**Open Geospatial Consortium**

Date: 19/08/2017 9:44 AM

External identifier of this OGC® document:

Internal reference number of this OGC® document 17-086

Version: 0.2

Category: OGC® Interface Standard

Editors: Peter Trevelyan

Paul Hershberg,

Steve Olson

**MetOcean GetPolygon Extension for WCS2.1**

**Copyright notice**

Copyright © 2017 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

This document is not an OGC Standard. This document is distributed for review and comment. This document is subject to change without notice and may not be referred to as an OGC Standard.

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.

Contents

[1. Scope 7](#_Toc498534803)

[2. Conformance 7](#_Toc498534804)

[3. References 9](#_Toc498534805)

[4. Terms and Definitions 10](#_Toc498534806)

[4.1 numerical weather prediction model 11](#_Toc498534807)

[4.2 Time Dimension 11](#_Toc498534808)

[4.3 Web Coverage Service 2.1 (WCS2.1) 11](#_Toc498534809)

[4.4 GetCapabilities operation 11](#_Toc498534810)

[4.5 DescribeCoverage operation 11](#_Toc498534811)

[4.6 Polygon 11](#_Toc498534812)

[4.7 GetPolygon operation 12](#_Toc498534813)

[5. Conventions 12](#_Toc498534814)

[5.1 Abbreviated terms 12](#_Toc498534815)

[5.2 Schema language 12](#_Toc498534816)

[5.3 UML notation 12](#_Toc498534817)

[6. Vocabularies 12](#_Toc498534818)

[7. Non-Normative (Informative) Material 14](#_Toc498534819)

[7.1 WCS2.1 14](#_Toc498534820)

[7.2 A Short NWP (Numerical Weather Prediction) Primer 16](#_Toc498534821)

[7.3 Coverages 16](#_Toc498534822)

[7.3.1 4D Coverages 16](#_Toc498534823)

[7.4 Polygons: 17](#_Toc498534824)

[7.4.1 Encoding 17](#_Toc498534825)

[8. The core GetPolygon requirement (normative) 17](#_Toc498534826)

[8.1 Requirements class: GetPolygon 17](#_Toc498534827)

[8.1.1 Requirements class overview 19](#_Toc498534828)

[8.1.2 Metoceanpolygon::GetPolygon 19](#_Toc498534829)

[8.1.3 rsub::RangeSubset 20](#_Toc498534830)

[8.2 Requirements class: PolygonDescription 21](#_Toc498534831)

[8.2.1 Requirements class overview 21](#_Toc498534832)

[8.2.2 Requirements class overview 23](#_Toc498534833)

[8.2.3 GetPolygon Description 23](#_Toc498534834)

[8.2.4 PolygonDescription as per PolygonRing 23](#_Toc498534835)

[8.2.5 PolygonDescription as per PolygonCircle 23](#_Toc498534836)

[8.2.6 PolygonRing 24](#_Toc498534837)

[8.2.7 GML::Polygon as per PolygonRing 24](#_Toc498534838)

[8.2.8 GML::LinearRing as per PolygonRing 24](#_Toc498534839)

[8.2.9 PolygonCircle 25](#_Toc498534840)

[8.2.10 GML::Polygon as per PolygonCircle 25](#_Toc498534841)

[8.2.11 GML::Ring 25](#_Toc498534842)

[8.2.12 GML::Curve 26](#_Toc498534843)

[8.2.13 GML:: CircleByCenterPoint 26](#_Toc498534844)

[8.3 Requirements class: VerticalTemporalDescription 27](#_Toc498534845)

[8.3.1 Requirements class overview 28](#_Toc498534846)

[8.3.2 VerticalTemporalDescription Properties 28](#_Toc498534847)

[8.4 Requirements class: SubsetByTrim 29](#_Toc498534848)

[8.4.1 Requirements class overview 30](#_Toc498534849)

[8.4.2 SubsetByTrim Properties 30](#_Toc498534850)

[8.4.3 DimensionTrim Properties 31](#_Toc498534851)

[8.4.4 An example an encoding 31](#_Toc498534852)

[8.5 Requirements class: SubsetByInterpolation 32](#_Toc498534853)

[8.5.1 Requirements class overview 34](#_Toc498534854)

[8.5.2 SubsetByInterpolation Properties 34](#_Toc498534855)

[8.5.3 Interpolation Properties 34](#_Toc498534856)

[8.5.4 InterpolationPerAxis Properties 34](#_Toc498534857)

[8.5.5 CIS::GeneralGrid Properties 35](#_Toc498534858)

[8.5.6 GridLimits Properties 35](#_Toc498534859)

[8.5.7 Axis Properties 36](#_Toc498534860)

[8.5.8 IndexAxis Properties 36](#_Toc498534861)

[8.5.9 RegularAxis Properties 36](#_Toc498534862)

[8.5.10 IrregularAxis Properties 37](#_Toc498534863)

[A.1 Conformance class: getPolygon 38](#_Toc498534864)

[A.2 Conformance class: PathDescription 41](#_Toc498534865)

[A.3 Conformance class: PolygonExtent 41](#_Toc498534866)

[A.4 Conformance class: PolygonExtractionMethod 41](#_Toc498534867)

**Figures**

[Figure 1 — WCS GetCoverage operation UML class diagram 14](#_Toc498511810)

[Figure 2 — UML Diagram representing the coverage model (CIS 1.1). 15](#_Toc498511811)

[Figure 3 — Polygonal Subset (Extraction Pattern) 17](#_Toc498511812)

[Figure 4 – GetPolygon UML class diagram 19](#_Toc498511813)

[Figure 5 – PolygonDescription UML class diagram 22](#_Toc498511814)

[Figure 6 – VerticalTemporalDescription UML class diagram 28](#_Toc498511815)

[Figure 7 – SubsetByTrim UML class diagram 30](#_Toc498511816)

[Figure 8 – SubsetByInterpolation UML class diagram 33](#_Toc498511817)

**Tables**

Table 1 Summary of vocabularies within this standard 13

Table 2 METOCEANPOLYGON::GetPolygon properties 19

Table 3 RSUB::RangeSubset properties 20

Table 4 METOCEANPOLYGON::GetPolygon properties 23

Table 5 METOCEANPOLYGON::PolygonDescription properties as per PolygonRing 23

Table 6 METOCEANPOLYGON::PolygonDescription properties as per PolygonCircle 23

Table 7 METOCEANPOLYGON::PolygonRing properties 24

Table 8 GML::Polygon properties as per PolygonRing 24

Table 9 GML::LinearRing properties as per PolygonRing 24

Table 10 METOCEANPOLYGON::PolygonCircle properties 25

Table 11 GML::Polygon properties as per PolygonCircle 25

Table 12 GML::Ring 25

Table 13 GML::Curve 26

Table 14 GML:: CircleByCenterPoint 26

Table 15 METOCEANPOLYGON:: VerticalTemporalDescription properties 28

Table 16 METOCEANPOLYGON::SubsetByTrim properties 30

Table 17 METOCEANPOLYGON::DimensionTrim properties 31

Table 18 METOCEANPOLYGON::SubsetByInterpolation properties 34

Table 19 INT:: Interpolation properties 34

Table 20 INT:: InterpolationPerAxis Properties 34

Table 21 CIS:: GeneralGrid properties 35

Table 22 CIS:: GridLimits properties 35

Table 23 CIS::Axis properties 36

Table 24 CIS::IndexAxis properties 36

Table 25 CIS::RegularAxis properties 36

1. Abstract

This document defines an extension to WCS2.1, namely the extraction of data contained within a polygon defined either by a set of points or the radius and position of a circle point consisting of an information model and an XML encoding for the following two operations:

1. *GetCapabilities --* a WCS function that describes the services and operations via a GetCapabilities document.
2. *GetPolygon --* a WCS function that supports this operation to extract data from a multidimensional cube that lie within a polygon.

Metadata and vocabularies are defined that provide interoperability of these operations and documents using common semantics. The information model proposed supports MetOcean specific concepts and its user community, but these constructs may be useful and applicable to other communities.

1. Keywords

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

WCS, coverage, meteorology, oceanography, NWP, analysis, polygon observation, measurement, simulation, O&M and MetOcean

1. Preface

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.*

1. Security Considerations

This standard includes no explicit security considerations.

1. Submitting organizations

The following organizations submitted this Document to the Open Geospatial Consortium Inc.

Met Office, UK

NOAA’s National Weather Service

1. Submitters

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

|  |  |
| --- | --- |
| Name | Company |
| Peter Trevelyan | Met Office, UK |
| Paul Hershberg | National Oceanic and Atmospheric Administration (NOAA)  National Weather Service (NWS) |
| Steve Olson | National Oceanic and Atmospheric Administration (NOAA)  National Weather Service (NWS) |

1. Revision History

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Release | Authors | Paragraph Modified | Description |
| 12-11-2017 | 1.0 | Trevelyan/Hershberg/Olson | All | Created |
|  |  |  |  |  |
|  |  |  |  |  |

1. Scope

The purpose of the GetPolygon operation is to extract data contained within a polygon defined either by a set of points or the radius and position of a circle point. The need for the GetPolygon operation stems from active members of the OGC MetOcean Domain Working Group (DWG) who saw a manifest need for extraction of such information from gridded datasets.

This work has been done by members of the OGC MetOcean Domain Working Group.

1. Conformance

This standard defines:

* An amended GetCapabilities operation response that will list the GetPolygon operation and specify the token in the Sections element of the GetCapabilities request.
* A new operation “GetPolygon” that is used to extract data from a multidimensional cube contained within a polygon.
* The conformance classes that describe the GetPolygon operation.

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site[[1]](#footnote-1).

In order to conform to this OGC™interface standard, a software implementation shall choose to implement:

Any one of the conformance levels specified in Annex A (normative).

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

Requirements and conformance test URIs defined in this document are relative to:-http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/

This document establishes the following requirements and conformance classes:-

***GetPolygon*** of URI http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon getPolygon at a conceptual level in clause 8.1

The corresponding conformance class is **getPolygon** with URI http://www.opengis.net/spec/ WCS\_application-profile\_metocean\_polygon/1.0/conf/getPolygon See A.1

***PolygonDescription*** of http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/PolygonDescription defining the **PolygonRing** at a conceptual level in clause 8.2

The corresponding conformance class is **PolygonDescription** with URI http://www.opengis.net/spec/ WCS\_application-profile\_metocean\_polygon/1.0/conf/getPolygon/PolygonDescription. See A.2

***VerticalTemporalDescription*** of http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/VerticalTemporalDescription defining the **PolygonRing** at a conceptual level in clause 8.3

The corresponding conformance class is **PolygonDescription** with URI http://www.opengis.net/spec/ WCS\_application-profile\_metocean\_polygon/1.0/conf/getPolygon/VerticalTemporalDescription. See A.3

***SubsetByTrim*** of http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/SubsetByTrim defining the **VerticalTemporalDescription** at a conceptual level in clause 8.4

The corresponding conformance class is **VerticalTemporalDescription** with URI http://www.opengis.net/spec/ WCS\_application-profile\_metocean\_polygon/1.0/conf/getPolygon/SubsetByTrim. See A.4

***SubsetByInterpolation*** of http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/SubsetByInterpolation defining the **VerticalTemporalDescription** at a conceptual level in clause

The corresponding conformance class is **VerticalTemporalDescription** with URI http://www.opengis.net/spec/ WCS\_application-profile\_metocean\_polygon/1.0/conf/getPolygon/SubsetByInterpolation. See

1. References

The following normative documents contain provisions that, through referenced 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.

OGC 08-131r3 – The Specification Model – A Standard for Modular Specification, https://portal.opengeospatial.org/files/?artifact\_id=34762

ISO 19103:2005 – Geographic information - Conceptual schema language, https://www.iso.org/standard/37800.html

ISO 8601:2004 - Data elements and interchange formats – Information interchange – Representation of dates and times

OGC Abstract Specification Topic 1 – Feature geometry (aka ISO 19107), https://www.iso.org/standard/26012.html

OGC Abstract Specification Topic 2 – Spatial Referencing by Coordinates (aka ISO 19111:2007), https://www.iso.org/standard/41126.html

OGC Abstract Specification Topic 6 – Schema for Coverage geometry and functions (aka ISO 19123:2005), https://www.iso.org/standard/40121.html

OGC Abstract Specification Topic 11 – Geographic information - Metadata (aka ISO 19115:2014), https://www.iso.org/standard/53798.html

OGC Abstract Specification Topic 20 – Observations and Measurements (aka ISO 19156:2011), https://www.iso.org/standard/32574.html

OGC 07-036 Geography Mark-up Language (aka ISO 19136:2007 or GML3.2.1)

OGC® A MetOcean Profile for WCS2.1- Core OGC Document 15-045r6, <https://portal.opengeospatial.org/files/?artifact_id=72750&version=1>

OGC® Web Coverage Service 2.1 Interface Standard - Core OGC Document 09-110r7, <https://portal.opengeospatial.org/files/?artifact_id=70516&version=1>

OGC® Web Coverage Service 2.0 Interface Standard – Scaling Extension 12-039r1,

<https://portal.opengeospatial.org/files/?artifact_id=54504&version=3>

OGC® Web Coverage Service 2.0 Interface Standard – Interpolation 12-0494r3,

<https://portal.opengeospatial.org/files/?artifact_id=54502&version=3>

OGC® Web Coverage Service 2.0 CRS Extension 11-053r1,

<https://portal.opengeospatial.org/files/?artifact_id=47658&version=1>

OGC Best Practice for using Web Map Services (WMS) with time dependant or Elevation dependant data. Document 12-111, <https://portal.opengeospatial.org/files/?artifact_id=56394&version=2>

OGC Observations and Measurements v2.0 XML OGC Document 10-025r1, <http://www.opengis.net/doc/IS/OMXML/2.0>

OGC SWE Common Data Model Encoding Standard v2.0 OGC Document 08-094r1, [http://www.opengis.net/doc/IS/SWECommon/2.0](http://www.opengis.net/doc/IS/SWECommon/2.0%20)

Unified Code for Units of Measure (UCUM) – Version 1.9, 2013

Unified Modelling Language (UML). Version 2.3. May 2010

Extensible Mark-up Language (XML) – Version 1.0 (Fourth Edition), August 2006

XML Schema – Version 1.0 (Second Edition), October 2004

Coverage Implementation Schema OGC 09-146r6, <https://portal.opengeospatial.org/files/?artifact_id=65455&version=1>

1. Terms and Definitions

This document uses the terms defined in Sub-clause 5.3 of [OGC 06-121r8], which is based on the ISO/IEC Directives, Part 2, Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

For the purposes of this document, the following additional terms and definitions apply. There is some variation in the specific use of some technical terms within the meteorological domain. We have attempted to follow common usage, referring where possible to the WMO No.306[*http://www.wmo.int/pages/prog/www/WMOCodes*](http://www.wmo.int/pages/prog/www/WMOCodes)*.*

* 1. numerical weather prediction model

A numerical weather prediction model is a mathematical model of the atmosphere and oceans used to predict the weather based on current weather conditions and are normally run at set times each day.

Synonyms for numerical weather prediction model: forecast model, NWP Model, simulation

An Example of a numerical weather prediction model: The ECMWF model that runs twice per day and creates a ten day prediction of the global atmosphere.

* 1. Time Dimension

A typical numerical model simulation has the property of start and end time of the simulation. Intermediate times within the simulation are times for which the data may be used as estimate of the forecast conditions at that time. The use of time within this document does assume that the “Time Dimension” of the simulation falls within the start and end times of the simulation. The standard ISO8601 notation is used to describe time.

* 1. [Web Coverage Service](http://www.opengeospatial.org/standards/wcs) 2.1 (WCS2.1)

Web Coverage Service (WCS) is an OGC standard that refers to the exchange of geospatial information as ‘coverages’: digital geospatial information representing space-varying phenomena.

* 1. GetCapabilities operation

The getCapabilities is a WCS operation involving a machine to machine communication. A getCapabilities request to a WCS server returns a list of what operations and services (“capabilities”) are being offered by that server.

* 1. DescribeCoverage operation

A DescribeCoverage is a WCS operation involving a machine-to-machine communication. A DescribeCoverage request to a WCS server returns additional information about a coverage that a client wants to query. Generally speaking, a DescribeCoverage response includes information about the CRS, the metadata, the domain, the range and the formats available. A client generally will need to issue a DescribeCoverage request before it can make the proper GetCoverage request.

* 1. Polygon

A Polygon, in this document, consist of a plane 2D shape with a depth and time dimension. The polygon may be multi-dimensional, and in the case of aviation is often four dimensions, i.e. x, y, z, t.

* 1. GetPolygon operation

GetPolygon is a newly proposed MetOcean operation involving a machine to machine communication. A GetPolygon request to a WCS server returns a polygon coverage based on a trajectory path with a lateral and vertical extent (the polygon).

1. Conventions
   1. Abbreviated terms

GML Geography Mark-up Language

O&M Observations and Measurements

OGC Open Geospatial Consortium

MetOcean Meteorological/Oceanographic

NWP Numerical Weather Prediction

SWE OGC Sensor Web Enablement

UML Unified Modelling Language

WCS2.1 OGC Web Coverage Service version 2.1

WMO World Meteorological Organisation

XML W3C Extensible Markup Language

XSD W3C XML Schema Definition Language

* 1. Schema language

The XML implementation specified in this Standard is described using the XML Schema language (XSD) [XML Schema Part 1: Structures, XML Schema Part 2: Datatypes] and Schematron [ISO/IEC 19757-3, Information technology — Document Schema Definition

Languages (DSDL) — Part 3: Rule-based validation — Schematron].

* 1. UML notation

The diagrams that appear in this standard are presented using the Unified Modeling Language (UML) static structure diagram.

**Note:** Within the context of this standard, the following color scheme is used to identify the package in which the class exists. This is just for informative purposes.

Blue: WCS2.1 plus extensions (rsub, scal, int and crs)

Orange: CIS (Coverage Implantation Schema 1.1)

Green: This standard

1. Vocabularies

This standard defines a number of properties that require the use of codes or vocabulary items. In some cases, a list of terms are provided. The MetOcean Profile, on which this builds, has a specific vocabulary provided by the WMO (World Meteorological Office). These vocabularies are concerned with the naming of parameters (variables) used in the rangeSubset element, the coordinate reference systems (aka fixedSurfacetypeAndUnits) used in the srsName attribute, the units of measure, and the significance of time codes. The following table lists the references used within this document.

Table 1 Summary of vocabularies within this standard

| **Code list** | **Code reference** |
| --- | --- |
| GRIB edition 2 | http://codes.wmo.int/\_grib2 |
| Discipline | http://codes.wmo.int/grib2/codeflag/\_0.0 |
| Fixed surface types and units | http://codes.wmo.int/grib2/codeflag/\_4.5 |
| Parameter category | http://codes.wmo.int/grib2/codeflag/\_4.1 |
| Parameter number | http://codes.wmo.int/grib2/codeflag/\_4.2 |

1. Non-Normative (Informative) Material

The GetPolygon extension for WCS2.1 is an initiative of the MetOcean DWG to enhance the WCS2.1 core profile to extract coverages other than those extracted using the simple SLICE and TRIM methods provided by the core GetCoverage operation shown in Figure 1. For reference to the core WCS2.1 see OGC® Web Coverage Service 2.1 Interface Standard - Core OGC Document 09-110r7 (<https://portal.opengeospatial.org/files/?artifact_id=70516&version=1>). This specific extension is designed specifically to extract polygons from multidimensional cubes such as those created by numerical simulations (i.e. NWP) commonly found in the MetOcean community.



Figure 1 — WCS GetCoverage operation UML class diagram

The need for this work arises out of the growing need to transfer increasing amounts of data across networks. This can, and should, be done more efficiently by sub-setting the data. This profile specifies how data contained within a polygon may be extracted on the WCS server and transferred to the client. The document also details how the OGC’s WCS2.1 Core is extended to include an additional GetPolygon operation. Finally, the Coverage Implementation Schema (CIS1.1) is used to describe the vertical and temporal definition of the Polygon.

* 1. WCS2.1

The WCS2.1 files (see <https://portal.opengeospatial.org/files/?artifact_id=67116&version=1>) that form the core standard and the extensions (see below) describe the GetCapabilities, DescribeCoverage and GetCoverage operations. The GetPolygon extension will use the following extensions to WCS core:

* WCS Range Subsetting Extension, version 1.0.0, (OGC 12-040 )
* WCS Scaling Extension, version 1.0.0, (OGC 12-039)
* WCS Range Subsetting version 1.0, (OGC 12-040)
* WCS Interpolation Extension, version 1.0.0, (OGC 12-049)
* WCS CRS Extension version 1.0, (OGC 11-053)

The main benefit of WCS2.1 core to the MetOcean Profile, and specifically the getPolygon operation, is that it allows the description of a CIS1.1 Coverage (see Figure 2). This is important as CIS1.1 supports multi-dimensional coverages and the encoding of coverage types such as polygons.

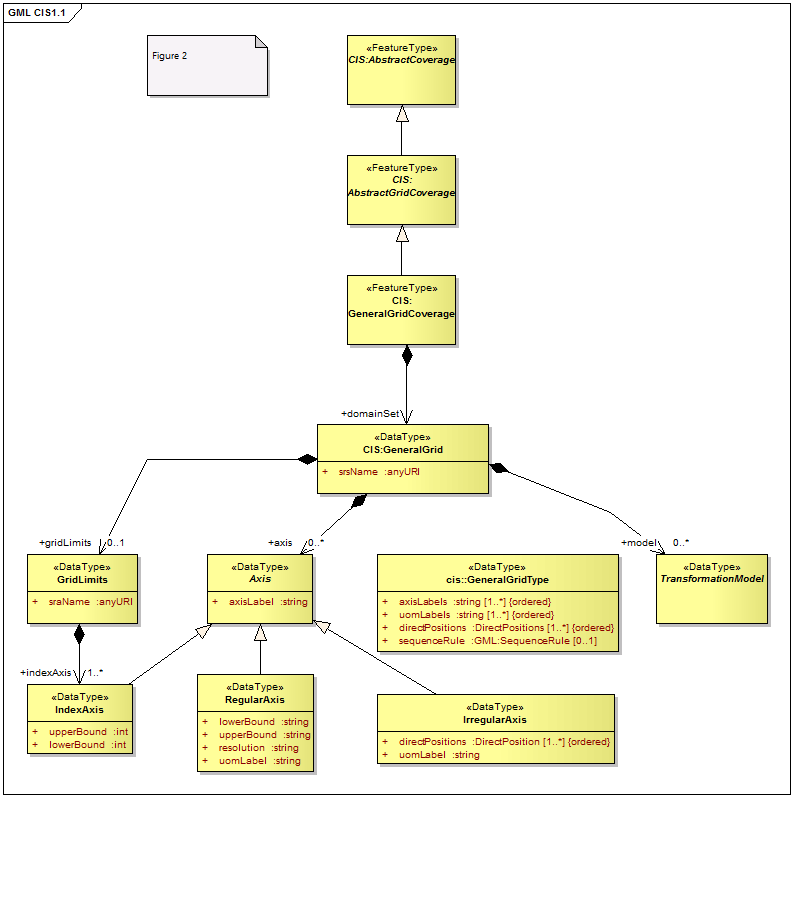


Figure 2 — UML Diagram representing the coverage model (CIS 1.1).

* 1. A Short NWP (Numerical Weather Prediction) Primer

The term “NWP model” refers to a computer simulation used to forecast the future state of the ocean/atmosphere. A NWP model is normally “run” at a set time and repeated at regular intervals during the day. Each model run has a “Time Dimension” that has “start time”, “end time” and intermediate times. These intermediate times are at set intervals and often referred to as “forecast times”. For each “forecast time”, there will be an estimate of the atmospheric/oceanic conditions for that time. The model start time is a notional time that sets a “base” time for the time intervals. Thus a time interval may have an absolute time e.g. 2017-05-15T00:00:00Z. Or, a time interval may use the start time as reference e.g. PT30H. This signifies a time that is 30 hours ahead of the “start time”.

* 1. Coverages

A “coverage” contains a “DomainSet” component describing the coverage’s domain (i.e. the locations for which values are stored in the coverage) and a “range­Set” component containing the values of the coverage. A “coverage” also contains a RangeType element that describes the coverage's range set data structure that consists of one or more fields (also referred to as parameters) that uses the SWE Common [OGC 08-094] Data­Record. The metadata component represents an extensible slot for metadata. The CIS1.1 UML diagram is shown in Figure 2.

* + 1. 4D Coverages

A typical NWP forecast is normally a set of 2D rectified grids; although more advanced, grids may be used. A typical model run contains literally thousands of 2D grids and each may described using the CIS1.1 “General Grid” coverage model.

These coverages may described and accessed by using the OGC’s Web Coverage Service (WCS). The main services are “GetCapabilities”, “DescribeCoverage” and “GetCoverage”. A problem with this approach is that the metadata returned by the GetCapabilites response and the number of GetCoverage requests quickly becomes unmanageable. By creating a multidimensional coverage from the 2D coverages, the amount of data and number of GetCoverage requests are greatly reduced, often by a factor of 100.

The key to creating a multidimensional coverage is OGC’s CIS1.1 “Coverage” model. This model makes it much easier to describe the “Domain Set” as a multi-dimensional geometry object.

A typical numerical simulation has a number of different vertical coordinates (i.e. pressure, height above mean sea level, height above ground, surface, and max wind level). By forming a 4D coverage from all of the 2D coverages that share the same horizontal, vertical, and temporal domains, there is a significant reduction in number of coverages, thus reducing complexity. This is a challenge, as the vertical and temporal axes are not regular and need to be enumerated. The “GeneralGridCoverage” as described in CIS1.1 makes this possible.

This key concept therefore changes the traditional view of data as being a set of 2D fields; each with a level, level type, parameter name, and forecast period. We can now describe the whole atmosphere as a multidimensional cube with properties, e.g. temperature, wind speed, and humidity. This results in the ability to make multidimensional (4/5/\* D) geospatial queries that are much more efficient, for both the WCS2.1 GetCoverage, GetPolygon and GetCorridor operations. It is important to note that there are special cases where the vertical axis has no vertical dependency (e.g. surface, max wind level). It is also possible that some parameters (e.g. temperature) may belong to more than one coverage (e.g. surface, isobaric, etc.)

* 1. Polygons:

A typical polygonal extraction pattern is show in Figure 3 using a set of points to describe the polygon. A special case of a polygon extraction is to use a circle with a radius and centre point position.

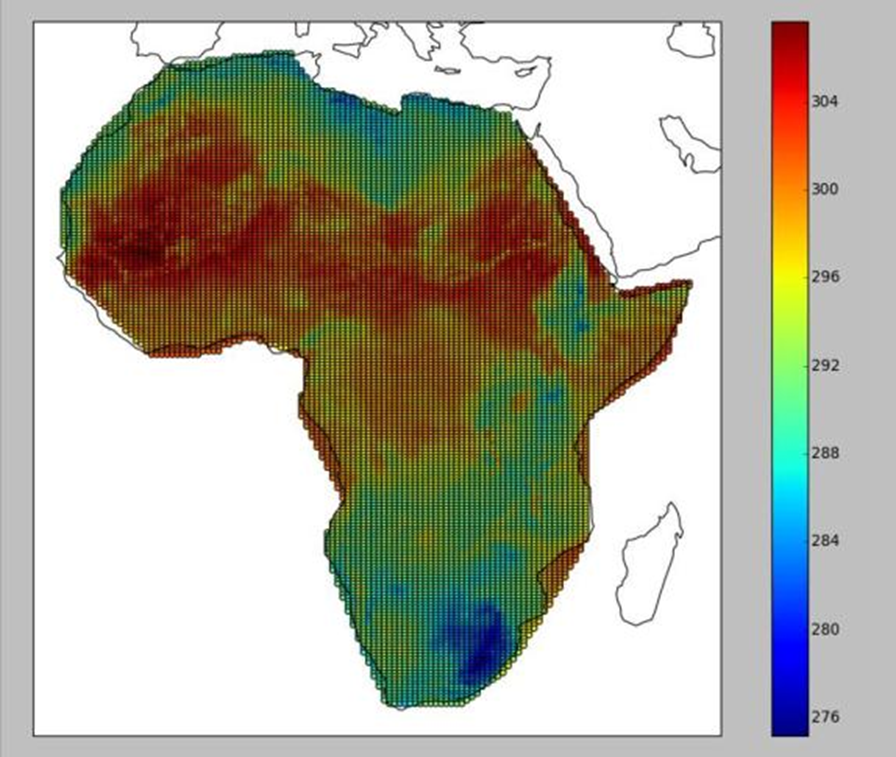


Figure 3 — Polygonal Subset (Extraction Pattern)

* + 1. Encoding

Encoding of the polygon in GML is likely to be very verbose and in such cases, a binary format such as NetCDF may be used. An alternative to GML e.g. JSON, will provide a very accessible encoding.

1. The core GetPolygon requirement (normative)
   1. Requirements class: GetPolygon

This clause establishes the GetPolygon extension core for conformance class getPolygon. Clients and servers supporting the requirements class support the extraction of a polygon from a multidimensional data cube. A UML diagram of class GetPolygon is shown in Figure 4.

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/WCS\_application\_profile\_metocean\_polygon/1.0/req/getPolygon** | |
| **Dependency** | **http:/www.opengis.net/spec/WCS/2.1/conf/core/getCoverage** |
| **Requirement** | **/req/getPolygon/structure**  A metoceanpolygon:GetPolygon instance **shall** conform to Figure 4 and Table 2 |
| **Requirement** | /**req/getPolygon/getCapabilities-response-conformance-class-in-profile**  A WCS service implementing this extension **shall** include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response:  <http://www.opengis.net/spec/WCS_application_profile_metocean_polygon/1.0/conf/getPolygon> |
| **Requirement** | **/req/getPolygon/request-valid-identifier**  The coverageId parameter value in a *GetPolygon* request **shall** be equal to the identifier of one of the coverages offered by the server addressed. |
| **Requirement** | **/req/getPolygon/acceptable-format**  If a *GetPolygon* request contains a format parameter then this parameter **shall** contain a MIME type identifier occurring in some WCS::formatSupported element of the response to a successful GetCapabilities request to this server. |
| **Requirement** | **/req/getPolygon/acceptable-mediaType**  If a *GetPolygon* request contains a mediaType parameter then this parameter **shall** contain a MIME type identifier of fixed value “multipart/related”. |
| **Requirement** | **/req/getPolygon/polygon-description**  The *GetPolygon* request **shall** contain a valid PolygonDescription element within the *GetPolygon* request element. |
| **Requirement** | **/req/getPolygon/range-component**  The parameter value of the RangeComponent of the wcs:RangeItem element **shall** contain a parameter that is part of the requested coverage. |
| **Requirement** | **/req/getPolygon/response-encoding**  The contents of the response to a successful *GetPolygon* request shall be en­cod­ed as specif­ied by the request format parameter, if this parameter is present, and in the coverage’s Native Format if this parameter is not present. |



Figure 4 – GetPolygon UML class diagram

* + 1. Requirements class overview

The GetPolygon requirements class defines the structure of the GetPolygon operation.

* + 1. Metoceanpolygon::GetPolygon

The new operation GetPolygon allows for the extraction of a polygon. The extra conformance classes are used to further define the possible options outlined here in the getPolygon conformance class. The GetPolygon operation is derived from wcs:GetCoverage and inherits the version and service elements. The GetPolygon properties are listed in Table 2.

Table 2 METOCEANPOLYGON::GetPolygon properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| metoceanpolygon:polygonDescription | The definition of the extraction pattern (polygon) to be used by the GetPolygon request. | metoceanpolygon:PolygonDescription | One  (mandatory) |
| wcs::coverageId | Identifier of a coverage offered by the service on hand | NCName | one (mandatory) |
| wcs: mediaType | If present, enforces a multipart encoding | anyURI, fixed to “multipart/related” | zero or one (optional) |
| wcs:format | MIME type identifier of the format in which the coverage returned is encoded | anyURI | zero or one (optional) |
| rsub:RangeSubset | Selection is based on the coverage’s range type definition where identifiable components are given; in the MetOcean domain, these take the form of defined parameters. | rsub:RangeItem. | One  (mandatory) |

* + 1. rsub::RangeSubset

The RangeSubset properties are listed in Table 3.

Table 3 RSUB::RangeSubset properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| rsub:RangeComponent | Range component name (parameter name) | string | one (mandatory) |

* 1. Requirements class: PolygonDescription
     1. Requirements class overview

This clause establishes the PolygonDescription conformance class. Clients and servers supporting the requirements class support the extraction of a polygon from a multidimensional data cube. PolygonDescription is mandatory. A UML diagram of class PolygonDescription is shown in Figure 5.

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/PolygonDescription** | |
| **Dependency** | **http:/www.opengis.net/spec/WCS/2.1/conf/core/getCoverage** |
| **Requirement** | /**req/getPolygon/PolygonDescription/structure**  A metoceanpolygon:PolygonDescription instance **shall** conform to Figure 5 and Table 4, Table 5, Table 6, Table 7, Table 8, Table 9, Table 10, Table 11, Table 12, Table 13, and Table 14. |
| **Requirement** | /**req/getPolygon/PolygonDescription/PolygonRing**  The PolygonRing element **shall** be derived from PolygonGeometry |
| **Requirement** | /**req/getPolygon/PolygonDescription/PolygonCircle**  The PolygonCircle element **shall** be derived from PolygonGeometry. |
| **Requirement** | /**req/getPolygon/PolygonDescription/PolygonGeometry**  The server **shall** support at leastone of the conformance classes i.e.PolygonDescription as per PolygonRing **(8.2.4) or** PolygonDescription as per PolygonCircle **(8.2.5)** |



Figure 5 – PolygonDescription UML class diagram

* + 1. Requirements class overview

The PolygonDescription requirements class defines the structure of the polygon as extracted by the GetPolygon operation.

* + 1. GetPolygon Description

The Polygon may be defined in one of two ways; either be a ring or a circle. The two methods are described as “per Ring” or “per Circle”. The GetPolygon properties are listed in Table 4.

Table 4 METOCEANPOLYGON::GetPolygon properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| metoceanpolygon:polygonDescription | The definition of the extraction pattern (polygon) to be used by the GetPolygon request. | metoceanpolygon:PolygonDescription | One  (mandatory) |

* + 1. PolygonDescription as per PolygonRing

In this clause the polygon will be defined as a set of points to form a ring. The PolygonDescription properties, as per PolygonRing, are listed in Table 5.

Table 5 METOCEANPOLYGON::PolygonDescription properties as per PolygonRing

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| metoceanpolygon:polygonGeometry | The geometry of the polygon defined as a set of points. | metoceanpolygon:PolygonRing | One  (mandatory) |

* + 1. PolygonDescription as per PolygonCircle

In this clause the polygon will be defined as a circle with centre point and radius. The PolygonDescription properties, as per PolygonCircle, are listed in Table 6.

Table 6 METOCEANPOLYGON::PolygonDescription properties as per PolygonCircle

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| metoceanpolygon:polygonGeometry | The geometry of the polygon defined as a circle. | metoceanpolygon:PolygonCircle | One  (mandatory) |

* + 1. PolygonRing

In this clause the polygon will be defined as a ring. The PolygonRing properties are listed in Table 7.

Table 7 METOCEANPOLYGON::PolygonRing properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| metoceanpolygon:polygon | The specification of the polygon as a Ring | gml:Polygon | One  (mandatory) |

* + 1. GML::Polygon as per PolygonRing

A Polygon is a special surface that is defined by a single surface. The boundary of this patch is coplanar and the polygon uses planar interpolation in its interior. The elements gml:exterior and gml:interior describe the surface boundary of the polygon. The Polygon properties, as per PolygonRing, are listed in Table 8.

Table 8 GML::Polygon properties as per PolygonRing

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| gml:exterior | The specification of the polygon Ring as per LinearRing | gml:LinearRing | One  (mandatory) |

* + 1. GML::LinearRing as per PolygonRing

A LinearRing is defined by four or more coordinate tuples, with linear interpolation between them; the first and last coordinates shall be coincident. The number of direct positions in the list shall be at least four. The LinearRing properties, as per PolygonRing, are listed in Table 9.

Table 9 GML::LinearRing properties as per PolygonRing

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| gml:posList | The specification of the polygon | gml:doubleList | One  (mandatory) |

* + 1. PolygonCircle

In this clause the polygon will be defined as a circle with centre point and radius. The PolygonCircle properties are listed in Table 10.

Table 10 METOCEANPOLYGON::PolygonCircle properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| metoceanpolygon:polygon | The specification of the polygon as a Circle | gml:Polygon | One  (mandatory |

* + 1. GML::Polygon as per PolygonCircle

A Polygon is a special surface that is defined by a single surface . The boundary of this patch is coplanar and the polygon uses planar interpolation in its interior. The elements exterior and interior describe the surface boundary of the polygon. The Polygon properties, as per PolygonCircle, are listed in Table 11.

Table 11 GML::Polygon properties as per PolygonCircle

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| gml:exterior | The specification of the polygon Circle | gml: Ring | One  (mandatory) |

* + 1. GML::Ring

A ring is used to represent a single connected component of a surface boundary as specified in ISO 19107:2003, 6.3.6. Every gml:curveMember references or contains one gml:Curve. In the context of a ring, a curve describes the boundary of the surface. The curve is then described by referencing one or more gml:segments. The Ring properties are listed in Table 12.

Table 12 GML::Ring

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| gml:curveMember | Defines the polygon as a set of connected curves | gml: Curve | One or more  (mandatory) |

* + 1. GML::Curve

A curve is composed of one or more curve segments. The element gml:segments encapsulates the segments of the curve. The Curve properties are listed in Table 13.

Table 13 GML::Curve

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| gml:segments | A special case where the segment defines a circle | gml: CircleByCenterPoint | One or more  (mandatory) |

* + 1. GML:: CircleByCenterPoint

A gml:CircleByCenterPoint is a gml:ArcByCenterPoint with identical start and end angle to form a full circle. This representation can be used only in 2D. The CircleByCenterPoint properties are listed in Table 14.

Table 14 GML:: CircleByCenterPoint

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| gml:pos | The centre point of the circle. | gml:doubleList | One  (mandatory) |
| gml:radius | The radius of the circle. | gml:double | One  (mandatory) |

* 1. Requirements class: VerticalTemporalDescription

This clause establishes the VerticalTemporalDescription conformance class. Clients and servers supporting the requirements class support the extraction of a polygon from a multidimensional data cube. The vertical and temporal extents represent the third and fourth dimensions of that cube. As stated in Section 7.3.1, the vertical and temporal axes are not regular and thus need to be enumerated. This necessitates using the “GeneralGridCoverage” as described in CIS1.1, which makes this enumeration possible. A UML diagram of class VerticalTemporalDescription is shown in Figure 6.

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/VerticalTemporalDescription** | |
| **Dependency** | **http:/www.opengis.net/spec/WCS/2.1/conf/core/getCoverage** |
| **Dependency** | **http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf** |
| **Dependency** | **http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf** |
| **Requirement** | **/req/getPolygon/VerticalTemporalDescription/structure**  Ametoceanpolygon:VerticalTemporalDescriptioninstance **shall** conform to Figure 6 and Table 15 |
| **Requirement** | /**req/getPolygon/PolygonDescription/PolygonGeometry**  The server **shall** support at leastone of the conformance classes i.e[SubSetByTrim](http://www.opengis.net/spec/WCS_application-profile_metocean_polygon/1.0/req/getPolygon/SubSetByTrim) and SubsetByInterpolation |



Figure 6 – VerticalTemporalDescription UML class diagram

* + 1. Requirements class overview

The polygon extraction pattern uses a 2D polygon that is replicated over selected levels in the vertical dimension and selected times in the temporal dimension.

* + 1. VerticalTemporalDescription Properties

There are two methods of selecting the levels and times: subset by interpolation and subset by trim. These methods are described in sections 8.4 and 8.5. The VerticalTemporalDescription properties are listed in Table 15.

Table 15 METOCEANPOLYGON:: VerticalTemporalDescription properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| subsetByInterpolation |  | SubsetByInterpolation | Zero or one (optional) |
| subsetByTrim |  | SubsetByTrim | Zero or one (optional) |

* 1. Requirements class: SubsetByTrim

This clause establishes the SubsetByTrim conformance class. Clients and servers supporting the requirements class support the extraction of a polygon from a multidimensional data cube that has a vertical and temporal extent. A UML diagram of class SubSetByTrim is shown in Figure 7.

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/SubsetByTrim** | |
| **Dependency** | **http:/www.opengis.net/spec/WCS/2.1/conf/core/getCoverage** |
| **Dependency** | **http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf** |
| **Dependency** | **http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf** |
| **Requirement** | **/req/getPolygon/SubsetByTrim/structure**  Ametoceanpolygon:SubesetByTrim instance **shall** conform to Figure 7, Table 16 and Table 17 |
| **Requirement** | **/req/getPolygon/SubsetByTrim/request-valid-dimension** Every dimension value in a GetPolygon Trim request shall be equal to one of the axis­Lab­els dimension names specified in the cover­age’s domain set, unless the server offers a WCS CRS extension which overrides this re­qui­rement. |
| **Requirement** | **/req/getPolygon/request-no-duplicate-dimension** A GetPolygonrequest **shall** contain at most one subsetting operation for each of the dimen­sions of the coverage addressed. |
| **Requirement** | **/req/getPolygon/SubsetByTrim/Polygon-trim-within-extent**  Let the extent of the coverage’s CIS::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 ≤ trimLow ≤ trimHigh ≤ H |



Figure 7 – SubsetByTrim UML class diagram

* + 1. Requirements class overview

The polygon extraction pattern uses a 2D polygon, that is replicated over selected levels in the vertical dimension and selected times in the temporal dimension. These levels and times are selected by “trimming” the MetOcean cube in the vertical and temporal dimensions.

* + 1. SubsetByTrim Properties

The limits define the upper and lower times and levels to be extracted from the parent MetOcean cube. The extraction, as per the polygon, is only extracted on levels and times from the MetOcean cube that lie between the upper and lower limits. The SubsetByTrim properties are listed in Table 16.

Table 16 METOCEANPOLYGON::SubsetByTrim properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| dimensionTrim | Subsetting specifications, at most one per subsetting dimension | DimensionTrim | One  (mandatory |

* + 1. DimensionTrim Properties

The DimensionTrim properties are listed in Table 17.

Table 17 METOCEANPOLYGON::DimensionTrim properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| dimension | Name of dimension along which to subset | string | One or more  (mandatory) |
| trimLow | Lower bound of cutout along dimension | string | One  (mandatory |
| trimHigh | Upper bound of cutout along dimension | string | One  (mandatory |

* + 1. An example an encoding

<?xml version="1.0" encoding="UTF-8"?>  
<metoceanpolygon:GetPolygon xmlns:gml="http://www.opengis.net/gml/3.2"">  
 <wcs:CoverageId>CoverageId0</wcs:CoverageId>  
 <rsub:RangeSubset>  
 <rsub:RangeItem>  
 <rsub:RangeComponent>Temperature</rsub:RangeComponent>  
 </rsub:RangeItem>  
 <rsub:RangeItem>  
 <rsub:RangeComponent>Wind\_Speed</rsub:RangeComponent>  
 </rsub:RangeItem>  
 <rsub:RangeItem>  
 <rsub:RangeComponent>Wind\_Direction</rsub:RangeComponent>  
 </rsub:RangeItem>  
 </rsub:RangeSubset>  
 <metoceanpolygon:polygonDescription>  
 <metoceanpolygon:PolygonDescription>  
 <metoceanpolygon:polygonGeometry>  
 <metoceanpolygon:PolygonRing gml:id="ID000">  
 <gml:Polygon gml:id="Model\_Boundary-Geometry"  
 uomLabels="deg deg" axisLabels="lat Lon"  
 srsDimension="2"  
 srsName="http://www.opengis.net/def/crs/EPSG/0/4326">  
 <gml:exterior>  
 <gml:LinearRing>  
 <gml:posList> -90.0 -180.0 90.0 -180.0 90.0 180.0  
 -90.0 180.0 -90.0 -180.0 </gml:posList>  
 </gml:LinearRing>  
 </gml:exterior>  
 </gml:Polygon>  
 </metoceanpolygon:PolygonRing>  
 </metoceanpolygon:polygonGeometry>  
 <metoceanpolygon:verticaTemporalDescription>  
 <metoceanpolygon:VerticalTemporalDescription>  
 <metoceanpolygon:subsetByTrim>  
 <metoceanpolygon:SubsetByTrim>  
 <metoceanpolygon:dimensionTrim>  
 <metoceanpolygon:DimensionTrim uomLabel="hPA">  
 <metoceanpolygon:dimension>Pressure</metoceanpolygon:dimension>  
 <metoceanpolygon:trimLow>200.0</metoceanpolygon:trimLow>  
 <metoceanpolygon:trimHigh>1000.0</metoceanpolygon:trimHigh>  
 </metoceanpolygon:DimensionTrim>  
 </metoceanpolygon:dimensionTrim>  
 <metoceanpolygon:dimensionTrim>  
 <metoceanpolygon:DimensionTrim uomLabel="ISO8601">  
 <metoceanpolygon:dimension>Time</metoceanpolygon:dimension>  
 <metoceanpolygon:trimLow>2017-05-14T00:00:00Z</metoceanpolygon:trimLow>  
 <metoceanpolygon:trimHigh>2017-05-15T00:00:00Z</metoceanpolygon:trimHigh>  
 </metoceanpolygon:DimensionTrim>  
 </metoceanpolygon:dimensionTrim>  
 </metoceanpolygon:SubsetByTrim>  
 </metoceanpolygon:subsetByTrim>  
 </metoceanpolygon:VerticalTemporalDescription>  
 </metoceanpolygon:verticaTemporalDescription>  
 </metoceanpolygon:PolygonDescription>  
 </metoceanpolygon:polygonDescription>  
</metoceanpolygon:GetPolygon>

* 1. Requirements class: SubsetByInterpolation

This clause establishes the SubsetByInterpolation conformance class. Clients and servers supporting the requirements class support the extraction of a polygon from a multidimensional data cube that has a vertical and temporal extent by interpolating the data to a defined multidimensional grid. A UML diagram showing the SubsetByInterpolation class is shown in Figure 8.

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/WCS\_application-profile\_metocean\_polygon/1.0/req/getPolygon/SubsetByInterpolation** | |
| **Dependency** | **http://www.opengis.net/spec/WCS\_application\_profile\_metocean\_polygon/1.0/req/getPolygon** |
| **Requirement** | **/req/getPolygon/SubsetByInterpolation/structure**  Ametoceanpolygon:VerticalTemporalDescriptioninstance **shall** conform to Figure 8 and Table 18, Table 19,Table 20,Table 21,Table 22,Table 23Table 24 andTable 25 |
| **Requirement** | **/req/getPolygon/SubsetByInterpolation/grid-order**  In a coverage using the *grid-irregular* scheme, the directPosition values in any CIS::IrregularAxis **shall** be listed in strictly monotonic order, expressed in the units of measure of this axis as defined in the CRS identified in the srsName item of the envelope. |
| **Requirement** | **/req/getPolygon/SubsetByInterpolation/grid-order**  In a coverage using the *grid-irregular* scheme, for any two CIS::DisplacementAxis­Nest elements their set of axis names **shall** be disjoint.  All combinations of axis types index and regular (from class *grid-regular*) as well as irregular and displaced (from class *grid-irregular*) are permitted. However, no two axes may have the same name (i.e., axis label). |



Figure 8 – SubsetByInterpolation UML class diagram

* + 1. Requirements class overview

This conformance class describes how a grid (target grid) is defined onto which the data contained within the polygon is extracted by interpolation. The “target” grid is only defined in the vertical and temporal dimensions as there is no interpolation in the horizontal. Thus the grid points will not be interpolated in the horizontal domain, but will be in the vertical and temporal dimensions.

* + 1. SubsetByInterpolation Properties

The SubsetByInterpolation properties are shown in Table 18.

Table 18 METOCEANPOLYGON::SubsetByInterpolation properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| interpolation | Interpolation method to be applied to the named axis GetPolygon result preparation | int:IntInterpolation | One  (mandatory) |
| generalGrid |  | cis:GeneralGrid | One  (mandatory |

* + 1. Interpolation Properties

The Interpolation properties are listed in Table 19.

Table 19 INT:: Interpolation properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| int:globalInterpolation | The default Interpolation method to be used | int:IntInterpolationPerAxis | One  (mandatory |

* + 1. InterpolationPerAxis Properties

The InterpolationPerAxis properties are listed in Table 20.

Table 20 INT:: InterpolationPerAxis Properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| int:axis | The axis for which the interpolation Method will be applied | anyURI | One  (mandatory |
| interpolationMethod | The interpolation method. | anyURI | One  (mandatory |

* + 1. CIS::GeneralGrid Properties

Describing the direct positions in multi-dimensional coordinate space, depending on the type of grid. The GeneralGrid properties are listed in Table 21.

Table 21 CIS:: GeneralGrid properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| cis:srsName | URL identifying the Index CRS of the domain set grid array in this coverage | anyURI | One  (mandatory) |
| cis:axisLabels | Axes involved in the “nest” of displaced direct positions; these axes shall form a subset of the CIS::General­Grid axisLabels | NCName | One or more (mandatory) |
| cis:GridLimits | In addition, the limits of the underlying array are given by the CIS::gridLimits component | cis:IndexAxis | Zero or one (optional) |
| cis:Axis | grid axis identifiers, all distinct within a grid | cis:axisLabel | One or more  (mandatory) |

* + 1. GridLimits Properties

The grid limits in the CIS::Axis structure contains information about the grid boundaries in the coverage’s CRS The GridLimits properties are listed in Table 22.

Table 22 CIS:: GridLimits properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| srsName | URL identifying the Index CRS of the domain set grid array in this coverage | anyURI | One  (mandatory) |
| cis:indexAxis | all axes of the Index CRS referenced in srsName, in proper sequence | CIS:: IndexAxis | One or more (mandatory) |
| axisLabels | Axes involved in the “nest” of displaced direct positions; these axes shall form a subset of the CIS::General­Grid axisLabels | string | One or more (mandatory) |

* + 1. Axis Properties

The Axis properties are listed in Table 23.

Table 23 CIS::Axis properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| axisLabel | identifier of this axis | string | One |

* + 1. IndexAxis Properties

The IndexAxis properties are listed in Table 24.

Table 24 CIS::IndexAxis properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| lowerBound | Lowest array coordinate along this axis | integer | One  (mandatory) |
| upperBound | Highest array coordinate along this axis | integer | One  (mandatory) |

* + 1. RegularAxis Properties

The RegularAxis properties are listed in Table 25.

Table 25 CIS::RegularAxis properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| lowerBound | Lowest coordinate along this grid axis | string | One  (mandatory) |
| upperBound | Highest coordinate along this axis | string | One  (mandatory) |
| resolution | grid resolution along this axis | string | One  (mandatory) |
| cis:axisLabel | Shorthand axis identifier with scope given by the coverage document | string | One  (mandatory) |
| cis:uomLabel | Shorthand identifier of the Unit of Measure used on this axis (as indicated in the CRS definition for this axis) | string | One  (mandatory) |

* + 1. IrregularAxis Properties

The IrregularAxis properties are listed in Table 25.

Table 25 CIS::IrregularAxis properties

| **Name** | **Definition** | **Data types and values** | **Multiplicity** |
| --- | --- | --- | --- |
| direct­Positions | Ordered sequence of direct positions along this axis | cis::Direct­PositionType | One or more (mandatory) |
| uomLabel | unit of measure in which values along this axis are expressed | string | One (mandatory) |

* + 1. An example an encoding

<?xml version="1.0" encoding="UTF-8"?>  
<metoceanpolygon:GetPolygon xmlns:cis="http://www.opengis.net/cis/1.1/gml"  
 xmlns:gml="http://www.opengis.net/gml/3.2" xmlns:xlink="http://www.w3.org/1999/xlink"  
 xmlns:swe="http://www.opengis.net/swe/2.0" xmlns:ows="http://www.opengis.net/ows/2.0"  
 xmlns:gco="http://www.isotc211.org/2005/gco" xmlns:gmd="http://www.isotc211.org/2005/gmd"  
 xmlns:sml="http://www.opengis.net/sensorml/2.0" xmlns:gmlcov="http://www.opengis.net/gmlcov/1.0"  
 xmlns:om="http://www.opengis.net/om/2.0" xmlns:sam="http://www.opengis.net/sampling/2.0"  
 xmlns:sams="http://www.opengis.net/samplingSpatial/2.0"  
 xmlns:gts="http://www.isotc211.org/2005/gts"  
 xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"  
 xmlns:gsr="http://www.isotc211.org/2005/gsr" xmlns:ns0="http://www.opengis.net/wcs/2.0"  
 xmlns:gss="http://www.isotc211.org/2005/gss"  
 xmlns:metocean="http://www.opengis.net/wcs/metoceanProfile/1.0"  
 xmlns:metoceanpolygon="http://www.opengis.net/wcs/metoceanProfile\_getPolygon/1.0"  
 xmlns:wcs="http://www.opengis.net/wcs/2.1" xmlns:gmlrgrid="http://www.opengis.net/gml/3.3/rgrid"  
 xmlns:int="http://www.opengis.net/WCS\_service-extension\_interpolation/1.0"  
 xmlns:scal="http://www.opengis.net/WCS\_service-extension\_scaling/1.0"  
 xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"  
 xsi:schemaLocation="http://www.opengis.net/wcs/metoceanProfile\_getPolygon/1.0 file:/C:/Users/PTrevelyan/WCS/MetOceanWCS/Polygons/Schemas/wcsMetOceanGetPolygonV3.xsd"  
 service="WCS" version="2.1.0">  
 <wcs:CoverageId>CoverageId0</wcs:CoverageId>  
 <rsub:RangeSubset>  
 <rsub:RangeItem>  
 <rsub:RangeComponent>Temperature</rsub:RangeComponent>  
 </rsub:RangeItem>  
 <rsub:RangeItem>  
 <rsub:RangeComponent>Wind\_Speed</rsub:RangeComponent>  
 </rsub:RangeItem>  
 <rsub:RangeItem>  
 <rsub:RangeComponent>Wind\_Direction</rsub:RangeComponent>  
 </rsub:RangeItem>  
 </rsub:RangeSubset>  
 <metoceanpolygon:polygonDescription>  
 <metoceanpolygon:PolygonDescription>  
 <metoceanpolygon:polygonGeometry>  
 <metoceanpolygon:PolygonCircle gml:id="ID000">  
 <gml:Polygon gml:id="ID001">  
 <gml:exterior>  
 <gml:Ring>  
 <gml:curveMember>  
 <gml:Curve gml:id="curve01">  
 <gml:segments>  
 <gml:CircleByCenterPoint numArc="1">  
 <gml:pos>27.10 -73.10</gml:pos>  
 <gml:radius uom="[nmi\_i]">250</gml:radius>  
 </gml:CircleByCenterPoint>  
 </gml:segments>  
 </gml:Curve>  
 </gml:curveMember>  
 </gml:Ring>  
 </gml:exterior>  
 </gml:Polygon>  
 </metoceanpolygon:PolygonCircle>  
 </metoceanpolygon:polygonGeometry>  
 <metoceanpolygon:verticaTemporalDescription>  
 <metoceanpolygon:VerticalTemporalDescription>  
 <metoceanpolygon:subsetByInterpolation>  
 <metoceanpolygon:SubsetByInterpolation>  
 <metoceanpolygon:interpolation>  
 <int:Interpolation>  
 <int:globalInterpolation>http://www.opengis.net/def/interpolation/OGC/1/linear</int:globalInterpolation>  
 <int:InterpolationPerAxis>  
 <int:axis>Time</int:axis>  
 <int:interpolationMethod>http://www.opengis.net/def/interpolation/OGC/1/linear</int:interpolationMethod>  
 </int:InterpolationPerAxis>  
 <int:InterpolationPerAxis>  
 <int:axis>pressure</int:axis>  
 <int:interpolationMethod>http://www.opengis.net/def/interpolation/OGC/1/barycentric</int:interpolationMethod>  
 </int:InterpolationPerAxis>  
 </int:Interpolation>  
 </metoceanpolygon:interpolation>  
   
 <metoceanpolygon:generalGrid>  
 <cis:GeneralGrid srsName="http://www.opengis.net/def/crs-compound?  
 1=http://http://www.opengis.net/def/crs/OGC/0/Time&amp;  
 2=http://www.codes.wmo.int/GRIB2/table4.5/IsobaricSurface"   
 axisLabels="Time pressure">  
 <cis:IrregularAxis uomLabel="ISO8601" axisLabel="Time">  
 <cis:C>PT0H</cis:C>  
 <cis:C>PT6H</cis:C>  
 <cis:C>PT12H</cis:C>  
 <cis:C>PT18H</cis:C>  
 <cis:C>PT24H</cis:C>  
 <cis:C>PT30H</cis:C>  
 <cis:C>PT36H</cis:C>  
 <cis:C>PT42H</cis:C>  
 <cis:C>PT48H</cis:C>  
 </cis:IrregularAxis>  
   
 <cis:IrregularAxis uomLabel="hPa" axisLabel="pressure">  
 <cis:C>1000.0</cis:C>  
 <cis:C>950.0</cis:C>  
 <cis:C>850.0</cis:C>  
 <cis:C>500.0</cis:C>  
 <cis:C>300.0</cis:C>  
 <cis:C>250.0</cis:C>  
 <cis:C>200.0</cis:C>  
 </cis:IrregularAxis>  
   
 <cis:GridLimits srsName="http://www.codes.wmo.int/def/crs/OGC/0/Index4D" axisLabels="k l" >  
 <cis:IndexAxis axisLabel="k" lowerBound="0" upperBound="8"/>  
 <cis:IndexAxis axisLabel="l" lowerBound="0" upperBound="6"/>  
 </cis:GridLimits>  
 </cis:GeneralGrid>  
 </metoceanpolygon:generalGrid>  
 </metoceanpolygon:SubsetByInterpolation>  
 </metoceanpolygon:subsetByInterpolation>  
 </metoceanpolygon:VerticalTemporalDescription>  
 </metoceanpolygon:verticaTemporalDescription>  
 </metoceanpolygon:PolygonDescription>  
 </metoceanpolygon:polygonDescription>  
</metoceanpolygon:GetPolygon>

1. : Conformance Class Abstract Test Suite (normative)
   1. Conformance class: getPolygon

|  |  |  |  |
| --- | --- | --- | --- |
| **Conformance Class** | | | |
| http://www.opengis.net/spec/WCS\_application\_profile\_metocean\_Polygon/1.0/conf/getPolygon | | | |
| **Requirements** | http://www.opengis.net/spec/WCS\_application\_profile\_metocean\_Polygon/1.0/req/getPolygon | | |
| **Dependency** |  | | |
| **Test** | /**conf/getPolygon/structure** | | |
| **Requirement** | |  |
| **Test purpose** | |  |
|  | **Test method** | |  |
| **Test Type** | |  |
| **Test** | /**conf/getPolygon/getCapabilities-response-conformance-class-in-profile** | | |
|  | **Requirement** | /**req/getPolygon/getCapabilities-response-conformance-class-in-profile** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test Type** |  | |
| **Test** | **/conf/getPolygon/request-valid-identifier** | | |
|  | **Requirement** | **/req/getPolygon/request-valid-identifier** | |
|  | **Test purpose** | If the MetOceanObservation contains the optional resultQuality property (from OM\_Observation) then that element **shall** reference the ResultMask by substitution with AbstractDQ\_Element. | |
|  | **Test method** | Retrieve a CoverageDescriptions document via a DescribeCoverage operation. Inspect the MetOceanObservation component and validate that the element “om:resultQuality” (if not null) property contains the element metocean:ResultMask. | |
|  | **Test type** | Conformance | |
| **Test** | **/conf/getPolygon/request-valid-identifier** | | |
|  | **Requirement** | **/req/getPolygon/request-valid-identifier** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | **/conf/getPolygon/acceptable-format** | | |
|  | **Requirement** | **/req/getPolygon/acceptable-format** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | **/conf/getPolygon/acceptable-mediaType** | | |
|  | **Requirement** | **/req/getPolygon/acceptable-mediaType** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | **/conf/getPolygon/path** | | |
|  | **Requirement** | **/req/getPolygon/path** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | **/conf/getPolygon/range-component** | | |
|  | **Requirement** | **/req/getPolygon/range-component** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | **/conf/getPolygon/response-encoding** | | |
|  | **Requirement** | **/req/getPolygon/response-encoding** | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | / | | |
|  | **Requirement** |  | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |
| **Test** | / | | |
|  | **Requirement** |  | |
|  | **Test purpose** |  | |
|  | **Test method** |  | |
|  | **Test type** | Conformance | |

* 1. Conformance class: PathDescription
  2. Conformance class: PolygonExtent
  3. Conformance class: PolygonExtractionMethod

1. [↑](#footnote-ref-1)