

Open Geospatial Consortium

Submission Date: <yyyy-mm-dd>

Approval Date: <yyyy-mm-dd>

Publication Date: <2020-08-12>

External identifier of this OGC® document: <http://www.opengis.net/doc/EG/ogcapi-coverages/1.0>

Internal reference number of this OGC® document: 20-075

Version: 0.1

Category: OGC® Engineering Guidance

Editor: Charles Heazel

OGC API-Coverages Users Guide

Copyright notice

Copyright © 2019 Open Geospatial Consortium

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

Warning

This document defines an the OGC Best Practices on the implementation and use of the OGC API-Coverages standards. This document is not an OGC Standard and may not be referred to as an OGC Standard. It is subject to change without notice.

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® Engineering Guidance

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 NON-INFRINGEMENT OF THIRD PARTY RIGHTS. THE COPYRIGHT HOLDER OR HOLDERS INCLUDED IN THIS NOTICE DO NOT WARRANT THAT THE FUNCTIONS CONTAINED IN THE INTELLECTUAL PROPERTY WILL MEET YOUR REQUIREMENTS OR THAT THE OPERATION OF THE INTELLECTUAL PROPERTY WILL BE UNINTERRUPTED OR ERROR FREE. ANY USE OF THE INTELLECTUAL PROPERTY SHALL BE MADE ENTIRELY AT THE USER'S OWN RISK. IN NO EVENT SHALL THE COPYRIGHT HOLDER OR ANY CONTRIBUTOR OF INTELLECTUAL PROPERTY RIGHTS TO THE INTELLECTUAL PROPERTY BE LIABLE FOR ANY CLAIM, OR ANY DIRECT, SPECIAL, INDIRECT OR CONSEQUENTIAL DAMAGES, OR ANY DAMAGES WHATSOEVER RESULTING FROM ANY ALLEGED INFRINGEMENT OR ANY LOSS OF USE, DATA OR PROFITS, WHETHER IN AN ACTION OF CONTRACT, NEGLIGENCE OR UNDER ANY OTHER LEGAL THEORY, ARISING OUT OF OR IN CONNECTION WITH THE IMPLEMENTATION, USE, COMMERCIALIZATION OR PERFORMANCE OF THIS INTELLECTUAL PROPERTY.

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

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

Table of Contents

1. Scope	5
2. References	6
3. Terms and Definitions	8
3.1. Coverage	8
3.2. Regular grid	8
3.3. Irregular grid	8
3.4. Displaced grid	8
3.5. Mesh	8
3.6. Partition [of a coverage]	8
3.7. Sensor model	8
3.8. Transformation grid	8
4. Conventions	9
4.1. Identifiers	9
5. Users Guide	10
5.1. Get Data From Point	10
5.1.1. Discussion	10
5.1.2. Return Type	11
5.2. Using API-Common	11
5.3. Using Metadata	12
5.4. Selecting Pixels	12
5.5. Range Types	16
5.6. Slicing and Subsetting	18
Annex A: Revision History	19
Annex B: Glossary	20
Annex C: Bibliography	22

i. Abstract

<Insert Abstract Text here>

ii. Keywords

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

ogcdoc, OGC document, <tags separated by commas>

iii. Preface

NOTE

Insert Preface Text here. Give OGC specific commentary: describe the technical content, reason for document, history of the document and precursors, and plans for future work. > Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium shall not be held responsible for identifying any or all such patent rights.

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

iv. Submitting organizations

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

Organization name(s)

v. Submitters

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

Name Affiliation

Chapter 1. Scope

NOTE

Insert Scope text here. Give the subject of the document and the aspects of that scope covered by the document.

Chapter 2. References

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

- Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., Berners-Lee, T.: IETF RFC 2616, **HTTP/1.1**, [RFC 2616](#)
- Rescorla, E.: IETF RFC 2818, **HTTP Over TLS**, [RFC 2818](#)
- Klyne, G., Newman, C.: IETF RFC 3339, **Date and Time on the Internet: Timestamps**, [RFC 3339](#)
- Berners-Lee, T., Fielding, R., Masinter, L.: IETF RFC 3986, **Uniform Resource Identifier (URI): Generic Syntax**, [RFC 3986](#)
- Duerst, M., Suignard, M.: IETF RFC 3987, **Internationalized Resource Identifiers (IRIs)**, [RFC 3987](#)
- Gregorio, J., Fielding, R., Hadley, M., Nottingham, M., Orchard, D.: IETF RFC 6570, **URI Template**, [RFC 6570](#)
- Nottingham, M.: IETF RFC 8288, **Web Linking**, [RFC 8288](#)
- OGC 19-072: **OGC API - Common - Part 1: Core**, (Draft) https://github.com/opengeospatial/oapi_common/blob/Master/19-072.pdf
- OGC 20-024: **OGC API - Common - Part 2: Geospatial Data**, (Draft) https://github.com/opengeospatial/oapi_common/blob/Master/20-024.pdf
- OGC 09-146: **OGC Coverage Implementation Schema (CIS)**, version 1.1, [CIS](#)
- OGC 19-008: **OGC GeoTIFF Standard**, Version 1.1, <http://docs.opengeospatial.org/is/19-008r4/19-008r4.html>
- OGC Schema: **OGC JSON Schema for Coverage Implementation Schema**, version 1.1, 2017, [CIS Schema](#)
- OGC 10-090: **OGC Network Common Data Form (NetCDF) Core Encoding Standard**, Version 1.0, http://portal.opengeospatial.org/files/?artifact_id=43732
- OGC 17-089: **OGC Web Coverage Service (WCS) Interface Standard - Core**, Version 2.1, ([WCS 2.1](#))
- Open API Initiative: **OpenAPI Specification 3.0.2**, [OpenAPI](#)
- **Schema.org**: [Schema.org](#)
- W3C: **HTML5**, W3C Recommendation, [HTML5](#)
- OGC: OGC 07-011, **Abstract Specification Topic 6: The Coverage Type and its Subtypes**, version 7.0 (identical to ISO 19123:2005), 2007
- OGC: OGC 08-094, **OGC® SWE Common Data Model Encoding Standard**, version 2, 2011
- OGC: OGC 12-000, **OGC® SensorML: Model and XML Encoding Standard**, version 2, 2014
- OGC: OGC 09-146r2, **GML 3.2.1 Application Schema – Coverages**, version 1.0.1, 2012

- OGC: OGC 16-083, **Coverage Implementation Schema – ReferenceableGridCoverage Extension**, version 1, 2017
- OGC: OGC 09-110r4, **Web Coverage Service (WCS) Core Interface Standard**, version 2, 2012
- OGC: OGC 13-102r2, **Name type specification – Time and index coordinate reference system definitions** (OGC Policy Document), version 1, 2014
- IETF: RFC 7159, **The JavaScript Object Notation (JSON) Data Interchange Format**, <https://www.ietf.org/rfc/rfc7159.txt>, 2014
- W3C: W3C JSON-LD 1.0, **A JSON-based Serialization for Linked Data**. <http://www.w3.org/TR/json-ld/>, 2014
- W3C: W3C JSON-LD 1.0 **Processing Algorithms and API**. <http://www.w3.org/TR/json-ld-api>, 2014
- W3C: W3C RDF 1.1 **Concepts and Abstract Syntax**. <https://www.w3.org/TR/2014/REC-rdf11-concepts-20140225/>, 2014

Chapter 3. Terms and Definitions

This document uses the terms defined in Sub-clause 5 of [OGC API - Common - Part 1: Core](#) (OGC 19-072), 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.

3.1. Coverage

feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain, as defined in OGC Abstract Topic 6

3.2. Regular grid

grid whose grid lines have a constant distance along each grid axis

3.3. Irregular grid

Grid whose grid lines have individual distances along each grid axis

3.4. Displaced grid

grid whose direct positions are topologically aligned to a grid, but whose geometric positions can vary arbitrarily

3.5. Mesh

coverage consisting of a collection of curves, surfaces, or solids, respectively

3.6. Partition [of a coverage]

separately stored coverage acting, by being referenced in the coverage on hand, as one of its components

3.7. Sensor model

mathematical model for estimating geolocations from recorded sensor data such as digital imagery

3.8. Transformation grid

grid whose direct positions are given by some transformation algorithm not further specified in this standard

Chapter 4. Conventions

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

4.1. Identifiers

The normative provisions in this document are denoted by the URI

<http://www.opengis.net/spec/{standard}/{m.n}>

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

Chapter 5. Users Guide

Everything up to now has been information about this document and how to use it. Topics related to the implementation of Coverages APIs are included in this section.

5.1. Get Data From Point

5.1.1. Discussion

Users might want to query a coverage for getting a data value from a single point. Typical use case is to check the height at given location from DEM but probably queries with more dimensions "temperature at location x,y,z at time ..." make sense as well.

WCS 2.0 defines two possibilities for getting a coverage from a single point.

1. Use subsets with lower limit = upper limit (trim to a point)

```
``&SUBSET=Lat(20,20)&SUBSET=Long(30,30)``
```

2. Use subsets with slice points (slice to a point)

```
``&SUBSET=Lat(20)&SUBSET=Long(30)``
```

The difference between the trimming and slicing is that trimming does not alter the dimensionality but slicing removes dimensions. If the coverage is 2D raster DEM in case 1) the result can be 1 by 1 pixel sized GeoTIFF. In case 2) the result must use format like GML because rasters have always 2 dimensions but the result must have zero.

Behavior in existing WCS 2.0 implementations vary. GeoServer sends 1x1 sized GeoTIFF with slice type of query but trim to point yields an error "Empty intersection after subsetting"

<https://osgeo-org.atlassian.net/browse/GEOS-9553>. MapServer sends 1x1 sized GeoTIFF with trim to point query but it does not accept slicing query. Because MapServer is limited to send coverages only as rasters it cannot use for example GML for 0-dimensional coverages.

To my knowledge it is much more common to use WMS GetFeatureInfo for pixel value queries than WCS. However, GetFeatureInfo does not suit well for other than 8 bit rasters, and if raster is styled for visualisation, some servers send the pixel value of the native data but others the pixel value after styling is applied. Thus GFI is a poor workaround and users deserve better.

I would like to see a recommendation about how to support point queries in OACov. From the client application side I believe that slicing together with some very simple format for rangeset would be convenient.

Unfortunately point queries are not as trivial as they may feel. In the next image query by point 1 that hits a data point should not give any troubles. Point 2 is also easy and those who think that

coverages are pixels can select the intersected pixel, and folks who prefer points can take the nearest point. The result is the same.

Queries by points 3 and 4 are not as easy. Should the result be an average of 2 or 4 nearest points, respectively (requires interpolation), or simply the value of some nearest neighbor? Which nearest? How to guarantee that different servers give uniform results? There are also coverages which are certainly not continuous and only query by point 1 is reasonable.



5.1.2. Return Type

0D support (slice to point) could return single value range set, which in `application/json` media-type could be either a number for a single band e.g.

```
403.51
```

or an object for multiple bands:

```
{ "red" : 0.434, "green" : 0.123, "blue" : 0.53 }
```

Trim to point could return 1x1 image.

5.2. Using API-Common

Provides a description of how API-Common Parts 1 and 2 should be used in a Coverage API. Key issues:

1) Collections vs. Coverages 2) Use of bbox and datetime - what is their purpose, what elements are they applied against, what are their limitations.

5.3. Using Metadata

API-Common provides sets of common metadata along with JSON schema for that metadata. Resource-specific APIs are expected to extend that metadata with information specific to their resource type.

In the case of coverages, we should distinguish between discovery metadata and exploitation metadata. Coverage metadata is that metadata used to identify and select a coverage. Exploitation metadata is metadata required to use the coverage. Note that there is some overlap between these categories. The API implementor should have the freedom to draw the line based on how they expect their API to be used.

Foot Stomp - metadata has to come from somewhere. If it is not in the coverage, then it cannot be populated in the API.

5.4. Selecting Pixels

Imagine to have some raster data with pixel size 10 by 10 units and bbox starting at 0,0. Let's put the anchor point at the centre of the pixel. Now user requests a subset as

`&SUBSET=E(572,612)&SUBSET=N(477,518)`



Which pixels should be selected?

A) All pixels that intersect with the subset



B) Only pixels which all totally contained by the subset



C) Only pixels whose anchor points are within the subset



With plain GetCoverage no resampling nor scaling should happen and if I understand right, in all three cases the envelope of the result is different from the envelope of the subsetting request. Also, if user sends a new request with subset envelope that is adjacent to the previous one, interpretation A would give duplicate pixels, interpretation B) would lead to rows of missing pixels, and only alternative C) would give a resultset that is adjacent to the first resultset.

These sections in the WCS 2.0 Core deal somehow with the case but I feel that they do not answer my question.

The WCS Core standard defines

the domain subsetting operation which delivers all data from a coverage inside a specified

request envelope (“bounding box”), relative to the coverage’s envelope – more precisely, the

intersection of the request envelope with the coverage envelope.

Requirement 32 /req/core/getCoverage-request-trim-within-extent:

Let the extent of the coverage’s gml:Envelope along the dimension specified in the trim

request range from L to H. Then, for the trim bounds trimLow and trimHigh the following

shall hold: $L \leq \text{trimLow} \leq \text{trimHigh} \leq H$.

Let further

c be the OfferedCoverage of the server addressed;

$low = tLow$ if specified in the request, otherwise low is set to the coverage's lower

bound in dimension $dname$;

$high = tHigh$ if specified in the request, otherwise $high$ is set to the coverage's upper

bound in dimension $dname$;

B be an envelope equal to the domain of c , except that in dimension $dname$ the extent

is given by the closed interval $[low, high]$;

Then, the following requirement holds:

Requirement 38 /req/core/getCoverage-response-trimming:

The response to a successful GetCoverage request on coverage identifier id of admissible

type containing no slicing and exactly one trimming operation with dimension name $dname$,

lower bound parameter evaluating to low , and upper bound parameter evaluating to $high$

shall be a coverage identical to c , but containing all points of c with location inside B , and

no other points.

NOTE This requirement does not specify the actual extent of the coverage returned. Possible options include: the minimal bounding box of the coverage returned, or the request bounding box. Servers are strongly encouraged to deliver the minimal bounding box.

I believe the correct answer is C. At least that's the way CubeWerx has implemented it. The way I see it, coverages typically deal with points, not pixels. When a coverage is returned in a pixel-based representation such as GeoTIFF, it represents each sample point by a pixel, where the sample point

is at the centre of each pixel. If a sample point intersects the requested subset, it should be returned in the GeoTIFF response as a pixel; otherwise it shouldn't.

CubeWerx's implementation of the various OGC web services distinguishes between what we call a "cell-based" extent (used, for example, by WMS) and a "grid-based" extent (used, for example, by WCS). A grid-based extent is always one pixel smaller in each dimension (0.5 pixels per side) than its corresponding cell-based extent.

So if there's a sample point at each integral unit and a coverage subset of E(572,612) N(477,518) is requested, the resulting GeoTIFF will contain 41x42 pixels (representing 41x42 sample points). Its grid-based extent is exactly what was requested (572,477 to 612,518) but its cell-based extent (that is, corner to corner) is 571.5,476.5 to 612.5,518.5. If you wanted to make a map request of this coverage subset, that'd be the extent you'd specify.

If a coverage subset of E(571.9,612.1) N(476.9,518.1) is requested, you'd get exactly the same response.

However, if a coverage subset of E(572.1,611.9) N(477.1,517.9) is requested, you'd get a response that's only 39x40 pixels in size, representing the subset E(573,611) N(478,517). It would have a grid-based extent of 573,478 to 611,517 and a cell-based extent of 572.5,478.5 to 610.5,517.5.

Reference This existing OGC document may have material you can reuse: <http://docs.opengeospatial.org/is/10-140r2/10-140r2.html> <<http://docs.opengeospatial.org/is/10-140r2/10-140r2.html>>

5.5. Range Types

In our implementation, we currently support coverages encoded in 16-bit PNG, with the 16-bit range mapped linearly to the range of values for the coverage, similar to how it is done in the [GeoPackage tile gridded coverage extension](#).

Two main points:

- How could it be possible to describe in the `.../{collectionId}` resource the range of the values that occur in the coverage? e.g. -11000..9000, or 1 to 27?

As far as I understand the RangeType does not specify this, it only says a 32-bit float or an unsigned 8 bit integer is used.

* How could a linear mapping be established from encoded values to units like meters, to use a format like PNG and maximize precision while minimizing storage size?

GeoServer WCS 2 example:


```

<swe:Quantity>
<swe:description>RED_BAND</swe:description>
<swe:nilValues>
<swe:nilValues>
<swe:nilValue reason="http://www.opengis.net/def/nil/OGC/0/unknown">0.0</swe:nilValue>
</swe:nilValues>
</swe:nilValues>
<swe:uom code="W.m-2.Sr-1"/>
<swe:constraint>
<swe:AllowedValues>
<swe:interval>100.0 150.0</swe:interval>
</swe:AllowedValues>
</swe:constraint>
</swe:Quantity>
</swe:field>

```

Here is an example using ISO 19115-2. I believe the Range Type in CIS is similar. The coverage is a SAR collection. Each pixel is an imaginary number (real and imaginary components) represented as 2s complement integers. "Count" is the SWE Common equivalent of integer. There is a lot of additional information you can add, but this was sufficient for my immediate needs.

I'm sure there is a range type you cannot represent using SWE Common, but I haven't found one yet.

```

<gco:Record>  <DataRecord xmlns="http://www.opengis.net/swe/2.0">
    <field name="RE16I">
        <Count axisID="Real">
            <description>Each component is stored in a 16-bit signed integer in
2s complement format.</description>
            <constraint>
                <AllowedValues> 0 65535 </AllowedValues>
            </constraint>
        </Count>
    </field>
    <field name="IM16I">
        <Count axisID="Imaginary">
            <description>Each component is stored in a 16-bit signed integer in
2s complement format.</description>
            <constraint>
                <AllowedValues> 0 65535 </AllowedValues>
            </constraint>
        </Count>
    </field>
</DataRecord>
</gco:Record>

```

5.6. Slicing and Subsetting

First we need to define two concepts:

subsetting - this appears to be the case where you select a subset of the range set that is at full resolution. To perform a selection you identify the range of values along each axis that should be included in the result.

thinning - appears to be the case where you reduce the resolution of the range set. This is done by removing elements from the range set and performing any smoothing or interpolation functions needed to maintain the integrity of the data.

Once we have agreed on these definitions, we can talk about the implications of these techniques for both regular and irregular grids.

Also consider edge cases: 1) non-terrestrial data 2) swath or curtain data (collection across a vertical column) 3) RF data - phase, frequency, doppler, etc.

Annex A: Revision History

Date	Release	Editor	Primary clauses modified	Description
2016-04-28	0.1	G. Editor	all	initial version

Annex B: Glossary

- **Conformance Test Module**

set of related tests, all within a single conformance test class ([OGC 08-131r3](#))

NOTE:	When no ambiguity is possible, the word test may be omitted. i.e. conformance test module is the same as conformance module . Conformance modules may be nested in a hierarchical way. This term and those associated to it are included here for consistency with ISO 19105.
--------------	---

- **Conformance Test Class; Conformance Test Level**

set of [conformance test modules](#) that must be applied to receive a single **certificate of conformance**. ([OGC 08-131r3](#))

NOTE:	When no ambiguity is possible, the word test may be left out, so conformance test class may be called a conformance class .
--------------	--

- **Executable Test Suite (ETS)**

A set of code (e.g. Java and CTL) that provides runtime tests for the assertions defined by the ATS. Test data required to do the tests are part of the ETS ([OGC 08-134](#))

- **Recommendation**

expression in the content of a document conveying that among several possibilities one is recommended as particularly suitable, without mentioning or excluding others, or that a certain course of action is preferred but not necessarily required, or that (in the negative form) a certain possibility or course of action is deprecated but not prohibited ([OGC 08-131r3](#))

NOTE:	"Although using normative language, a recommendation is not a requirement . The usual form replaces the shall (imperative or command) of a requirement with a should (suggestive or conditional)." (ISO Directives Part 2)
--------------	--

- **Requirement**

expression in the content of a document conveying criteria to be fulfilled if compliance with the document is to be claimed and from which no deviation is permitted ([OGC 08-131r3](#))

- **Requirements Class**

aggregate of all requirement modules that must all be satisfied to satisfy a conformance test class ([OGC 08-131r3](#))

- **Requirements Module**

aggregate of [requirements](#) and [recommendations](#) of a specification against a single [standardization target](#) type ([OGC 08-131r3](#))

- **Standardization Target**

entity to which some requirements of a standard apply ([OGC 08-131r3](#))

NOTE:	The standardization target is the entity which may receive a certificate of conformance for a requirements class.
--------------	---

Annex C: Bibliography

- W3C/OGC: **Spatial Data on the Web Best Practices**, W3C Working Group Note 28 September 2017, <https://www.w3.org/TR/sdw-bp/>
- W3C: **Data on the Web Best Practices**, W3C Recommendation 31 January 2017, <https://www.w3.org/TR/dwbp/>
- W3C: **Data Catalog Vocabulary**, W3C Recommendation 16 January 2014, <https://www.w3.org/TR/vocab-dcat/>
- IANA: **Link Relation Types**, <https://www.iana.org/assignments/link-relations/link-relations.xml>
- International Telecommunication Union, **ITU-T.800 : Information technology - JPEG 2000 image coding system: Core coding system**, June, 2019, <https://www.itu.int/rec/T-REC-T.800-201906-I/en>
- W3C, **RDF 1.1 Semantics**, February 2014, <https://www.w3.org/TR/rdf11-mt/>
- OGC: **OGC 13-102r2, Name type specification – Time and index coordinate reference system definitions** (OGC Policy Document), version 1, 2014