phone
numbers, etc)",
    "type": "string"
},
"publisher": {
    "description": "Organization or individual responsible
for making the layer available",
    "type": "string"
},
"theme": {
    "description": "Category where the layer can be grouped",
    "type": "string"
},
"crs": {
    "description": "Coordinate Reference System (CRS)",
    "$ref": "crs.json"
},
"epoch": {
    "description": "Epoch of the Coordinate Reference System

```

```

(CRS)",
    "type": "number"
},
"minScaleDenominator": {
    "description": "Minimum scale denominator for usage of
the layer",
    "type": "number"
},
"maxScaleDenominator": {
    "description": "Maximum scale denominator for usage of
the layer",
    "type": "number"
},
"minCellSize": {
    "description": "Minimum cell size for usage of the
layer",
    "type": "number"
},
"maxCellSize": {
    "description": "Maximum cell size for usage of the
layer",
    "type": "number"
},
"maxTileMatrix": {
    "description": "TileMatrix identifier associated with the
minScaleDenominator",
    "type": "string"
},
"minTileMatrix": {
    "description": "TileMatrix identifier associated with the
maxScaleDenominator",
    "type": "string"
},
"boundingBox": {
    "description": "Minimum bounding rectangle surrounding
the layer",
    "$ref": "2DBoundingBox.json"
},
"created": {
    "description": "When the layer was first produced",
    "$ref": "timeStamp.json"
},
"updated": {
    "description": "Last layer change/revision",
    "$ref": "timeStamp.json"
},
"style": {

```

```

        "description": "Style used to generate the layer in the
tileset",
        "$ref": "#/definitions/style"
    },
    "geoDataClasses": {
        "description": "URI identifying a class of data contained
in this layer (useful to determine compatibility with styles or processes)",
        "type": "array",
        "items":{
            "type": "string"
        }
    },
    "propertiesSchema": {
        "description": "Properties represented by the features in
this layer. Can be the attributes of a feature dataset (datatype=geometries)
or the rangeType of a coverage (datatype=coverage)",
        "$ref": "#/definitions/propertiesSchema"
    },
    "links": {
        "description": "Links related to this layer. Possible
link 'rel' values are: 'collection' for a URL pointing to the collection",
        "type": "array",
        "minItems": 1,
        "items":{
            {
                "$ref": "link.json"
            }
        }
    }
},
"propertiesSchema": {
    "description": "Attributes of the features or rangetypes of a
coverage. Defined by a subset of the JSON Schema for the properties of a
feature",
    "type": "object",
    "properties": {
        "type": { "type": "string", "enum": ["object"]},
        "required": { "description": "Implements 'multiplicity' by
citing property 'name' defined as 'additionalProperties'",
            "type": "array", "minItems": 1 },
        "properties": {
            "type": "object",
            "default": {},
            "additionalProperties": {
                "description": "No property names are defined but any
property name they should be described by JSON Schema. So"
            }
        }
    }
}

```

```

'additionalProperties' implements 'name' "",
    "type": "object",
    "properties":
    {
        "title": {"type": "string"},
        "description": {"description":"Implements
'description'", "type": "string"},
        "type": {"type": "string", "enum": [
            "array",
            "boolean",
            "integer",
            "null",
            "number",
            "object",
            "string"
        ]},
        "enum": { "description": "Implements
'acceptedValues'", "type": "array", "minItems": 1, "uniqueItems": true },
        "format": { "description": "Complements
implementation of 'type'", "type": "string"},
        "contentMediaType": { "description": "Implements
'mediaType'", "type": "string" },
        "maximum": { "description": "Implements 'range'",
"type": "number"},

        "exclusiveMaximum": { "description": "Implements
'range'", "type": "number"},

        "minimum": { "description": "Implements 'range'",
"type": "number"},

        "exclusiveMinimum": { "description": "Implements
'range'", "type": "number"},

        "pattern": {
            "type": "string",
            "format": "regex"
        },
        "maxItems": { "description": "Implements
'upperMultiplicity'",

            "type": "integer",
            "minimum": 0
        },
        "minItems": { "description": "Implements
'lowerMultiplicity'",

            "type": "integer",
            "minimum": 0,
            "default": 0
        },
        "observedProperty": {
            "type": "string"
        }
    }
}

```

```

        },
        "observedPropertyURI": {
            "type": "string",
            "format": "uri"
        },
        "uom": {
            "type": "string"
        },
        "uomURI": {
            "type": "string",
            "format": "uri"
        }
    }
}
},
"tilePoint": {
    "type": "object",
    "properties": {
        "coordinates": {
            "type": "array",
            "minItems": 2,
            "maxItems": 2,
            "items": {
                "type": "number"
            }
        },
        "crs": {
            "description": "Coordinate Reference System (CRS) of the
coordinates",
            "$ref": "crs.json"
        },
        "tileMatrix": {
            "description": "TileMatrix identifier associated with the
scaleDenominator",
            "type": "string"
        },
        "scaleDenominator": {
            "description": "Scale denominator of the tile matrix
selected",
        }
    }
}
}

```

```

        "type": "number"
    },
    "cellSize":
    {
        "description": "Cell size of the tile matrix selected",
        "type": "number"
    }
}
},
"style":
{
    "type": "object",
    "required":
    [
        "id"
    ],
    "properties":
    {
        "id":
        {
            "description": "An identifier for this style.
Implementation of 'identifier'",
            "type": "string"
        },
        "title":
        {
            "description": "A title for this style",
            "type": "string"
        },
        "description":
        {
            "description": "Brief narrative description of this
style",
            "type": "string"
        },
        "keywords":
        {
            "description": "keywords about this style",
            "type": "array",
            "items":
            {
                "type": "string"
            }
        },
        "links":
        {
            "description": "Links to style related resources."
        }
    }
}

```

Possible link 'rel' values are: 'style' for a URL pointing to the style description, 'styleSpec' for a URL pointing to the specification or standard used to define the style.",

```

        "type": "array",
        "minItems": 1,
        "items":
        {
            "$ref": "link.json"
        }
    },
    "examples":
    [
        {
            "id": "night",
            "title": "Topographic night style",
            "links":
            [
                {
                    "href":
                    "https://example.com/api/1.0/styles/night?f=mapbox",
                    "type": "application/vnd.mapbox.style+json",
                    "rel": "stylesheet"
                },
                {
                    "href":
                    "https://example.com/api/1.0/styles/night?f=sld10",
                    "type":
                    "application/vnd.ogc.sld+xml;version=1.0",
                    "rel": "stylesheet"
                },
                {
                    "href":
                    "https://example.com/api/1.0/styles/night/metadata?f=json",
                    "type": "application/json",
                    "rel": "describedBy"
                }
            ]
        },
        {
            "id": "topographic",
            "title": "Regular topographic style",
            "links":
            [
                {
                    "href":
                    "https://example.com/api/1.0/styles/topographic?f=mapbox",

```

```

        "type": "application/vnd.mapbox.style+json",
        "rel": "stylesheet"
    },
    {
        "href":
        "https://example.com/api/1.0/styles/topographic?f=sld10",
        "type":
        "application/vnd.ogc.sld+xml;version=1.0",
        "rel": "stylesheet"
    },
    {
        "href":
        "https://example.com/api/1.0/styles/topographic/metadata?f=json",
        "type": "application/json",
        "rel": "describedBy"
    }
]
}
],
{
    "dataType":
    {
        "type": "string",
        "enum": ["map", "vector", "coverage"]
    }
}
}
}

```

H.3. TileMatrixSetLimits JSON Schema

This JSON Schema is referenced by the TileSetMetadata schema to define limits for each tile matrix.

```
{
    "$schema": "https://json-schema.org/draft/2019-09/schema",
    "title": "TileMatrixLimits",
    "description": "A resource describing useful to create an array that describes the limits for a tile set TileMatrixSet based on the OGC TileSet Metadata Standard",
    "type": "object",
    "required":
    [
        "tileMatrix",
        "minTileRow",
        "maxTileRow",

```

```

        "minTileCol",
        "maxTileCol"
    ],
    "properties":
    {
        "tileMatrix":
        {
            "type": "string",
            "examples": ["5"]
        },
        "minTileRow":
        {
            "type": "number",
            "format": "integer",
            "minimum": 0,
            "examples": [0]
        },
        "maxTileRow":
        {
            "type": "number",
            "format": "integer",
            "minimum": 0,
            "examples": [1]
        },
        "minTileCol":
        {
            "type": "number",
            "format": "integer",
            "minimum": 0,
            "examples": [3]
        },
        "maxTileCol":
        {
            "type": "number",
            "format": "integer",
            "minimum": 0,
            "examples": [4]
        }
    }
}

```

H.4. Example XML encoding of TileSetMetadata

This example illustrates a TileSetMetadata instance encoded in XML.

```

<?xml version="1.0" encoding="UTF-8"?>
<TileSetMetadata xmlns="http://www.opengis.net/tms/2.0"
  xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/tms/2.0 ../../tileset.xsd">
  <tmsc:Title>Daraa multi-layer vector tiles</tmsc:Title>
  <tmsc:Description>Vector tiles of Daraa, Syria from NSG OpenStreetMap
  Topographic Data Store</tmsc:Description>
  <tmsc:Keywords>
    <tmsc:Keyword>Daraa</tmsc:Keyword>
    <tmsc:Keyword>Syria</tmsc:Keyword>
    <tmsc:Keyword>OpenStreetMap</tmsc:Keyword>
    <tmsc:Keyword>NSG</tmsc:Keyword>
    <tmsc:Keyword>TDS</tmsc:Keyword>
  </tmsc:Keywords>
  <AccessConstraints>unclassified</AccessConstraints>

  <TileMatrixSetURI>http://www.opengis.net/def/TileMatrixset/OGC/1.0/WebMercatorQuad</TileMatrixSetURI>
  <tmsc:CRS>
    <tmsc:URI>http://www.opengis.net/def/crs/EPSG/0/3857</tmsc:URI>
  </tmsc:CRS>
  <Epoch>2021.33</Epoch>
  <TileMatrixSetLimit>
    <TileMatrix>0</TileMatrix>
    <MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow>
    <MinTileCol>0</MinTileCol><MaxTileCol>0</MaxTileCol>
  </TileMatrixSetLimit>
  <TileMatrixSetLimit>
    <TileMatrix>1</TileMatrix>
    <MinTileRow>0</MinTileRow><MaxTileRow>0</MaxTileRow>
    <MinTileCol>1</MinTileCol><MaxTileCol>1</MaxTileCol>
  </TileMatrixSetLimit>
  <TileMatrixSetLimit>
    <TileMatrix>2</TileMatrix>
    <MinTileRow>1</MinTileRow><MaxTileRow>1</MaxTileRow>
    <MinTileCol>2</MinTileCol><MaxTileCol>2</MaxTileCol>
  </TileMatrixSetLimit>
  <TileMatrixSetLimit>
    <TileMatrix>3</TileMatrix>
    <MinTileRow>3</MinTileRow><MaxTileRow>3</MaxTileRow>
    <MinTileCol>4</MinTileCol><MaxTileCol>4</MaxTileCol>
  </TileMatrixSetLimit>
  <TileMatrixSetLimit>
    <TileMatrix>4</TileMatrix>
    <MinTileRow>6</MinTileRow><MaxTileRow>6</MaxTileRow>
  </TileMatrixSetLimit>

```

```
<MinTileCol>9</MinTileCol><MaxTileCol>9</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>5</TileMatrix>
    <MinTileRow>12</MinTileRow><MaxTileRow>12</MaxTileRow>
    <MinTileCol>19</MinTileCol><MaxTileCol>19</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>6</TileMatrix>
    <MinTileRow>25</MinTileRow><MaxTileRow>25</MaxTileRow>
    <MinTileCol>38</MinTileCol><MaxTileCol>38</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>7</TileMatrix>
    <MinTileRow>51</MinTileRow><MaxTileRow>51</MaxTileRow>
    <MinTileCol>76</MinTileCol><MaxTileCol>77</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>8</TileMatrix>
    <MinTileRow>102</MinTileRow><MaxTileRow>103</MaxTileRow>
    <MinTileCol>152</MinTileCol><MaxTileCol>154</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>9</TileMatrix>
    <MinTileRow>205</MinTileRow><MaxTileRow>207</MaxTileRow>
    <MinTileCol>305</MinTileCol><MaxTileCol>308</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>10</TileMatrix>
    <MinTileRow>411</MinTileRow><MaxTileRow>415</MaxTileRow>
    <MinTileCol>611</MinTileCol><MaxTileCol>617</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>11</TileMatrix>
    <MinTileRow>823</MinTileRow><MaxTileRow>831</MaxTileRow>
    <MinTileCol>1223</MinTileCol><MaxTileCol>1234</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>12</TileMatrix>
    <MinTileRow>1646</MinTileRow><MaxTileRow>1663</MaxTileRow>
    <MinTileCol>2446</MinTileCol><MaxTileCol>2468</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>13</TileMatrix>
    <MinTileRow>3292</MinTileRow><MaxTileRow>3326</MaxTileRow>
    <MinTileCol>4892</MinTileCol><MaxTileCol>4937</MaxTileCol>
</TileMatrixSetLimit>
```

```

<TileMatrixSetLimit>
    <TileMatrix>14</TileMatrix>
    <MinTileRow>6584</MinTileRow><MaxTileRow>6653</MaxTileRow>
    <MinTileCol>9784</MinTileCol><MaxTileCol>9875</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>15</TileMatrix>
    <MinTileRow>13169</MinTileRow><MaxTileRow>13306</MaxTileRow>
    <MinTileCol>19569</MinTileCol><MaxTileCol>19751</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>16</TileMatrix>
    <MinTileRow>26339</MinTileRow><MaxTileRow>26613</MaxTileRow>
    <MinTileCol>39139</MinTileCol><MaxTileCol>39503</MaxTileCol>
</TileMatrixSetLimit>
<TileMatrixSetLimit>
    <TileMatrix>17</TileMatrix>
    <MinTileRow>52679</MinTileRow><MaxTileRow>53227</MaxTileRow>
    <MinTileCol>78279</MinTileCol><MaxTileCol>79007</MaxTileCol>
</TileMatrixSetLimit>
<tmsc:BoundingBox>
    <tmsc:LowerLeft>35 32</tmsc:LowerLeft>
    <tmsc:UpperRight>37 33.2671397</tmsc:UpperRight>
</tmsc:BoundingBox>
<Layer>
    <tmsc:Identifier>AgricultureSrf</tmsc:Identifier>
    <GeometryType>polygons</GeometryType>
    <MinScaleDenominator>4265.4591676995678</MinScaleDenominator>
    <MinCellSize>1.19432856696</MinCellSize>
    <MaxTileMatrix>17</MaxTileMatrix>
    <tmsc:BoundingBox crs="http://www.opengis.net/def/crs/OGC/1.3/CRS84">
        <tmsc:LowerLeft>35.898213 32.4633913</tmsc:LowerLeft>
        <tmsc:UpperRight>36.5614696 32.8370158</tmsc:UpperRight>
    </tmsc:BoundingBox>

<GeoDataClass>http://example.com/def/geodataclasses/OSMNSGTDS</GeoDataClass>
    <PropertySchema>
        <tmsc:Title>Feature Code</tmsc:Title>
        <Type>string</Type>
        <AcceptedValues>EA010</AcceptedValues>
        <AcceptedValues>EA040</AcceptedValues>
        <AcceptedValues>AM020</AcceptedValues>
        <tmsc:Identifier>F_CODE</tmsc:Identifier>
    </PropertySchema>
    <PropertySchema>
        <Type>integer</Type>
        <tmsc:Identifier>FFN</tmsc:Identifier>
    </PropertySchema>

```

```

    </PropertySchema>
    <PropertySchema>
        <tmsc:Title>Feature subtype</tmsc:Title>
        <Type>integer</Type>
        <AcceptedValues>100380</AcceptedValues>
        <AcceptedValues>100384</AcceptedValues>
        <tmsc:Identifier>FCSUBTYPE</tmsc:Identifier>
    </PropertySchema>
    <PropertySchema>
        <Type>string</Type>
        <tmsc:Identifier>ZI005_FNA</tmsc:Identifier>
    </PropertySchema>
    <Link rel="http://www.opengis.net/def/rel/ogc/1.0/geodata"
 href="/Daraa/ogcapi/collections/AgricultureSrf"/>
    <DataType>vector</DataType>
</Layer>
<Layer>
    <tmsc:Identifier>TransportationGroundCrv</tmsc:Identifier>
    <GeometryType>lines</GeometryType>
    <MinScaleDenominator>4265.4591676995678</MinScaleDenominator>
    <MinCellSize>1.19432856696</MinCellSize>
    <MaxTileMatrix>17</MaxTileMatrix>
    <tmsc:BoundingBox crs="http://www.opengis.net/def/crs/OGC/1.3/CRS84">
        <tmsc:LowerLeft>35.9028738 32.4168138</tmsc:LowerLeft>
        <tmsc:UpperRight>36.5747694 33.1424348</tmsc:UpperRight>
    </tmsc:BoundingBox>

<GeoDataClass>http://example.com/def/geodataclasses/OSMNSGTDS</GeoDataClass>
    <PropertySchema>
        <tmsc:Title>Feature Code</tmsc:Title>
        <Type>string</Type>
        <AcceptedValues>AN010</AcceptedValues>
        <AcceptedValues>AP010</AcceptedValues>
        <AcceptedValues>AN030</AcceptedValues>
        <AcceptedValues>AQ040</AcceptedValues>
        <tmsc:Identifier>F_CODE</tmsc:Identifier>
    </PropertySchema>
    <PropertySchema>
        <Type>integer</Type>
        <tmsc:Identifier>FFN</tmsc:Identifier>
    </PropertySchema>
    <PropertySchema>
        <tmsc:Title>Feature subtype</tmsc:Title>
        <Type>integer</Type>
        <AcceptedValues>100143</AcceptedValues>
        <AcceptedValues>100152</AcceptedValues>
        <AcceptedValues>100161</AcceptedValues>
    </PropertySchema>

```

```

<tmsc:Identifier>FCSTSUBTYPE</tmsc:Identifier>
</PropertySchema>
<PropertySchema>
  <Type>string</Type>
  <tmsc:Identifier>ZI005_FNA</tmsc:Identifier>
</PropertySchema>
<Link rel="http://www.opengis.net/def/rel/ogc/1.0/geodata"
 href="/Daraa/ogcapi/collections/TransportationGroundCrv"/>
  <DataType>vector</DataType>
</Layer>
<CenterPoint>
  <Coordinates>36.2298413 32.65020355</Coordinates>
  <ScaleDenominator>17061.8366707982</ScaleDenominator>
  <CellSize>4.77731426782</CellSize>
  <TileMatrix>15</TileMatrix>
</CenterPoint>
<Link rel="self" href="/Daraa/ogcapi/tiles/WebMercatorQuad?f=json"
type="application/json" />
  <Link rel="alternate"
href="/Daraa/ogcapi/tiles/WebMercatorQuad?f=tilejson"
type="application/json+vnnd.mapbox.tilejson" />
    <Link rel="item"
href="/Daraa/ogcapi/tiles/WebMercatorQuad/{TileMatrix}/{tileRow}/{tileCol}"
templated="true" />
      <Link rel="item"
href="/Daraa/ogcapi/tiles/WebMercatorQuad/{TileMatrix}/{tileRow}/{tileCol}.js
on" type="application/geo+json" templated="true" />
        <Link rel="item"
href="/Daraa/ogcapi/tiles/WebMercatorQuad/{TileMatrix}/{tileRow}/{tileCol}.mv
t" type="application/vnnd.mapbox-vector-tile" templated="true" />
          <Link rel="dataset" href="/Daraa/ogcapi" />
            <Link rel="tiling-scheme"
href="/Daraa/ogcapi/TileMatrixSets/WebMercatorQuad" type="application/json"
/>
              <DataType>vector</DataType>
</TileSetMetadata>

```

H.5. TileSetMetadata XML Schema

This XML Schema can be used to validate the previous XML example.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited with XMLSpy v2008 sp1 (http://www.altova.com) by Joan Masó
(UAB-CREAF-MiraMon). Based on previous documents of Keith Pomakis. -->
<schema xmlns="http://www.w3.org/2001/XMLSchema">

```

```

xmlns:tms="http://www.opengis.net/tms/2.0"
xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:tmsc="http://www.opengis.net/tms/2.0/common"
targetNamespace="http://www.opengis.net/tms/2.0"
elementFormDefault="qualified" xml:lang="en">
    <annotation>
        <appinfo>TileMatrixSet 2021-02-02</appinfo>
        <documentation>
            This XML Schema Document encodes the TileMatrixSet data
            structures.
            TMS is an OGC Standard Copyright (c) 2021 Open Geospatial
            Consortium, Inc. All Rights Reserved.
            To obtain additional rights of use, visit
            http://www.opengeospatial.org/legal/.
        </documentation>
    </annotation>
    <!--
***** Includes and imports.

***** -->
<include schemaLocation="tilematrixset.xsd"/>
<include schemaLocation="tilematrixlimits.xsd"/>
<import namespace="http://www.w3.org/2005/Atom"
schemaLocation="../../../../kml/2.3/atom-author-link.xsd"/>
<import namespace="http://www.opengis.net/tms/2.0/common"
schemaLocation="common.xsd"/>
<!--
***** -->
<!-- ** The TileMatrixSetLink, TileMatrixSetLimits and TileMatrixLimits
element. ** -->
<!--
***** -->
<element name="TileSetMetadata">
    <annotation>
        <documentation>Metadata about the TileMatrixSet
reference.</documentation>
    </annotation>
    <complexType>
        <complexContent>
            <extension base="tmsc:DescriptionType">
                <sequence>
                    <element name="Version" type="string" minOccurs="0">
                        <annotation>
                            <documentation>Version of the Tile Set.

```

```

Changes if the data behind the tiles has been changed</documentation>
    </annotation>
</element>
<element name="PointOfContact" type="string"
minOccurs="0">
    <annotation>
        <documentation>Useful information to contact
the authors or custodians for the Tile Set</documentation>
    </annotation>
</element>
<element name="AccessConstraints" minOccurs="0">
    <annotation>
        <documentation>Restrictions on the
availability of the Tile Set that the user needs to be aware of before using
or redistributing the Tile Set</documentation>
    </annotation>
    <simpleType>
        <restriction base="string">
            <enumeration value="unclassified"/>
            <enumeration value="restricted"/>
            <enumeration value="confidential"/>
            <enumeration value="secret"/>
            <enumeration value="topSecret"/>
        </restriction>
    </simpleType>
</element>
<element name="License" type="string" minOccurs="0">
    <annotation>
        <documentation>License applicable to the
tiles</documentation>
    </annotation>
</element>
<element name="MediaType" type="tmsc:MediaTypeType"
minOccurs="0" maxOccurs="unbounded">
    <annotation>
        <documentation>Media types available for the
tiles</documentation>
    </annotation>
</element>
<element ref="tms:TileMatrixSet" minOccurs="0">
    <annotation>
        <documentation>A tileMatrixSet
definition</documentation>
    </annotation>
</element>
<element name="TileMatrixSetURI" type="anyURI"
minOccurs="0">

```

```

<annotation>
    <documentation>Reference to a tileMatrixSet.
    Points to a definition of the TileMatrixSet in the OGC NA definition server
    (http://www.opengis.net/def/tms/)</documentation>
</annotation>
</element>
<element ref="tmsc:CRS" minOccurs="0">
    <annotation>
        <documentation>Coordinate Reference System
(CRS)</documentation>
    </annotation>
</element>
<element name="Epoch" type="double" minOccurs="0">
    <annotation>
        <documentation>Epoch of the Coordinate
Reference System (CRS)</documentation>
    </annotation>
</element>
<element ref="tms:TileMatrixSetLimit" minOccurs="0"
maxOccurs="unbounded">
    <annotation>
        <documentation>Limits for the TileRow and
TileCol values for each TileMatrix in the tileMatrixSet. If missing, there
are no limits other than the ones imposed by the TileMatrixSet. If present
the TileMatrices listed are limited and the rest not available at
all</documentation>
    </annotation>
</element>
<element ref="tmsc:BoundingBox" minOccurs="0">
    <annotation>
        <documentation>
            Minimum bounding rectangle surrounding
            the visible layer presented by this tile
matrix
            set, in the supported CRS
</documentation>
    </annotation>
</element>
<element name="Created" type="dateTime"
minOccurs="0">
    <annotation>
        <documentation>Timestamp indicating when the
Tile Set was first produced</documentation>
    </annotation>
</element>
<element name="Updated" type="dateTime"
minOccurs="0">

```

```

        <annotation>
            <documentation>Timestamp of the last Tile Set
change/revision</documentation>
        </annotation>
    </element>
    <element ref="tms:Layer" minOccurs="0"
maxOccurs="unbounded">
        <annotation>
            <documentation>Style involving all layers
used to generate the tileset</documentation>
        </annotation>
    </element>
    <element ref="tms:Style" minOccurs="0">
        <annotation>
            <documentation>Style involving all layers
used to generate the tileset</documentation>
        </annotation>
    </element>
    <element ref="tms:CenterPoint" minOccurs="0">
        <annotation>
            <documentation>Location of a tile that nicely
represents the tileset. Implementations may use this center value to set the
default location or to present a representative tile in a user
interface</documentation>
        </annotation>
    </element>
    <element ref="tms:Link" minOccurs="0"
maxOccurs="unbounded">
        <annotation>
            <documentation>Links to related resources.
Possible link 'rel' values are: 'dataset' for a URL pointing to the dataset,
'tiles' for a URL template to get the tiles; 'alternate' for a URL pointing
to another representation of the TileSetMetadata (e.g a TileJSON file);
'tiling-scheme' for a definition of the TileMatrixSet</documentation>
        </annotation>
    </element>
    <element name="DataType" type="tms:DataTypeType">
        <annotation>
            <documentation>Type of data represented in
the tiles</documentation>
        </annotation>
    </element>
    <any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
        <annotation>
            <documentation>This is an extension point for
other properties in the TileMatrixSet.</documentation>

```

```

                </annotation>
            </any>
        </sequence>
        <attribute name="id" type="string">
            <annotation>
                <documentation/>
            </annotation>
        </attribute>
    </extension>
</complexContent>
</complexType>
</element>
<!-- ===== -->
<element name="Layer" type="tms:GeospatialDataType"/>
<complexType name="GeospatialDataType">
    <annotation>
        <documentation>Geospatial data type</documentation>
    </annotation>
    <complexContent>
        <extension base="tmsc:DescriptionType">
            <sequence>
                <element ref="tmsc:Identifier">
                    <annotation>
                        <documentation>Unique identifier of the Layer.  
Implemetion of 'identifier'</documentation>
                    </annotation>
                </element>
                <element name="GeometryType" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation>The geometry type of the features shown in this layer</documentation>
                    </annotation>
                    <simpleType>
                        <restriction base="string">
                            <enumeration value="points"/>
                            <enumeration value="lines"/>
                            <enumeration value="polygons"/>
                        </restriction>
                    </simpleType>
                </element>
                <element name="FeatureType" type="string" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation>Feature type identifier. Only applicable to layers of datatype 'geometries'</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </element>
        <element name="PointOfContact" type="string"
minOccurs="0">
            <annotation>
                <documentation>Useful information to contact the
authors or custodians for the layer. (e.g. e-mail address, a physical
address, phone numbers, etc)</documentation>
            </annotation>
        </element>
        <element name="Publisher" type="string" minOccurs="0">
            <annotation>
                <documentation>Organization or individual
responsible for making the layer available</documentation>
            </annotation>
        </element>
        <element name="Theme" type="string" minOccurs="0"
maxOccurs="unbounded">
            <annotation>
                <documentation>Category where the layer can be
grouped</documentation>
            </annotation>
        </element>
        <element ref="tmsc:CRS" minOccurs="0">
            <annotation>
                <documentation>Coordinate Reference System
(CRS)</documentation>
            </annotation>
        </element>
        <element name="Epoch" type="double" minOccurs="0">
            <annotation>
                <documentation>Epoch of the Coordinate Reference
System (CRS)</documentation>
            </annotation>
        </element>
        <element name="MinScaleDenominator" type="double"
minOccurs="0">
            <annotation>
                <documentation>Minimum scale denominator for
usage of the layer</documentation>
            </annotation>
        </element>
        <element name="MaxScaleDenominator" type="double"
minOccurs="0">
            <annotation>
                <documentation>Maximum scale denominator for
usage of the layer</documentation>
            </annotation>

```

```

</element>
<element name="MinCellSize" type="double" minOccurs="0">
    <annotation>
        <documentation>Minimum cel size for usage of the
layer</documentation>
    </annotation>
</element>
<element name="MaxCellSize" type="double" minOccurs="0">
    <annotation>
        <documentation>Maximum cell size for usage of the
layer</documentation>
    </annotation>
</element>
<element name="MaxTileMatrix" type="tmsc:CodeType"
minOccurs="0">
    <annotation>
        <documentation>TileMatrix identifier associated
with the MinScaleDenominator</documentation>
    </annotation>
</element>
<element name="MinTileMatrix" type="tmsc:CodeType"
minOccurs="0">
    <annotation>
        <documentation>TileMatrix identifier associated
with the MaxScaleDenominator</documentation>
    </annotation>
</element>
<element ref="tmsc:BoundingBox" minOccurs="0">
    <annotation>
        <documentation>Minimum bounding rectangle
surrounding the layer</documentation>
    </annotation>
</element>
<element name="Created" type="dateTime" minOccurs="0">
    <annotation>
        <documentation>Timestamp indicating when the
layer was first produced</documentation>
    </annotation>
</element>
<element name="Updated" type="dateTime" minOccurs="0">
    <annotation>
        <documentation>Timestamp of the last layer
change/revision</documentation>
    </annotation>
</element>
<element ref="tms:Style" minOccurs="0"
maxOccurs="unbounded">

```

```

<annotation>
    <documentation>Style used to generate the layer
in the tileset</documentation>
</annotation>
</element>
<element name="GeoDataClass" type="tmsc:CodeType"
minOccurs="0" maxOccurs="unbounded">
    <annotation>
        <documentation>URI identifying a class of data
contained in this layer (useful to determine compatibility with styles or
processes)</documentation>
    </annotation>
</element>
<element ref="tms:PropertySchema" minOccurs="0"
maxOccurs="unbounded">
    <annotation>
        <documentation>Properties represented by the
features in this layer. Can be the attributes of a feature dataset
(datatype=geometries) or the rangeType of a coverage
(datatype=coverage)</documentation>
    </annotation>
</element>
<element ref="tms:Link" minOccurs="0"
maxOccurs="unbounded">
    <annotation>
        <documentation>Links related to this layer.
Possible link 'rel' values are: 'collection' for a URL pointing to the
collection</documentation>
    </annotation>
</element>
<element name="DataType" type="tms:DataTypeType">
    <annotation>
        <documentation>Type of data represented in the
layer</documentation>
    </annotation>
</element>
<any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
    <annotation>
        <documentation>This is an extension point for
other properties in the TileMatrixSet.</documentation>
    </annotation>
</any>
</sequence>
<attribute name="id" type="string">
    <annotation>
        <documentation>

```

```

        </annotation>
    </attribute>
</extension>
</complexContent>
</complexType>
<!-- ===== -->
<element name="PropertySchema" type="tms:FeatureAttributeType"/>
<complexType name="FeatureAttributeType">
    <annotation>
        <documentation>Properties represented by the features in this layer. Can be the attributes of a feature dataset (datatype=geometries) or the rangeType of a coverage (datatype=coverage)</documentation>
    </annotation>
    <complexContent>
        <extension base="tmsc:DescriptionType">
            <sequence>
                <element name="Type" type="string">
                    <annotation>
                        <documentation>The data type of the attribute</documentation>
                    </annotation>
                </element>
                <element name="Pattern" type="string" minOccurs="0">
                    <annotation>
                        <documentation>Regular expression to validate the values of the attribute</documentation>
                    </annotation>
                </element>
                <element name="MediaType" type="tmsc:MediaTypeType" minOccurs="0">
                    <annotation>
                        <documentation>Encodings of a complex attribute (e.g. an image/png)</documentation>
                    </annotation>
                </element>
                <element name="AcceptedValues" type="string" minOccurs="0" maxOccurs="unbounded">
                    <annotation>
                        <documentation>Valid values of the attribute</documentation>
                    </annotation>
                </element>
                <element name="Range" type="string" minOccurs="0" maxOccurs="2">
                    <annotation>
                        <documentation>Range of valid values expressed as an array of two items</documentation>
                    </annotation>
                </element>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

        </annotation>
    </element>
    <element name="lowerMultiplicity"
type="nonNegativeInteger" minOccurs="0">
        <annotation>
            <documentation>Lower multiplicity of the
attribute. Default is 0 (optional)</documentation>
        </annotation>
    </element>
    <element name="upperMultiplicity"
type="nonNegativeInteger" minOccurs="0">
        <annotation>
            <documentation>Upper multiplicity of the
attribute. Default is many (unbounded)</documentation>
        </annotation>
    </element>
    <element name="ObservedProperty" type="string"
minOccurs="0">
        <annotation>
            <documentation>Measured phenomenon (variable)
label, commonly a descriptive name</documentation>
        </annotation>
    </element>
    <element name="ObservedPropertyURI" type="anyURI"
minOccurs="0">
        <annotation>
            <documentation>URI pointing to a representation
of the definition of the measured phenomenon (variable)</documentation>
        </annotation>
    </element>
    <element name="UoM" type="string" minOccurs="0">
        <annotation>
            <documentation>Units of measure characterizing
the values of the attribute</documentation>
        </annotation>
    </element>
    <element name="UoMURI" type="anyURI" minOccurs="0">
        <annotation>
            <documentation>URI pointing to a representation
of the definition of the units of measure characterizing the values of the
attribute</documentation>
        </annotation>
    </element>
    <element ref="tmsc:Identifier">
        <annotation>
            <documentation>Unique identifier of the
attribute</documentation>

```

```

        </annotation>
    </element>
    <any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
        <annotation>
            <documentation>This is an extension point for
other properties in the TileMatrixSet.</documentation>
        </annotation>
        </any>
    </sequence>
</extension>
</complexContent>
</complexType>
<!-- ===== -->
<element name="Style" type="tms:StyleType"/>
<complexType name="StyleType">
    <annotation>
        <documentation>Style used to generate the tiles</documentation>
    </annotation>
    <complexContent>
        <extension base="tmsc:DescriptionType">
            <sequence>
                <element ref="tms:Link" minOccurs="0"
maxOccurs="unbounded">
                    <annotation>
                        <documentation>Links to related resources.
Possible link 'rel' values are: 'style' for a URL pointing to the style
description, 'styleSpec' for a URL pointing to the specification or standard
used to define the style</documentation>
                    </annotation>
                </element>
                <element ref="tmsc:Identifier">
                    <annotation>
                        <documentation>Unique identifier of the
stype</documentation>
                    </annotation>
                </element>
            <any namespace="##other" processContents="lax"
minOccurs="0" maxOccurs="unbounded">
                <annotation>
                    <documentation>This is an extension point for
other properties in the TileMatrixSet.</documentation>
                </annotation>
                </any>
            </sequence>
        </extension>
    </complexContent>

```

```

    </complexType>
    <!-- ===== -->
<simpleType name="DataTypeType">
    <annotation>
        <documentation>Data type for tiles and layers</documentation>
    </annotation>
    <restriction base="string">
        <enumeration value="map"/>
        <enumeration value="vector"/>
        <enumeration value="coverage"/>
    </restriction>
</simpleType>
<!-- ===== -->
<element name="CenterPoint" type="tms:TilePointType"/>
<complexType name="TilePointType">
    <annotation>
        <documentation>Point in a tile matrix</documentation>
    </annotation>
    <sequence>
        <element name="Coordinates" type="tmsc:PositionType">
            <annotation>
                <documentation>Location of the center point of the tile
set.</documentation>
            </annotation>
        </element>
        <element name="ScaleDenominator" type="double" minOccurs="0">
            <annotation>
                <documentation>Scale denominator of the tile matrix
selected</documentation>
            </annotation>
        </element>
        <element name="CellSize" type="double" minOccurs="0">
            <annotation>
                <documentation>Cell size of the tile matrix
selected</documentation>
            </annotation>
        </element>
        <element name="TileMatrix" type="tmsc:CodeType" minOccurs="0">
            <annotation>
                <documentation>TileMatrix identifier associated with the
scaleDenominator</documentation>
            </annotation>
        </element>
    </sequence>
</complexType>
<element name="Link" type="tms:LinkType"/>
<complexType name="LinkType">

```

```

<annotation>
    <documentation>Templated link based on atom link</documentation>
</annotation>
<attribute name="href" use="required"/>
<attribute name="rel"/>
<attribute name="type" type="atom:atomMedia Type"/>
<attribute name="hreflang" type="atom:atomLanguageTag"/>
<attribute name="title"/>
<attribute name="length"/>
<attribute name="templated"/>
</complexType>
</schema>

```

H.6. TileMatrixSetLimits XML Schema

This XML Schema is referenced by the TileSetMetadata schema to define limits for each tile matrix.

```

<?xml version="1.0" encoding="UTF-8"?>
<!-- Edited with XMLSpy v2008 sp1 (http://www.altova.com) by Joan Masó
(UAB-CREAF-MiraMon). Based on previous documents of Keith Pomakis. -->
<schema xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:tms="http://www.opengis.net/tms/2.0"
  targetNamespace="http://www.opengis.net/tms/2.0"
  elementFormDefault="qualified" xml:lang="en">
    <annotation>
        <appinfo>TileMatrixSet 2021-02-02</appinfo>
        <documentation>
            This XML Schema Document encodes the TileMatrixSet data
            structures.
            TMS is an OGC Standard Copyright (c) 2021 Open Geospatial
            Consortium, Inc. All Rights Reserved.
            To obtain additional rights of use, visit
            http://www.opengeospatial.org/legal/.
        </documentation>
    </annotation>
    <!--
    ****
    Includes and imports.
    ****
    -->
    <!--import namespace="http://www.w3.org/1999/xlink"
    schemaLocation="../../../../../xlink/1.0.0/xlinks.xsd"-->
    <!--
    ****

```

```

***** -->
    <!-- ** TileMatrixSetLimits and TileMatrixLimits element.
** -->
    <!--
*****
***** -->
<element name="TileMatrixSetLimit">
    <annotation>
        <documentation>Metadata describing the limits of a TileMatrix
            for this layer.</documentation>
    </annotation>
    <complexType>
        <sequence>
            <element name="TileMatrix" type="string">
                <annotation>
                    <documentation>Reference to a TileMatrix
identifier</documentation>
                </annotation>
            </element>
            <element name="MinTileRow" type="nonNegativeInteger">
                <annotation>
                    <documentation>Minimum tile row index valid for this
layer. From 0 to maxTileRow</documentation>
                </annotation>
            </element>
            <element name="MaxTileRow" type="nonNegativeInteger">
                <annotation>
                    <documentation>Maximum tile row index valid for this
layer. From minTileRow to matrixWidth-1 of the
tileMatrix
                        section of this tileMatrixSet</documentation>
                </annotation>
            </element>
            <element name="MinTileCol" type="nonNegativeInteger">
                <annotation>
                    <documentation>Minimum tile column index valid for
this
                        layer. From 0 to maxTileCol</documentation>
                </annotation>
            </element>
            <element name="MaxTileCol" type="nonNegativeInteger">
                <annotation>
                    <documentation>Maximum tile column index valid for
this layer.
                        From minTileCol to tileHeight-1 of the tileMatrix
section
                            of this tileMatrixSet.</documentation>

```

```
        </annotation>
    </element>
    <any namespace="#other" processContents="lax" minOccurs="0"
maxOccurs="unbounded">
        <annotation>
            <documentation>This is an extension point for other
properties in the TileMatrixLimits.</documentation>
        </annotation>
        </any>
    </sequence>
</complexType>
</element>
</schema>
```

Annex I: Pseudocode (Informative)

This informative Annex provides pseudocode that illustrates how to get the tiles that cover a bounding box rectangle and how to get the CRS coordinates that bound a tile.

I.1. From BBOX to tile indices

The following fragment of pseudocode could be used to convert from a desired bounding box (`bBoxMinX`, `bBoxMinY`, `bBoxMaxX`, `bBoxMaxY`) in CRS coordinates to a range of tile set indices. This pseudocode uses the same notation that subclause 6.1.1 uses. In this pseudocode it is assumed that `bBoxMinX`, `bBoxMinY`, `bBoxMaxX`, `bBoxMaxY`, `tileMatrixMinX`, `tileMatrixMinY`, `tileMatrixMaxY`, `tileSpanX` and `tileSpanY` are floating point variables (IEEE-754) that have accuracy issues derived from the finite precision of the representation. These accuracy issues could be amplified in a typical `floor()` rounding down function that could return a value ± 1 than what is expected. To overcome this issue this code uses a small value (`epsilon`) added or subtracted in a place that is not affected by CRS coordinate precision.

```
// to compensate for floating point computation inaccuracies
epsilon = 1e-6

tileMinCol = floor((bBoxMinX - tileMatrixMinX) / tileSpanX + epsilon)
tileMaxCol = floor((bBoxMaxX - tileMatrixMinX) / tileSpanX - epsilon)
tileMinRow = floor((tileMatrixMaxY - bBoxMaxY) / tileSpanY + epsilon)
tileMaxRow = floor((tileMatrixMaxY - bBoxMinY) / tileSpanY - epsilon)

// to avoid requesting out-of-range tiles
if (tileMinCol < 0) tileMinCol = 0
if (tileMaxCol >= matrixWidth) tileMaxCol = matrixWidth-1
if (tileMinRow < 0) tileMinRow = 0
if (tileMaxRow >= matrixHeight) tileMaxRow = matrixHeight-1
```

To fetch all the tiles that cover this bounding box, a client would scan through `tileMinCol` to `tileMaxCol` and `tileMinRow` to `tileMaxRow`, all inclusive. A total of $(\text{tileMaxCol} - \text{tileMinCol} + 1) \times (\text{tileMaxRow} - \text{tileMinRow} + 1)$ will be fetched.

I.2. From tile indices to BBOX

The following pseudocode could be used to convert from a pair of tile indices (`tileCol`, `tileRow`) to the bounding box (in CRS coordinates) of this tile defined by the upper-left corner (`leftX`, `upperY`) of the tile:

```
leftX = tileCol * tileSpanX + tileMatrixMinX  
upperY = tileMatrixMaxY - tileRow * tileSpanY
```

and the lower-right corner (rightX , lowerY) of the tile:

```
rightX = (tileCol+1) * tileSpanX + tileMatrixMinX  
lowerY = tileMatrixMaxY - (tileRow+1) * tileSpanY
```

Annex J: Changes from previous version (Informative)

This annex enumerates the main changes done from the previous version of this Standard. These changes were carefully designed to provide conceptual descendent compatibility. Unfortunately, provided compatibility for XML documents is very restrictive and XML encoding of the previous version will not validate with the schemas of this version. There are many other small changes between both standards.

NOTE

At the point of finalizing this document no OGC standard references version 1.0. New standards are encouraged to use this version.

J.1. TileSetMetadata

In version 1.0 there was a concept of TileMatrixSetLink (and data structure) designed to allow a dataset to declare the use of a tile matrix set defined elsewhere and, if needed, a limited coverage for this tile matrix set. In this standard, this concept has been extended beyond recognition into the TileSetMetadata structure that contains the metadata describing a set of tiles representing the same geospatial data and all conforming to a tile matrix set (a tileset), the limited coverage, and the suggested center point. Even if the new data structure looks different and covers more use cases, the previous functionality provided by the TileMatrixSetLink data structure is still included in the TileSetMetadata structure.

Equivalences from the old version into the new version are:

Previous version (1.0)	Current version (2.0)
TileMatrixSetLink	TileSetMetadata

Table 38. Equivalences between the old and new data structures

J.2. OWS Common dependency removed

In the version 1.0 there was a dependency on the OWS Common BoundingBox data structure. It is uncertain at this point if the new OGC API - Common will provide recyclable data structures. This version of the standard has copied the BoundingBox class and remove the OWS Common dependency.

// Temporarily keeping this anchor for older links to this annex...

Chapter 10. Extending TileMatrixSets for additional dimensions (Informative)

This informative annex proposes approaches for extending TileMatrixSets and TileSet metadata for indexing and accessing 3D, 4D and n-D ($n > 2$) data as tiles, regardless of whether a simple file-based data store, a database (e.g. a GeoPackage) or a web API is used.

10.1. Extension approaches

All of these approaches assume that the multi-dimensional content spans the two dimensions defined by 2D TileMatrixSets, which are usually either latitude and longitude for geographic CRSes, or X/Easting or Y/Northing for projected CRSes, as well as other extra dimensions. The TileSet CRS can have more than two dimensions by re-using a common 2D TileMatrixSet but selecting a compatible CRS that includes additional dimensions, as specified in [TileMatrixSet CRS Compatibility](#).

10.1.1. No explicit tiling of extra dimensions

One way to extend the use of TileMatrixSet to support additional dimensions is to simply regroup all content falling within a 2D tile boundary of the TileMatrixSet together in the same tile, for the full extent of those additional dimensions. In other words, the additional dimensions are not tiled.

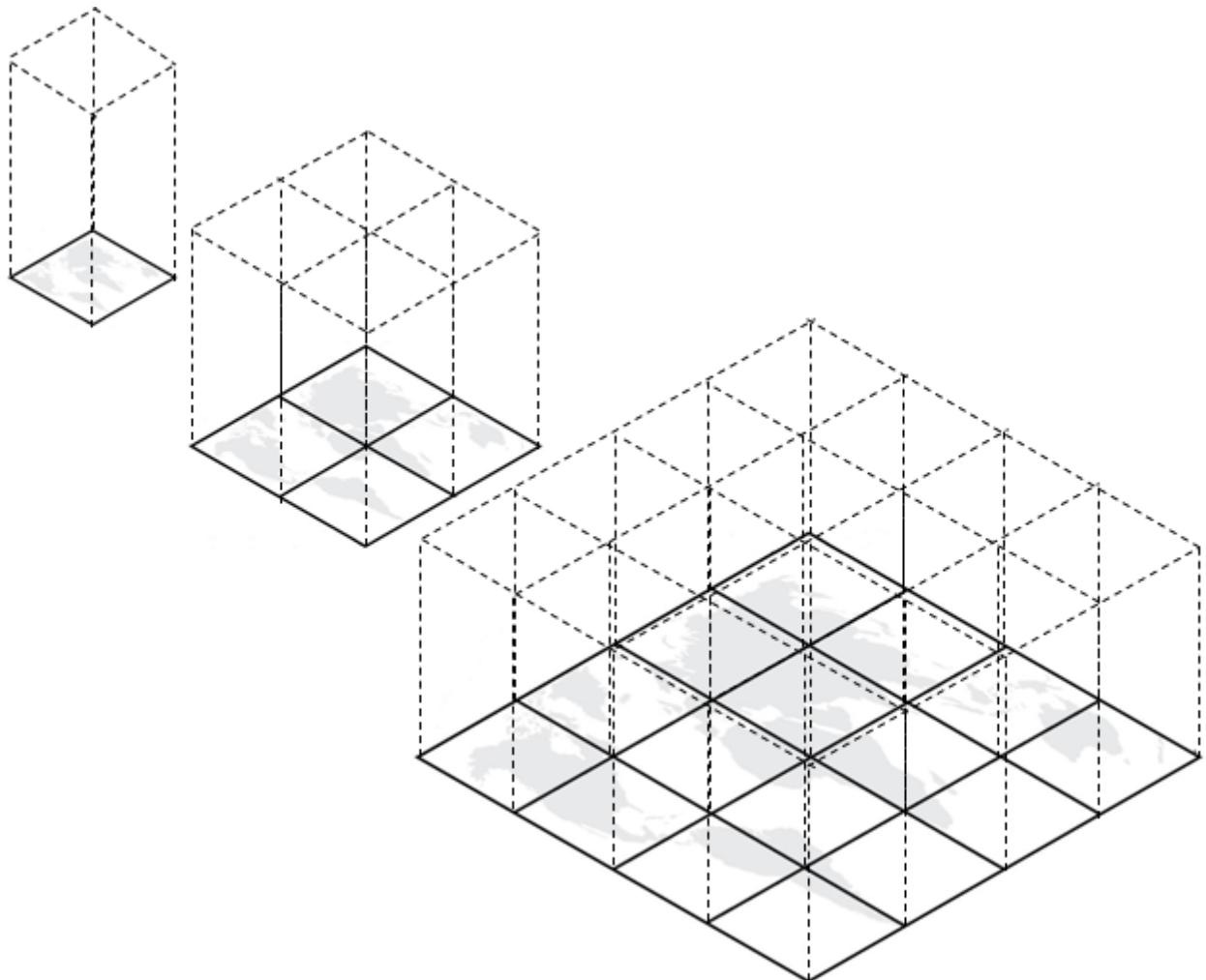


Figure 25. Extending TileMatrixSet vertically with no division

Storing, indexing and accessing the content for the additional dimensions can then be achieved through separate mechanisms, such as using formats supporting multiple dimensions (e.g. netCDF for coverages, glTF for 3D models, LAS for point clouds). Despite not being structured as tiles, extra dimensions could still be subset if the APIs such as the `subset` (trimming or slicing) and `datetime` (temporal slicing) query parameters. Note that the data density or resolution of the extra dimensions can still be reduced for the lower resolution 2D tile matrices.

This approach is the only one supported by WMTS 1.0 where it is called "extra dimensions", and it is also used in OGC CDB 1.x (see definition of the [CDB Global Grid](#)).

10.1.2. Same division for all tile matrices

Another approach is to also "tile" (divide) the extra dimension(s), but tile them in the same number of pieces for all n-D tile matrices.

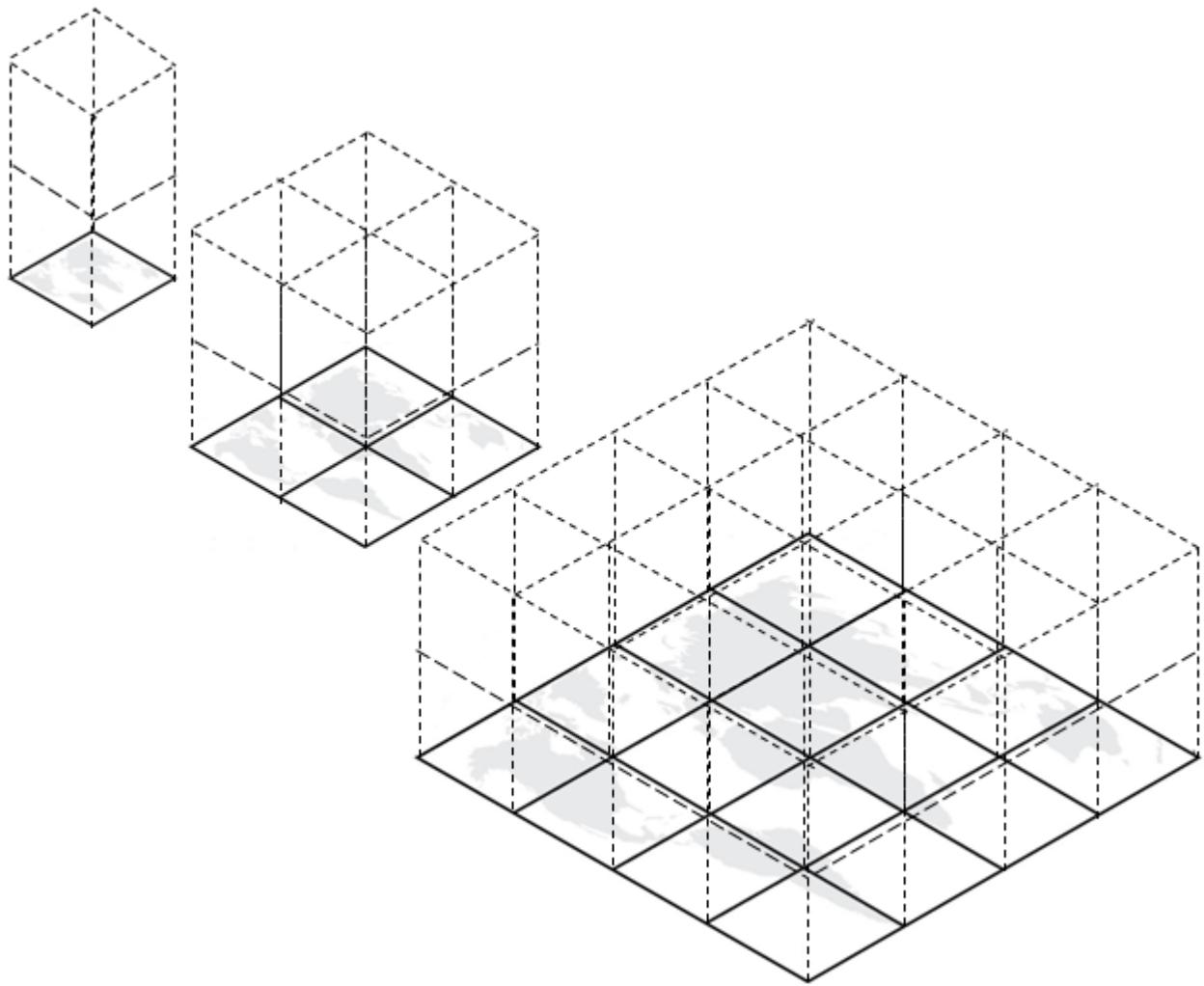


Figure 26. Extending TileMatrixSet vertically with the same tilings for all tile matrices (the illustration shows 2 "vertical" tile divisions for the extra "vertical" dimension for all tile matrices)

Fixed divisions for the extra dimensions may be suitable to scenarios where the extra dimensions have a limited extent, such as dividing all tile matrices by a small number of floors, by a few segments of altitudes, by month of the year or by a few years. Again note that the data density or resolution of the extra dimensions can still be reduced for the lower resolution 2D tile matrices

10.1.3. Octrees and Orthotrees (Hyperoctrees)

Finally, it might be beneficial to increase the number of tiles (divisions) for those additional dimensions for higher resolution tile matrices. Often, a 2D TileMatrixSet is built as a quad-tree, and a similar approach can be used with additional dimensions where it becomes an octree, or *orthree* for higher dimensions (orthant being the generalization of quadrants and octants), sometimes also called a *hyperoctree*.

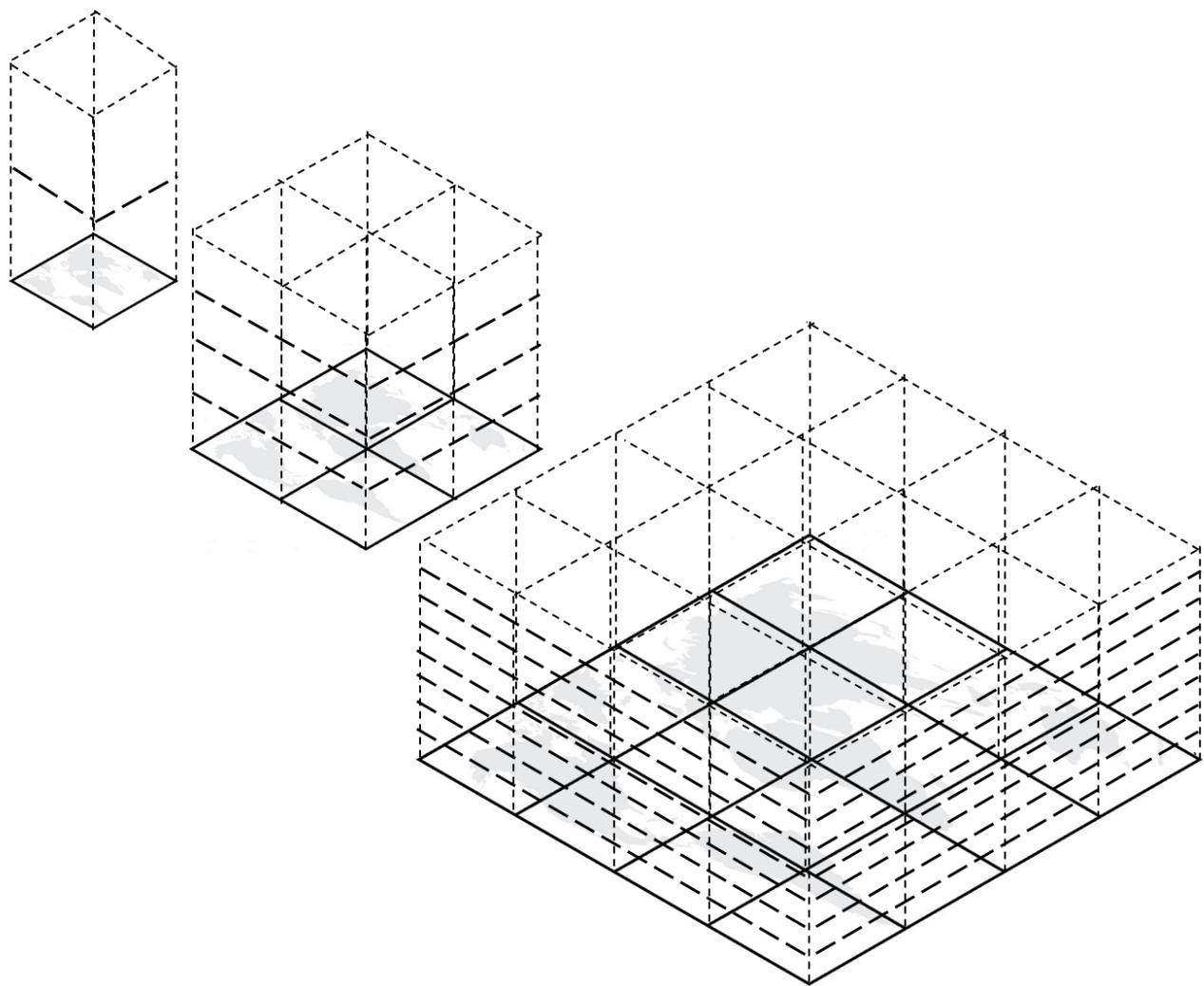


Figure 27. Extending TileMatrixSet vertically as an octree (The lowest resolution has 2 "vertical" tile divisions while the highest resolution has 8 "vertical" tile divisions)

10.2. Extended properties for TileMatrix & limits

An `extraDimensions` property can be added consisting of a list of identified (with dimension names) objects (associative array), each one with a description on how the additional dimensions are tiled. This information can be added to either a *TileMatrixSet's TileMatrix* or to a *TileSet's TileMatrixSetLimit*. Including this property in the *TileSet's TileMatrixSetLimits* allows for the re-use of common 2D TileMatrixSets.

Names	Definition	Data type and values	Multiplicity and use
identifier	Dimension identifier ^c	ows:CodeType	One (mandatory)
min	minimum value for that dimension (lower bound)	number	One (mandatory)
max	maximum value for that dimension (upper bound)	number	One (mandatory)

Names	Definition	Data type and values	Multiplicity and use
uom	unit of measure for defining min, max and resolution	string (URI)	One (mandatory)
divisions	in how many segments (trims in <i>Coverages</i> term) the space between min & max is divided at this particular tile matrix	positiveInteger	Zero or one (optional) Defaults to 1 if unspecified
resolution	resolution of the data in the extra dimension. For a coverage, this would be separate slices at that resolution interval, where the number of slices is equal to (<code>max - min</code>) / <code>divisions</code> / <code>resolution</code> , (+1 for <i>point</i> rather than <i>area</i> coverage). For 3D models, point clouds, vector features, the resolution would represent the average accuracy level at which you will find distinct coordinates.	number	Zero or one (optional)
temporalDivision	an alternative to <i>divisions</i> for temporal dimensions allowing to divide by unequal temporal units ^a	enumeration ^b	Zero or one (optional)
temporalResolution	an alternative to <i>resolution</i> for temporal dimensions allowing to specify a resolution in unequal temporal units ^a	enumeration ^b	Zero or one (optional)

^a Temporal units such as years and month have unequal number of days, and even days have unequal number of seconds when considering leap seconds. If these are not used, then *divisions* would divide the temporal range equally, and (potentially fractional) seconds would be assumed as the resolution unit.

^b Possible values are: second, minute, hour, day, week, month, year, ...

^c These *ExtraDimension* identifiers SHALL be unique (different) within the context of the TileMatrixSet.

Table 39. Properties for a ExtraDimension data structure

10.2.1. Example TileMatrix extended to 4 dimensions (2 extra dimensions)

```

{
  "identifier" : "0",
  "scaleDenominator" : 139770566.0071794390678,
  "matrixWidth" : 4, "matrixHeight" : 2,
  "tileWidth" : 256, "tileHeight" : 256,
  "pointOfOrigin" : [ 90, -180 ],
  "extraDimensions" :
  {
    "elevation" : {
      "min" : -12000, "max" : 12000,
      "uom" : "http://www.opengis.net/def/uom/SI/metre",
      "divisions" : 2,
      "resolution" : 1000
    },
    "time" : {
      "min" : "2021-01-01", "max" : "2021-12-31",
      "uom" : "http://www.opengis.net/def/uom/ISO-8601/0/Gregorian",
      "divisions" : 1,
      "temporalDivision" : "year",
      "temporalResolution" : "month"
    }
  }
}

```

In this example the low-resolution overview tile matrix would feature $4 \times 2 \times 2 \times 1$ (16) tiles, each containing e.g. a 4+D coverage (a tile may contain additional dimensions beyond what is specified in the TileMatrixSet) with $256 \times 256 \times 12 \times 12$ values (equivalent to 144 regular 2D lat/lon slices or tiles).

This *extraDimensions* property supports any of the three approaches proposed above:

- The first approach (no explicit tiling of extra dimensions) does not strictly require this *extraDimensions* property, but may benefit from the ability to explicitly list the available dimensions and their extent, and to also specify a different *resolution* at each tile matrix.
- The second approach (same division for all tile matrices) is handled by specifying the same *divisions* value for all tile matrices. Note that the *resolution* may still differ.
- For the third approach (octrees and orthotrees), both the *divisions* and *resolution* properties would differ for each tile matrix.

10.3. Data contained in tiles

10.3.1. Vector Features

Many vector formats support geometry with an extra coordinate for the depth dimension to allow defining 3D geometry for simple features such as points, lines and polygons. Some formats may also explicitly support defining solids such as polyhedrons. Different approaches can be used to tile those features, e.g. picking one of the three aforementioned ways to divide (or not divide) those extra dimensions, and deciding whether to clip the features at tile boundaries (potentially marking artificial segments to facilitate reconstruction) or allow them to spill onto neighboring tiles so as to preserve the features whole. For formats that do not allow extra dimensions, properties that are not supposed to be spatial may contain coordinates (temporal).

10.3.2. Coverages

Coverage tiles can contain additional dimensions, which may or may not have gone through a trim operation. A precise subset operation corresponds to each of the three proposed approach, which would also typically make use of a re-scaling (downsampling) operation as well to produce the lower resolution tiles.

10.3.3. Point Clouds

Point cloud data can be stored in tiles based on TileMatrixSets extended to 3D space, and thinned for lower resolution tile matrices. High-resolution point clouds would benefit from dividing the vertical dimension (using the second or third approach).

10.3.4. Point Features instantiating 3D models

One way to define 3D environments is to define 3D models in local 3D space and then instantiate them once or more by geo-referencing them and orienting them (i.e. defining a *GeoPose*). This can be accomplished by the use of point features and optional orientation and/or scaling property (which could either consist of a single or multiple values, e.g. only allowing yaw orientation or scaling all dimensions by the same factor, or also allowing to pitch and roll orientation, or separately scale the x, y and z dimensions). In CDB 1.x, this approach is used for both geo-typical as well as for geo-specific models.

It is especially suitable for shared geo-typical models which are defined only once and re-used many times, including in multiple tiles, for example to build forests re-using models of trees while varying their sizes and orientations. Such models (and potentially their textures as well) would then need to be accessible separately from the tiling hierarchy. For example, vector tiles could be available as usual at, e.g.:

```
trees/tiles/GNOSISGlobalGrid/13/5200/5715.json  
trees/tiles/GNOSISGlobalGrid/13/5200/5715.mvt
```

while referencing shared 3D models, available at `models/{modelId}` and textures at `textures/{textureId}`, e.g.:

```
trees/models/coniferous_tree01.glb  
trees/models/coniferous_tree01.e3d  
trees/textures/1.jpg
```

A specific schema can be defined for position, orientation (e.g. yaw, pitch, roll), and scale (x, y, z), and a feature encoded as GeoJSON could look like:

```
{  
  "type" : "Feature",  
  "id" : 1175,  
  "geometry" :  
  {  
    "type" : "Point",  
    "coordinates" : [ -117.1577729394728, 32.8687124736055, 0 ]  
  },  
  "properties" :  
  {  
    "modelId" : "coniferous_tree01",  
    "modelScale" : [ 1.0, 0.8, 1.1 ],  
    "modelOrientation" : [ 323.0, 0, 0 ]  
  }  
}
```

The vertical position could either be relative to the terrain, to facilitate the use of different elevation models, or absolute coordinates in the CRS (e.g. relative to the WGS84 spheroid).

With this approach, only the points are clipped to the tile's boundary, while the models themselves may extend beyond. Techniques could be used to manage handling this scenario, such as still including the points from neighboring tiles spilling into the current tile, but flagging them as such.

10.3.5. Batched 3D Models

Another approach is to define a single 3D mesh batching all content within the tile. This approach is used by *3D Tiles* and *i3s* tilesets.

This could either clip the geometry exactly at the tile's boundary, or allow spilling onto the neighboring tiles, in which case information about the precise 3D bounding volume of the tile's content can come in handy.

Ideally, the mesh is defined with the center of the tile (e.g latitude, longitude and spheroid height) as the local origin, with the local axes lined up to the TileMatrixSet axes and the vertical axis orthogonal to them, and no separate geo-referencing or orientation information is required. This way, a tile from anywhere in the world opened in a 3D model viewer or editor would always have

for example its buildings appear properly oriented upwards.

It is also useful to identify which portions of the mesh (e.g. triangles or faces) make up a particular features, e.g. the terrain, or a particular building, if supported by the 3D model format. The tiles batching 3D models can then follow the regular TileMatrixSet hierarchy. As an example the following paths could offer glTF, E3D and (*3D Tiles*) Batched 3D Model (consisting of glTF plus a header including e.g. a feature table) containing all buildings in a particular tile:

```
buildings/tiles/GNOSISGlobalGrid/13/5200/5715.glb  
buildings/tiles/GNOSISGlobalGrid/13/5200/5715.e3d  
buildings/tiles/GNOSISGlobalGrid/13/5200/5715.b3dm
```

10.4. Relationship with *3D Tiles* and *i3s*

The *3D Tiles* and *i3s* OGC community standards describe Bounding Volume Hierarchies (BVH) of 3D data. Those BVH *tiles*ets allow dimensions of each tile to differ. While a TileMatrixSet can be used as the basis for producing *3D Tiles* or *i3s* BVH tiles, not all *3D Tiles* or *i3s* tiles need to be based on a TileMatrixSet.

When deciding on the use of a TileMatrixSet to define such tiles, the space is partitioned exactly the same way, regardless of the content within that space. This has the advantage of allowing to deterministically access data for a particular portion of space from a fixed location irrespective of what or how much data may be contained in that space.

An alternative approach allowed by BVH is to distribute the data in tiles based on density, with the objective to balance the amount of data per tile, reducing the overhead of having many tiles where data is sparse, while avoiding a heavy load per tile in dense areas.

3D datasets based on a TileMatrixSet can be distributed as tiles just like 2D tilesets instead (or in addition to) as Bounding Volume Hierarchies (e.g. *3D Tiles* and/or *i3s*), where the latter can simply reference the former (e.g. linking to *.b3dm* files organized in TileMatrixSet paths).

Annex K: Revision History

Date	Release	Author	Paragraph modified	Description
2017-08-04	v0.1	Joan Maso	All	Initial draft
2018-06-07	v0.7	Joan Maso	All	Document ready for the RFC period. http://www.opengeospatial.org/standards/requests/169
2019-02-07	v0.14	Joan Maso	All	Comments accepted introduced. Document sent to the TC for final approval
2019-02-28	v0.15	Joan Maso	Section 4 and Annex D	Tiling scheme and tile set concept added. EPSG:32661 and 32662 replaced by EPSG:5041 and 5042 respectively

Annex L: Bibliography

- [1] Cavazzi S.: OGC Testbed-13: Vector Tiles Engineering Report (OGC 17-041),
<https://docs.ogc.org/per/17-041.html>
- [2] de la Beaujardiere J.: Web Map Service (WMS), version 1.3.0 (OGC 06-042),
http://portal.opengeospatial.org/files/?artifact_id=14416
- [3] Dengler K., Heinen T., Huber A., Molch, K., Mikusch E.: The EOC Geoservice: Standardized Access to Earth Observation Data Sets and Value Added Products. PV 2013, 4.-6.Nov. 2013, Frascati, Italy. <http://elib.dlr.de/86351>
- [4] DGIWG 124 Defence Profile of OGC Web Map Tile Service 1.0, 17 October 2017.
https://portal.dgiwg.org/files/?artifact_id=68271&format=pdf
- [5] Digital Globe: Web Map Tile Service Developer Guide. Cloud Services | August 2013. https://dg-cms-uploads-production.s3.amazonaws.com/uploads/document/file/174/DGCS_DeveloperGuide_WMTS.pdf
- [6] DLR: Geospatial Web Services Lists of services and their current status.
<https://geoservice.dlr.de/web/services#tilecache>
- [7] Graham D.: Project Report from the SOFWERX Geospatial Tech Sprint II (CDB X) OGC Discussion Paper (OGC 20-092)
- [8] Hobona G, Idol, T.: OGC Vector Tiles Pilot 2: Summary Engineering Report (OGC 19-088r2),
<http://docs.ogc.org/per/19-088r2.html>
- [9] Ingensand J. and Maia K.: OGC Vector Tiles Pilot: Tiled Feature Data Conceptual Model Engineering Report (OGC 18-076), <https://docs.opengeospatial.org/per/18-076.html>
- [10] Jacovella-St-Louis, J.: OGC Testbed-14: CityGML and AR Engineering Report (OGC 18-025),
<https://docs.ogc.org/per/18-025.html#ClientServerCommunication>
- [11] MapBox: MBTiles Specification. <https://github.com/mapbox/mbtiles-spec>
- [12] Masó J.: OGC Testbed-15: Maps and Tiles API Engineering Report (OGC 19-069),
<https://docs.ogc.org/per/19-069.html>
- [13] Masó J., Pomakis K. and Julià N.: Web Map Tile Service (WMTS), version 1.0. (OGC 07-057r7)
http://portal.opengeospatial.org/files/?artifact_id=35326
- [14] Masó J.: Map Tile Service (WMTS) Simple Profile, version 1.0 (OGC 13-082r2),
<http://docs.opengeospatial.org/is/13-082r2/13-082r2.html>
- [15] Meek S.: OGC Vector Tiles Pilot: Summary Engineering Report (OGC 18-086r1),

<https://docs.opengeospatial.org/per/18-086r1.html>

- [16] Müller M.: Symology Encoding (SE) Implementation Specification, version 1.1.0 (OGC 05-077r4), http://portal.opengeospatial.org/files/?artifact_id=16700
- [17] NGA: Implementation Practice Web Mercator Map Projection. NGA.SIG.0011_1.0_WEBMERC, <https://nsgreg.nga.mil/doc/view?i=4105>
- [18] NGA: Map Projections for Tiled Raster Graphics, NGA.SIG.0014_1.0_PROJRAS, V. 1.0, 24 April 2015 <https://nsgreg.nga.mil/doc/view?i=1810>
- [19] NGA, Mercator-Addendum, Map Projections for Tiled Raster Graphics Addendum for the Mercator Projection, NGA.SIG.0014_1.0_PROJRAS, V. 0.2, 2016-05-17
- [20] NGA, National System for Geospatial-Intelligence (NSG) Web Map Tile Service 1.0.0 Interoperability Standard NGA.STND.0063_1.1_WMTS, 2018-04-27
- [21] NGA: UPS-Amendment, Map Projections for Tiled Raster Graphics Amendment for Universal Polar Stereographic, NGA.SIG.0014_1.0_PROJRAS, V. 0.1, 2016-06-03.
- [22] NGA: The Universal Grids and the Transverse Mercator and Polar Stereographic Map Projections, NGA.SIG.0012_2.0.0_UTMUPS, V. 2.0.0, 25 March 2014. <https://nsgreg.nga.mil/doc/view?i=4106>
- [23] Portele C.: Geography Markup Language (GML) Encoding Standard, version 3.2.1. (OGC 07-036) http://portal.opengeospatial.org/files/?artifact_id=20509
- [24] Reed C.: OGC CDB Core Standard: Volume 1: Model and Physical Data Store Structure (OGC 15-113r5), <https://docs.opengeospatial.org/is/15-113r5/15-113r5.html>
- [25] Rollins C. and Paniccia M.: National Geospatial-Intelligence Agency ESRI Users Conference, July 22, 2015. https://www.slideshare.net/NGA_GEOINT/ngas-position-on-webmercator
- [26] Snyder J. P.: Map projections: A working manual. Professional Paper 1395. By: <https://pubs.usgs.gov/pp/1395/report.pdf>
- [27] Stefanakis, E.: Web Mercator Projection & Raster Tile Maps. Two cornerstones of Online Map Service Providers. Annual Conference – 2017 – Ottawa, May 31- June 2, 2017. http://www2.unb.ca/~estef/talks/CCA2017_Technical_Session_Stefanakis.pdf
- [28] Taleisnik S.: OGC Vector Tiles Pilot 2: Tile Set Metadata Engineering Report (OGC 19-082r1), [http://docs.opengeospatial.org/per/19-082r1.html#_metadatadate](https://docs.opengeospatial.org/per/19-082r1.html#_metadatadate)
- [29] Vretanos P.A.: OGC Vector Tiles Pilot: WFS 3.0 Vector Tiles Extension Engineering Report (OGC 18-078), <https://docs.opengeospatial.org/per/18-078.html>

[30] Vretanos P.A.: OGC Vector Tiles Pilot: WMTS Vector Tiles Extension Engineering Report (OGC 18-083), <https://docs.opengeospatial.org/per/18-083.html>

[31] Yutzler J.: Vector Tiles Pilot Extension Engineering Report (OGC 18-101), <http://docs.opengeospatial.org/per/18-101.html>