

**OGC® DOCUMENT: 15-108R3**

External identifier of this OGC® document: [http://www.opengis.net/doc/IS/WCS\\_application-profile\\_metocean\\_corridor/1.0](http://www.opengis.net/doc/IS/WCS_application-profile_metocean_corridor/1.0)

**OGC®**  
Making location count.

# OGC METOCEAN APPLICATION PROFILE FOR WCS2.1: PART 1 METOCEAN GETCORRIDOR EXTENSION

---

**STANDARD  
Implementation**

**APPROVED**

**Version:** 1.0

**Submission Date:** 2017-08-19

**Approval Date:** 2019-02-04

**Publication Date:** 2021-03-22

**Editor:** Peter Trevelyan, Paul Hershberg, Steve Olson

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

## License Agreement

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

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

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

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

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

None of the Intellectual Property or underlying information or technology may be downloaded or otherwise exported or reexported in violation of U.S. export laws and regulations. In addition, you are responsible for complying with any local laws in your jurisdiction which may impact your right to import, export or use the Intellectual Property, and you represent that you have complied with any regulations or registration procedures required by applicable law to make this license enforceable.

Suggested additions, changes and comments on this standard are welcome and encouraged. Such suggestions may be submitted using the online change request form on OGC web site: [http://portal.opengeospatial.org/public\\_ogc/change\\_request.php](http://portal.opengeospatial.org/public_ogc/change_request.php)

## Copyright notice

Copyright © 2021 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.ogc.org/legal/>

## Note

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.

# CONTENTS

I. ABSTRACT .....	vi
II. KEYWORDS .....	vi
III. PREFACE .....	vii
IV. SECURITY CONSIDERATIONS .....	viii
V. SUBMITTING ORGANIZATIONS .....	ix
VI. SUBMITTERS .....	ix
1. SCOPE .....	2
2. CONFORMANCE .....	4
3. NORMATIVE REFERENCES .....	7
4. TERMS AND DEFINITIONS .....	10
5. CONVENTIONS .....	14
5.1. Abbreviated terms .....	14
5.2. Schema language .....	14
5.3. UML notation .....	15
6. VOCABULARIES .....	17
7. NON-NORMATIVE (INFORMATIVE) MATERIAL .....	19
7.1. WCS2.1 .....	21
7.2. Key Concepts .....	22
7.3. Coverages .....	23
7.4. Time Dependant data (from WMS Best Practice OGC document:12-111r1) .....	24
7.5. Trajectories and corridors: .....	26
7.6. Methods of extraction .....	31
8. THE CORE GETCORRIDOR REQUIREMENT (NORMATIVE) .....	43
8.1. Requirements class: GetCorridor .....	43
8.2. Requirements class: PathDescription .....	47
8.3. Requirements class: CorridorExtent .....	52
8.4. Requirements class: CorridorExtractionMethod .....	59
8.5. Requirements Class: GetCorridor-post-xml .....	66

8.6. Requirements Class: GetCorridor-simple .....	67
8.7. Requirements Class: GetCorridor-simple-kvp .....	67
<b>ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE</b>	
(NORMATIVE) .....	70
A.1. Conformance class: GetCorridor .....	70
A.2. Conformance class: PathDescription .....	73
A.3. Conformance class: CorridorExtent .....	75
A.4. Conformance class: CorridorExtractionMethod .....	77
A.5. Conformance class: GetCorridor-post-xml .....	81
A.6. Conformance class: GetCorridor-simple .....	82
A.7. Conformance class: GetCorridor-simple-get-kvp .....	83
<b>ANNEX B (INFORMATIVE) REVISION HISTORY</b> .....	86
<b>BIBLIOGRAPHY</b> .....	88

## LIST OF TABLES

---

Table 1 – Summary of vocabularies within this standard .....	17
Table 2 – METOCEANCORRIDOR::GetCorridor properties .....	45
Table 3 – RSUB::RangeSubset properties .....	46
Table 4 – RSUB::Rangeltem properties .....	46
Table 5 – METOCEANCORRIDOR::PathDescription properties .....	48
Table 6 – METOCEANCORRIDOR::SegmentDefinition properties .....	49
Table 7 – METOCEANCORRIDOR::cis:V properties .....	50
Table 8 – METOCEANCORRIDOR::DiscretisationMethodValue Properties .....	50
Table 9 – CIS::DisplacementAxisNest .....	50
Table 10 – CIS::GridLimits .....	51
Table 11 – METOCEANCORRIDOR::CorridorExtent Properties .....	57
Table 12 – CIS::AxisExtent Properties .....	58
Table 13 – SCAL::Scaling .....	58
Table 14 – SCAL::ScaleToSize Properties .....	59
Table 15 – METOCEANCORRIDOR::CorridorExtractionMethod properties .....	61
Table 16 – METOCEANCORRIDOR::CorridorOrientationValue Properties .....	62
Table 17 – METOCEANCORRIDOR::CorridorExtractionMethodValue Properties .....	62
Table 18 – INT::Interpolation Properties .....	63
Table 19 – INT::InterpolationPerAxis .....	63
Table 20 – METOCEANCORRIDOR::GetCorridor request URL encoding .....	68

# LIST OF FIGURES

---

Figure 1 – WCS GetCoverage operation UML class diagram .....	20
Figure 2 – UML Diagram representing the coverage model (CIS 1.1) .....	22
Figure 3 – A typical flight path .....	26
Figure 4 – way points .....	28
Figure 5 – Segments equal in distance .....	28
Figure 6 – Segments equal in Time .....	29
Figure 7 – Discrete Segments equal in distance .....	29
Figure 8 – Discrete Segments equal in time .....	30
Figure 9 – Corridor aligned with trajectory axis .....	30
Figure 10 – Alternative view of the corridor aligned with trajectory axis .....	31
Figure 11 – Corridor vertical aligned perpendicular to the trajectory .....	31
Figure 12 – Extraction of the trajectory using grid-box semantics .....	32
Figure 13 – A 'Voxel' view of the trajectory .....	33
Figure 14 – Extraction of the corridor using grid-box semantics .....	34
Figure 15 – A "Voxel" view of the corridor .....	34
Figure 16 – Extraction of a corridor, using point based semantics .....	35
Figure 17 – Extraction of a corridor, using point based semantics .....	36
Figure 18 – Trajectory with point based semantics .....	37
Figure 19 – Trajectory with point based semantics .....	37
Figure 20 – Corridor grid with vertical aligned with grid .....	38
Figure 21 – Corridor grid with vertical aligned with grid .....	39
Figure 22 – Corridor grid perpendicular to the trajectory .....	40
Figure 23 – Corridor grid perpendicular to the trajectory .....	40
Figure 24 – GetCorridor UML class diagram .....	44
Figure 25 – CorridorPath UML class diagram .....	48
Figure 26 – CorridorExtent UML class diagram .....	54
Figure 27 – CorridorExtractionMethod UML class diagram .....	61



## ABSTRACT

---

This document defines an extension to WCS2.1, namely the extraction of data along a corridor defined by a path and corridor extent consisting of an information model and an XML encoding for the following two operations:

- a) **GetCapabilities** – a WCS function that describes the services and operations via a GetCapabilities document.
- b) **GetCorridor** – a WCS function that supports this operation to extract data from a multidimensional cube along a path, or corridor.

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.



## KEYWORDS

---

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

ogcdoc, OGC document, WCS, coverage, collection, meteorology, oceanography, NWP, analysis, result mask, observation, measurement, simulation, O&M, trajectory, corridor and MetOcean



## 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.



## SECURITY CONSIDERATIONS

---

No security considerations have been made for this standard.





## SUBMITTING ORGANIZATIONS

---

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

- Met Office, UK
- NOAA's National Weather Service



## SUBMITTERS

---

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

NAME	AFFILIATION
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

# SCOPE

---

# 1

## SCOPE

---

The purpose of the GetCorridor operation is to extract a corridor based on a trajectory from a multidimensional coverage. The need for the getCorridor 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.

2

# CONFORMANCE

---

## CONFORMANCE

---

This standard defines:

- An amended GetCapabilities operation response that will list the GetTrajectory operation and specify the token in the Sections element of the GetCapabilities request.
- A new operation “GetCorridor” that is used to extract data from a multidimensional cube along a path or trajectory.
- The conformance classes that describe the GetCorridor 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<sup>1</sup>.

In order to conform to this OGC™ interface standard, a software implementation shall choose to implement: <http://cite.opengeospatial.org/>

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\\_corridor/1.0/](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/)

This document establishes the following requirements and conformance classes:

- **GetCorridor** of URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor) defining **getCorridor** at a conceptual level in clause 8.1.

The corresponding conformance class is **getCorridor** with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/getCorridor](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor) See A.1

- **PathDescription** of [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/PathDescription](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/PathDescription) defining the **PathDescription** at a conceptual level in clause 8.2

The corresponding conformance class is **PathDescription** with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/PathDescription](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/PathDescription) . See A.2

---

<sup>1</sup>[https://portal.opengeospatial.org/files/?artifact\\_id=55234](https://portal.opengeospatial.org/files/?artifact_id=55234)

- **CorridorExtent** of URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/CorridorExtent](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtent) defining the **CorridorExtent** at a conceptual level in clause 8.3;

The corresponding conformance class is **CorridorExtent** with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/CorridorExtent](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtent) . See A.3

- **CorridorExtractionMethod** of URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/CorridorExtractionMethod](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtractionMethod) defining the **CorridorExtractionMethod** at a conceptual level in clause 8.4;

The corresponding conformance class is **CorridorExtractionMethod** with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/CorridorExtractionMethod](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod) . See A.4

- **GetCorridor-post-xml** of URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-post-xml](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-post-xml) defining GetCorridor-post-xml on the conceptual level in clause 8.5

The corresponding conformance class is offering with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/getCorridor-post-xml](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-post-xml) . See A.5

- **GetCorridor-simple** of URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-simple](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple) defining GetCorridor-simple on the conceptual level in clause 8.6

The corresponding conformance class is offering with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/getCorridor-simple](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple) See A.6

- **GetCorridor-simple-kvp** of URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-simple-get-kvp](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple-get-kvp) defining GetCorridor-simple on the conceptual level in clause 8.7

The corresponding conformance class is offering with URI [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/getCorridor-simple-get-kvp](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple-get-kvp) See A.7

3

# NORMATIVE REFERENCES

---

## NORMATIVE REFERENCES

---

The following documents are referred to in the text in such a way that some or all of their content constitutes requirements of this document. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

Policy SWG: OGC 08-131r3, *The Specification Model – Standard for Modular specifications*. Open Geospatial Consortium (2009). [https://portal.ogc.org/files/?artifact\\_id=34762&version=2](https://portal.ogc.org/files/?artifact_id=34762&version=2)

ISO: ISO/TS 19103:2005, *Geographic information – Conceptual schema language*. International Organization for Standardization, Geneva (2005). <https://www.iso.org/standard/37800.html>

ISO: ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*. International Organization for Standardization, Geneva (2004). <https://www.iso.org/standard/40874.html>

ISO: ISO 19107:2019, *Geographic information – Spatial schema*. International Organization for Standardization, Geneva (2019). <https://www.iso.org/standard/66175.html>

ISO: ISO 19111:2007, *Geographic information – Spatial referencing by coordinates*. International Organization for Standardization, Geneva (2007). <https://www.iso.org/standard/41126.html>

ISO: ISO 19123:2005, *Geographic information – Schema for coverage geometry and functions*. International Organization for Standardization, Geneva (2005). <https://www.iso.org/standard/40121.html>

ISO: ISO 19115-1:2014, *Geographic information – Metadata – Part 1: Fundamentals*. International Organization for Standardization, Geneva (2014). <https://www.iso.org/standard/53798.html>

ISO: ISO 19156:2011, *Geographic information – Observations and measurements*. International Organization for Standardization, Geneva (2011). <https://www.iso.org/standard/32574.html>

ISO: ISO 19136:2007, *Geographic information – Geography Markup Language (GML)*. International Organization for Standardization, Geneva (2007). <https://www.iso.org/standard/32554.html>

Peter Baumann: OGC 17-089r1, *OGC Web Coverage Service (WCS) 2.1 Interface Standard – Core*. Open Geospatial Consortium (2018). <http://docs.opengeospatial.org/is/17-089r1/17-089r1.html>

Marie-Françoise Voidrot-Martinez, Chris Little, Jürgen Seib, Roy Ladner, Adrian Custer, Jeff de La B: OGC 12-111r1, *OGC Best Practice for using Web Map Services (WMS) with Time-Dependent or Elevation-Dependent Data*. Open Geospatial Consortium (2014). [https://portal.ogc.org/files/?artifact\\_id=56394](https://portal.ogc.org/files/?artifact_id=56394)



- Simon Cox: OGC 10-025r1, *Observations and Measurements – XML Implementation*. Open Geospatial Consortium (2011). [https://portal.ogc.org/files/?artifact\\_id=41510](https://portal.ogc.org/files/?artifact_id=41510)
- Alexandre Robin: OGC 08-094r1, *OGC® SWE Common Data Model Encoding Standard*. Open Geospatial Consortium (2011). [https://portal.ogc.org/files/?artifact\\_id=41157](https://portal.ogc.org/files/?artifact_id=41157)
- Peter Baumann, Eric Hirschorn, Joan Masó: OGC 09-146r8, *OGC Coverage Implementation Schema with Corrigendum*. Open Geospatial Consortium (2019). <http://docs.opengeospatial.org/is/09-146r8/09-146r8.html>
- Arliss Whiteside Jim Greenwood : OGC 06-121r9, *OGC Web Service Common Implementation Specification*. Open Geospatial Consortium (2010). [https://portal.ogc.org/files/?artifact\\_id=38867](https://portal.ogc.org/files/?artifact_id=38867)
- UCUM: Unified Code for Units of Measure (UCUM) – Version 1.9, 2013, <http://unitsofmeasure.org/>
- OMG UML 2.5.1, *Unified Modeling Language*. (2017). <https://www.omg.org/spec/UML/2.5.1/>
- W3C: Extensible Mark-up Language (XML) – Version 1.0 (Fifth Edition), August 2008
- W3C: XML Schema – Version 1.0 (Second Edition), October 2004
- Peter Baumann, Jinsongdi Yu: OGC 12-039, *OGC® Web Coverage Service Interface Standard – Scaling Extension*. Open Geospatial Consortium (2014). [https://portal.ogc.org/files/?artifact\\_id=54504](https://portal.ogc.org/files/?artifact_id=54504)
- Peter Baumann, Jinsongdi Yu: OGC 12-049, *OGC® Web Coverage Service Interface Standard – Interpolation Extension*. Open Geospatial Consortium (2014). [https://portal.ogc.org/files/?artifact\\_id=54502](https://portal.ogc.org/files/?artifact_id=54502)
- Peter Baumann, Jinsongdi Yu: OGC 11-053r1, *OGC® Web Coverage Service Interface Standard – CRS Extension*. Open Geospatial Consortium (2014). [https://portal.ogc.org/files/?artifact\\_id=54209](https://portal.ogc.org/files/?artifact_id=54209)
- Peter Trevelyan, Paul Hershberg, Steve Olson: OGC 15-045r7, *OGC MetOcean Application profile for WCS2.1: Part 0 MetOcean Metadata*. Open Geospatial Consortium (2020)



4

# TERMS AND DEFINITIONS

---

## TERMS AND DEFINITIONS

---

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

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

This document also uses terms defined in the OGC Standard for Modular specifications (OGC 08-131r3), also known as the ‘ModSpec’. The definitions of terms such as standard, specification, requirement, and conformance test are provided in the ModSpec.

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

### 4.1. numerical weather prediction model

---

A Numerical Weather Prediction (NWP) 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.

### 4.2. reference time

---

Reference time is a temporal parameter used to represent a time axis that can be mapped to some relevant referent time other than validity time. The semantic meaning can differ for different types of data. For numerical weather forecasts, it may be a nominal time where observations have been assimilated to initialize the calculation. Synonym for reference time: model run time.

An example of reference time: 2017-12-12T00.00.00Z

**Note 1 to entry:** “reference time” will be used in preference to “model run time” as it is more generic and includes services that may be continually updated.

## 4.3. validity time

---

Validity time is an attribute value specified by an instant in, or duration of, universal chronological time that identifies when information is valid or applicable. In ISO 19156, the validity time has the semantics of phenomenonTime. Deciding if the data have a 'validity time' is an important step.

Synonyms for validity time: verification time.

An example of validity time 2017-12-12T12.00.00Z

**Note 1 to entry:** Forecast models running with different reference times will have, for some fields, the same verification time if the durations of the different model runs overlap.

## 4.4. GRIB

---

GRIB stands for Gridded Binary. GRIB is a WMO (World Meteorological Organisation) format for gridded binary data exchanged between member countries, including a controlled vocabulary defined in tables.

## 4.5. Web Coverage Service 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.

## 4.6. 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.

## 4.7. DescribeCoverage

---

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.

## 4.8. path

---

The path is simply the route or course along which something travels or moves, for example the path of an aeroplane.

## 4.9. Corridor

---

A Corridor, in this document, is a trajectory (aka path) with a lateral and vertical extent. The corridor may be multi-dimensional, and in the case of aviation is often four dimension, i.e. x, y, z, t.

## 4.10. GetCorridor operation

---

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



5

# CONVENTIONS

---

## CONVENTIONS

---

This sections 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.

### 5.1. ABBREVIATED TERMS

---

**GML** Geography Markup 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

**WCS2.0** OGC Web Coverage Service version 2.0

**WCS2.1** OGC Web Coverage Service version 2.1

**WMO** World Meteorological Organization

**XML** W3C Extensible Markup Language

**XSD** W3C XML Schema Definition Language

### 5.2. 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].

## 5.3. 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.



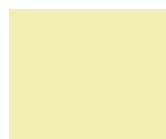
Blue: WCS2.1 plus extensions (rsub, scal, int and crs)



Orange: CIS (Coverage Implantation Schema 1.1)



Green: This standard



Tan: WCS2.1



6

# VOCABULARIES

---

This standard defines a number of properties that require the use of codes or vocabulary items. In some cases a list of terms is 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 `rsub: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	<a href="http://codes.wmo.int/ grib2">http://codes.wmo.int/ grib2</a>
Discipline	<a href="http://codes.wmo.int/grib2/codeflag/ 0.0">http://codes.wmo.int/grib2/codeflag/ 0.0</a>
Fixed surface types and units	<a href="http://codes.wmo.int/grib2/codeflag/ 4.5">http://codes.wmo.int/grib2/codeflag/ 4.5</a>
Parameter category	<a href="http://codes.wmo.int/grib2/codeflag/ 4.1">http://codes.wmo.int/grib2/codeflag/ 4.1</a>
Parameter number	<a href="http://codes.wmo.int/grib2/codeflag/ 4.2">http://codes.wmo.int/grib2/codeflag/ 4.2</a>

7

# NON-NORMATIVE (INFORMATIVE) MATERIAL

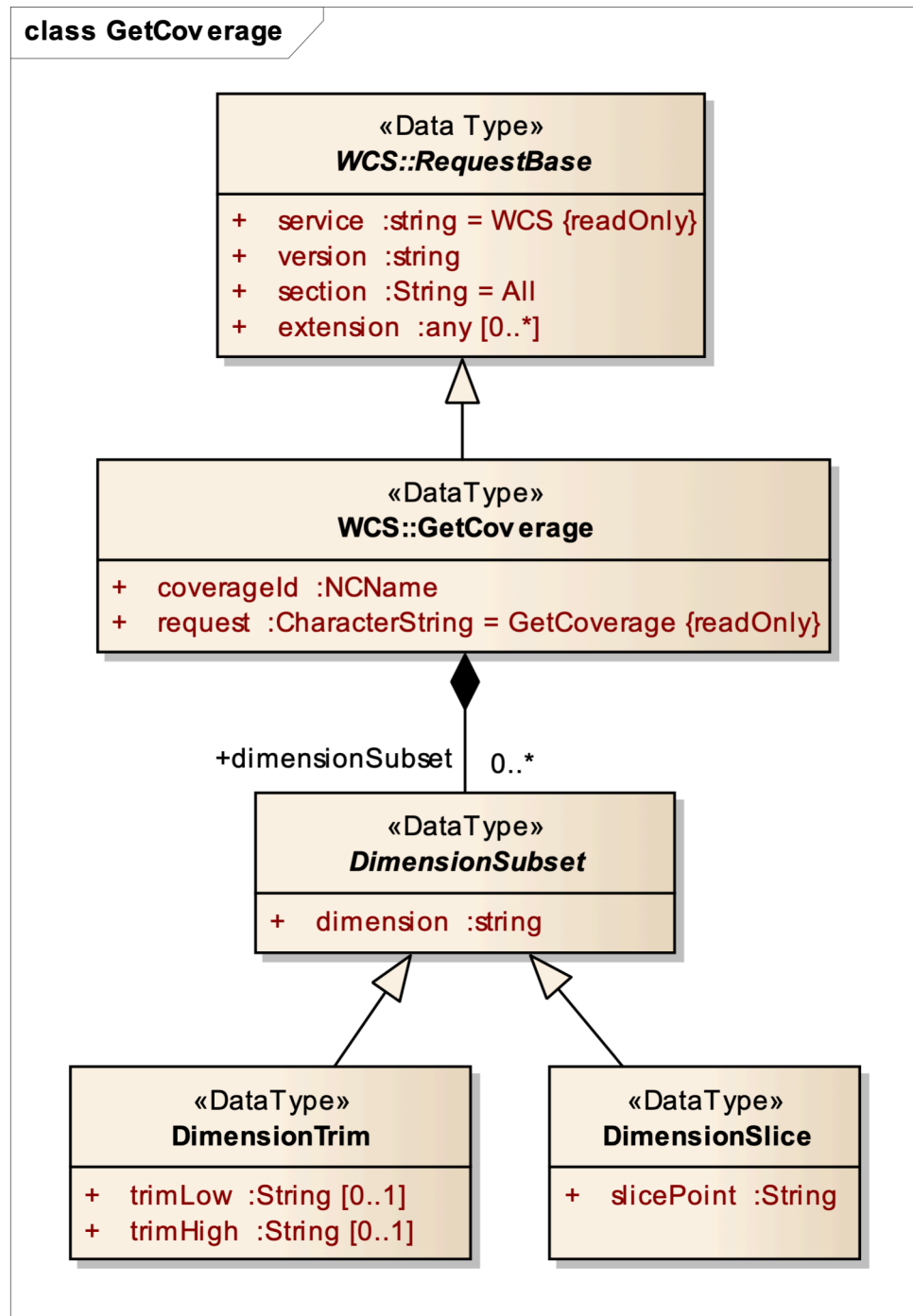
---

## NON-NORMATIVE (INFORMATIVE) MATERIAL

---

The GetCorridor 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 see Figure 1. See the OGC® Web Coverage Service 2.1 Interface Standard – Core ([OGC 17-089r1](#)). This specific extension is designed specifically to extract corridors 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 (in this case corridors) data on the server and transferring the relevant data to the client. The obvious candidate for this service is the OGC's WCS2.1 and it is therefore logical to extend this standard to include an additional GetCorridor extension. The advent of the Coverage Implementation Schema (CIS1.1) has made this much easier through the use of axis specific definitions.

## 7.1. WCS2.1

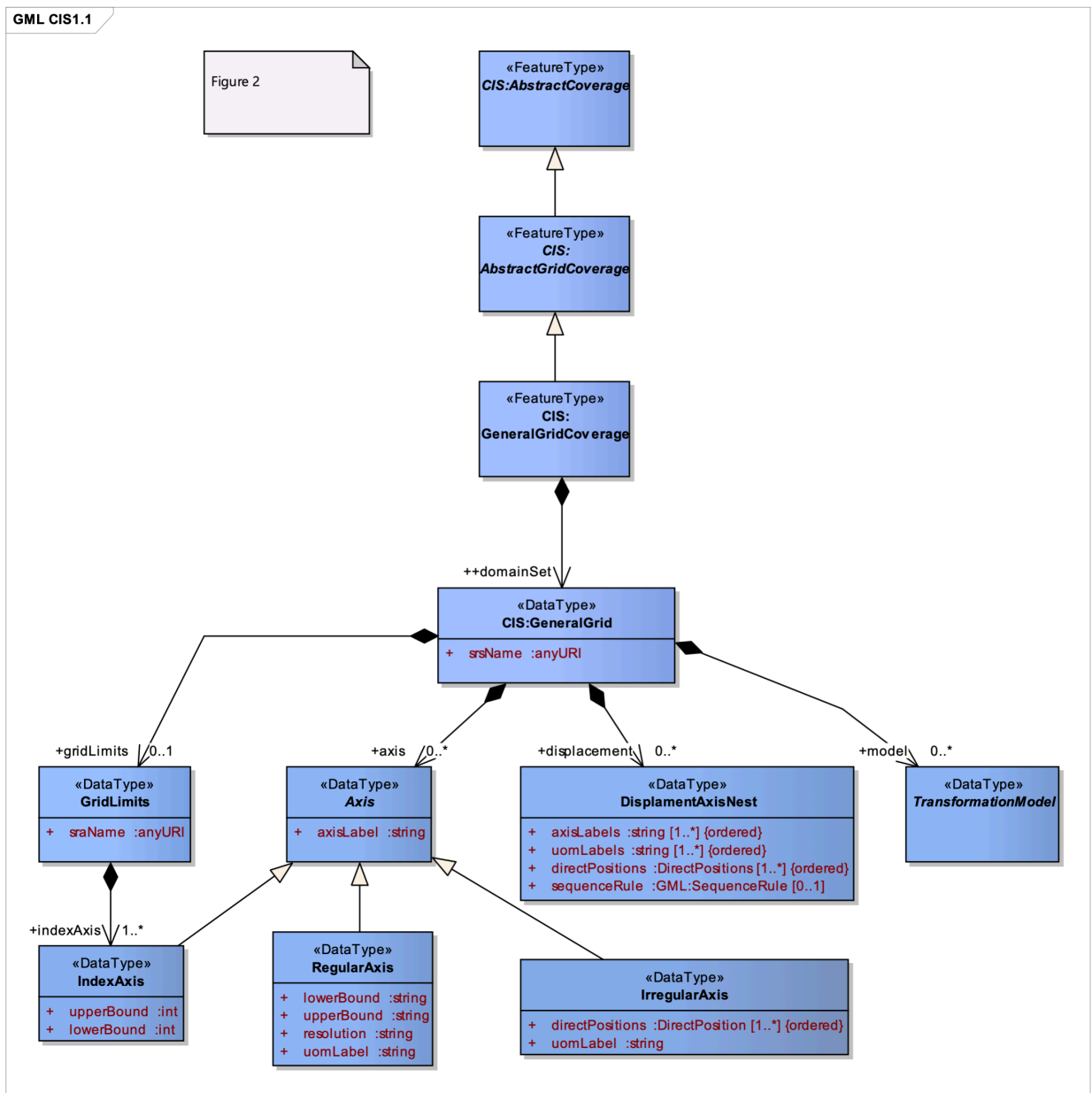
---

The WCS2.1 files (see [https://portal.opengeospatial.org/files/?artifact\\_id=67116&version=1](https://portal.opengeospatial.org/files/?artifact_id=67116&version=1)) form the core standard and the extensions (see below) describe the GetCapabilities, DescribeCoverage and GetCoverage operations. The GetCorridor 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 Interpolation Extension, version 1.0.0, [OGC 12-049](#)
- WCS CRS Extension, version 1.0, [OGC 11-053r1](#)

The main benefit of WCS2.1 to the MetOcean Profile and specifically the getCorridor operation is that it allows the description of a CIS 1.1 Coverage (see Figure 2). This is important in that CIS 1.1 supports multi-dimensional coverages and the encoding of coverage types such as corridors.

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



## 7.2. KEY CONCEPTS

### 7.2.1. 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. The nominal “start” time is known amongst the MetOcean community,

as the “model run time” (i.e. a notional starting point). All forecast times for a specific model run are therefore relative to this “reference” time. It is important to note that term “reference time” will be used in preference to “model run time”, as it is more generic and includes services that may be continually updated.

## 7.3. 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. In addition, 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.

### 7.3.1. 4D Coverages

A typical NWP forecast may typically be expressed as a set of 2D coverages, but these are not always based on rectified grids. Also, a typical model run contains thousands of 2D coverages and the metadata returned by the GetCapabilities response quickly becomes unmanageable. The problem can be simplified by identifying, wherever possible, “4D Coverages”. This was made much easier with the adoption of the OGC’s CIS 1.1.

In addition, a typical numerical simulation has a number of different vertical coordinates F(i.e. pressure, height above mean sea level, height above ground, surface, max wind level). By forming a 4D coverage from all of the 2D coverages that share the same horizontal, vertical and temporal domains, the number of coverage identifiers can be significantly reduced, thus reducing complexity. This is a challenge, as the vertical and temporal axes are not regular and need to be enumerated. Fortunately, the “GeneralGridCoverage” as described CIS makes this possible with the use of the IrregularAxis component (see figure 2).

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 multidimensional cube with properties, e.g. temperature, wind and humidity. It is therefore possible to make geospatial queries. For WCS2.1, this equates to the GetCoverage operation and the proposed new operation GetCorridor. There are special cases where the vertical axis has no vertical dependency (e.g. surface, max wind level), and are described with a simple name. The properties of each coverage are given common names such as temperature, humidity etc. It will be common for some coverages to have these parameters, or properties, in common.



## 7.4. TIME DEPENDANT DATA (FROM WMS BEST PRACTICE OGC DOCUMENT:12-111R1)

---

Complex data sets can have temporal dependencies of many kinds. This document adopts the phrase ‘validity time’ that is essentially identical to the concept of ‘phenomenonTime’ from ISO 19156:2011, Geographic information – Observations and measurements. ISO 19156:20011 refers to the applicability of the data using a chronological Gregorian calendar.

Furthermore, data are temporally dependent relative to some reference time instant. Examples include:

- Observations with an accession time into a data repository.
- Numerical weather forecasts with a nominal time where observations have been assimilated to initialize the calculation.
- Alerts with a time denoting when they are issued or published.

The diversity of such reference times precludes defining a dimension type with explicit semantics, although the need for a mechanism to distinguish data based on some temporal referent is widely shared. Thus, the definition of a generic dimension, named `referenceTimeAxis`, may be used for such occasions and is supported in this profile.

This WCS2.1 standard uses a combination of time stamp, a list of time stamps, or a `start_time/end_time/time_interval`, to enumerate time. The semantics of this time stamp string representation is built from both time components and specific separators. A full time stamp string representation has the following format:

“YYYY-MM-DDThh:mm:ss.SSSZ”

Where:

- YYYY indicates a 4-digit year
- MM indicates a month
- DD indicates a day of a month
- T is the separator between the date part and the time part
- hh indicates an hour
- mm indicates a minute
- ss indicates a second
- SSS indicates a millisecond
- Z is the time zone designator for the zero UTC offset

The precision of a time stamp “t” is determined by the last time component. Time stamps may be associated with a time zone. If no time zone is specified with a time stamp “t”, then “t” is assumed to be in local time.

A time interval is expressed as tmin/tmax/r, where tmin and tmax are timestamps that define the lower and upper bounds of the interval and r is the resolution. The interval contains all timestamps tmin + i \* r, i >= 0, that are lower or equal than tmax. A resolution r is represented by the format P [n1Y] [n2M] [n3D] [T [n4H] [n5M] [n6S]] where:

- P is a starting character.
- Y is the year designator that follows the value n1 for the number of years.
- M is the month designator that follows the value n2 for the number of months.
- D is the day designator that follows the value n3 for the number of days.
- T is the time designator that precedes the time components of the representation.
- H is the hour designator that follows the value n4 for the number of hours.
- M is the minute designator that follows the value n5 for the number of minutes.

Some examples of time:

#### 1- A Time stamp

2015-05-15T00:00:00Z

#### 2- A list of time stamps

2015-05-15T00:00:00Z, 2015-05-15T06:00:00Z, etc.

#### 3- A start and end time

2015-05-15T00:00:00Z/2015-05-17T12:00:00Z

#### 4- Example of a start time/end time/interval

2015-05-15T00:00:00Z/2015-05-17T00:00:00Z/PT12H

Note that time duration refers to a span of time relative to the reference time. Reference time is specified using the following construct

```
<om:parameter
  xmlns:om="http://www.opengis.net/om/2.0"
  xmlns:gml="http://www.opengis.net/gml/3.2"
  xmlns:xlink="http://www.w3.org/1999/xlink">
<om:NamedValue>
<om:name xlink:href="http://codes.wmo.int/grib2/codeflag/1.2/
significanceOfReferenceTime" xlink:title="Start of Forecast"/>
<om:value xsi:type="gml:TimeInstantType" gml:id="referenceTime">
<gml:timePosition>2015-05-15T03:30:00Z</gml:timePosition>
</om:value>
```

</om:NamedValue>  
</om:parameter>

Examples of time duration include PT0H, PT6H, PT12H. Consequently, PT12H denotes a time duration of 12 hours relative to the reference time.

When no reference time is specified, but the times are relative to a starting and end point then a recurring time interval can be used. An example of this is:

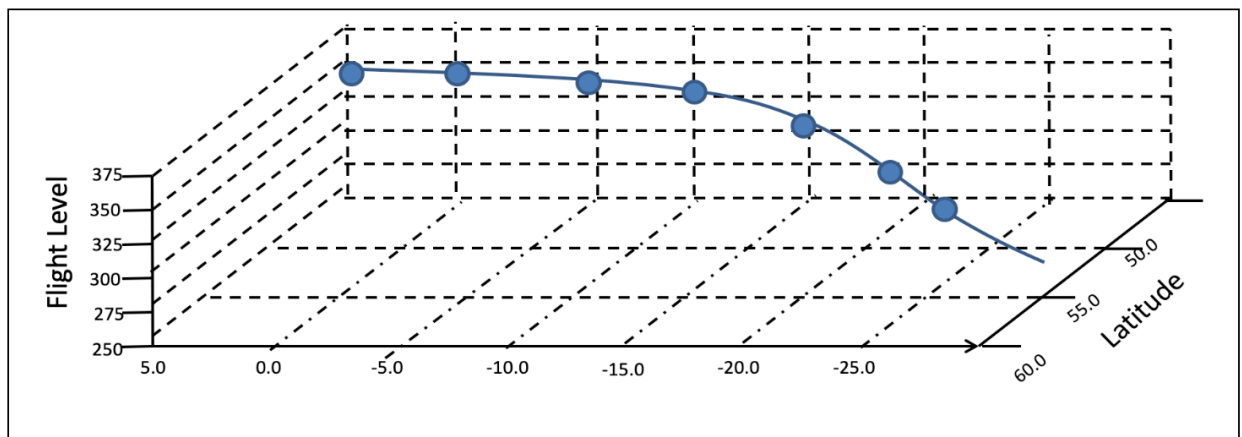
2015-05-15T00:00:00Z/2015-05-17T00:00:00Z/PT12H

When no reference time is specified, a set of timestamps may also be used. An example of this is: 2015-05-15T00:00:00Z, 2015-05-15T12:00:00Z, 2015-05-15T18:00:00Z/ *Note where times are irregular, the form start/end/interval is not appropriate.*

## 7.5. TRAJECTORIES AND CORRIDORS:

In this document a trajectory with a lateral and vertical extent is known as a corridor. The corridor may be multi-dimensional, as in the case of aviation that has four dimension, i.e. x, y, z, t). A typical flight path is shown in Figure 3.

Figure 3 – A typical flight path



It will be constructive to present some scenarios and how they can be used to extract a trajectory/corridor. These examples are based on MetOcean use cases, but can be easily extended to other communities of interest. It is not expected that all scenarios will be available on the server, and it will be necessary to advertise different levels of server capability via the GetCapabilities response document.

## 7.5.1. Scenario 1: Specifying the path

In scenario 1, the path will be defined within the GetCorridor request as a specialized element metocean:PathDescription, this is a specialised type of a cis:GeneralGridType. This example is based on Figure 3, although the detail is not exact!

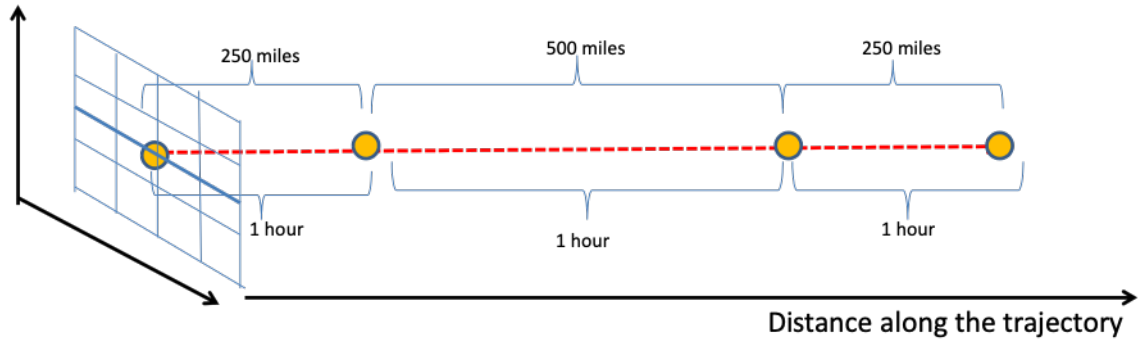
```
<metoceancorridor:path
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:metoceancorridor="http://www.opengis.net/wcs/metoceanProfile_getCorridor/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:cis="http://www.opengis.net/cis/1.1/gml"
xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<metoceancorridor:PathDescription srsName="http://www.opengis.net/def/crs-compound?
1=http://www.opengis.net/def/crs/EPSG/0/4326&
2=http://http://www.opengis.net/def/crs/OGC/0/AnsiDate&
3= http://codes.wmo.int/grib2/codeflag/4.5/_102/Specific_altitude_above_mean_sea_level"
axisLabels="Lat Lon Time Specific_altitude_above_mean_sea_level">
<cis:DisplacementAxisNest axisLabels="Lat Lon Time Specific_altitude_above_mean_sea_level"
uomLabels="deg deg ISO8601 ft">
<cis:P><cis:C>45.0</cis:C><cis:C>5.0</cis:C><cis:C>2017-05-15T00:00:00Z</cis:C><cis:C>3600</cis:
C></cis:P>
<cis:P><cis:C>46.0</cis:C><cis:C>0.0</cis:C><cis:C>2017-05-15T00:15:00Z</cis:C><cis:C>3500</cis:
C></cis:P>
<cis:P><cis:C>47.0</cis:C><cis:C>-5.0</cis:C><cis:C>2017-05-15T00:30:00Z</cis:C><cis:C>34200</cis:
C></cis:P>
<cis:P><cis:C>48.0</cis:C><cis:C>-10.0</cis:C><cis:C>2017-05-15T00:45:00Z</cis:C><cis:C>3400</cis:
C></cis:P>
<cis:P><cis:C>49.0</cis:C><cis:C>-15.0</cis:C><cis:C>2017-05-15T01:00:00Z</cis:C><cis:C>3200</cis:
C></cis:P>
<cis:P><cis:C>51.0</cis:C><cis:C>-20.0</cis:C><cis:C>2017-05-15T01:15:00Z</cis:C><cis:C>3000</cis:
C></cis:P>
<cis:P><cis:C>52.0</cis:C><cis:C>-25.0</cis:C><cis:C>2017-05-15T01:30:00Z</cis:C><cis:C>2000</cis:
C></cis:P>
</cis:DisplacementAxisNest>
<cis:GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index1D" axisLabels="i">
<cis:IndexAxis axisLabel="i" lowerBound="0" upperBound="6"/>
</cis:GridLimits>
</metoceancorridor:PathDescription>
</metoceancorridor:path>
```

## 7.5.2. Scenario 2: Breaking the path into segments

In scenario 2, the trajectory may, if required, be broken up into segments, therefore creating extra “sample points”. If a time dimension is defined, then a choice can be made for having segments equal in time or distance. Breaking up the path into segments, either by time or distance, must be specified in the GetCorridor request in the scal:TargetAxisSize element as part of the scaling specification. For a trajectory broken up into equal time segments, the distances will be equal only between “way points” , as a constant speed is assumed.

Figure 4 represents a typical trajectory with just the 'way points'.

Figure 4 – way points

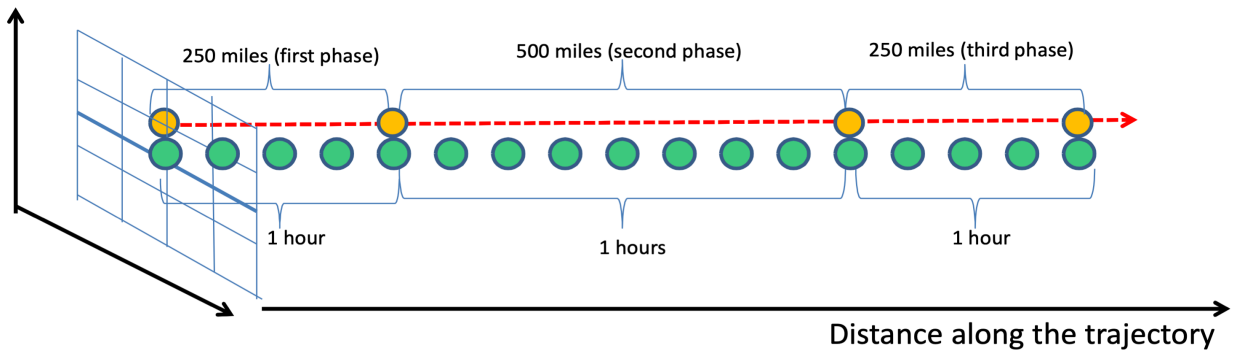


Note each yellow marker represents a “way point” in time and space. Essentially the 4D trajectory is transformed into a 3D feature.

### 7.5.3. Scenario 3: Breaking the path into segments of equal distance

In this scenario, the trajectory is broken up into 16 segments (see Figure 5) equal in distance (i.e. 62.5 miles). For the first and last phases, this equates to 15 minute intervals. For the second phase, the time interval is 7.5 mins i.e. twice the first and last phases. Note that the segment points and the wave points overlap, but this is not always the case.

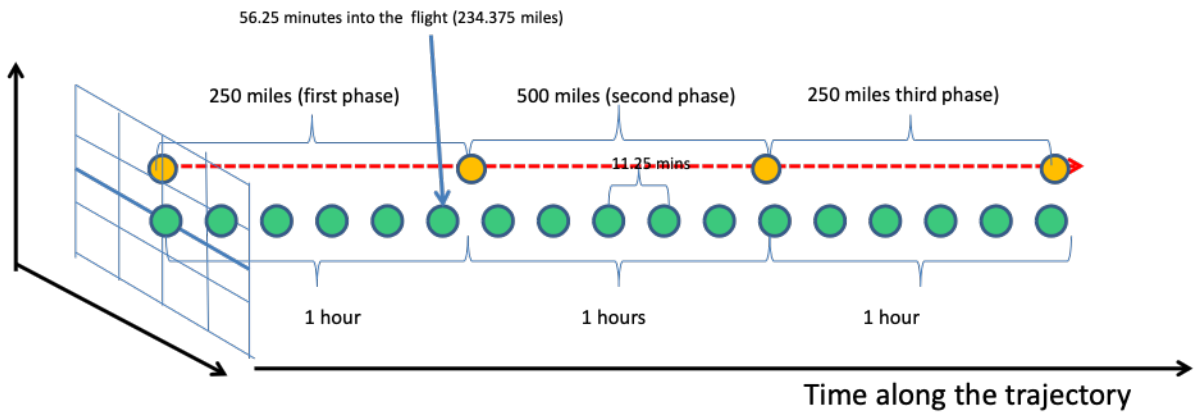
Figure 5 – Segments equal in distance



### 7.5.4. Scenario 4: Breaking the path into segments of equal time

Figure 6 shows a trajectory broken up into 16 segments equal in time (i.e. 11.25 minutes). For the first five segments, the speed is assumed to be 250 mph. For the next 6 segments, the average speed is assumed to be 500 mph. Note that the segment points and the “way points” do not overlap for each point. This may not be what is really required and it would be better to break up each “phase” into a discrete number of segments.

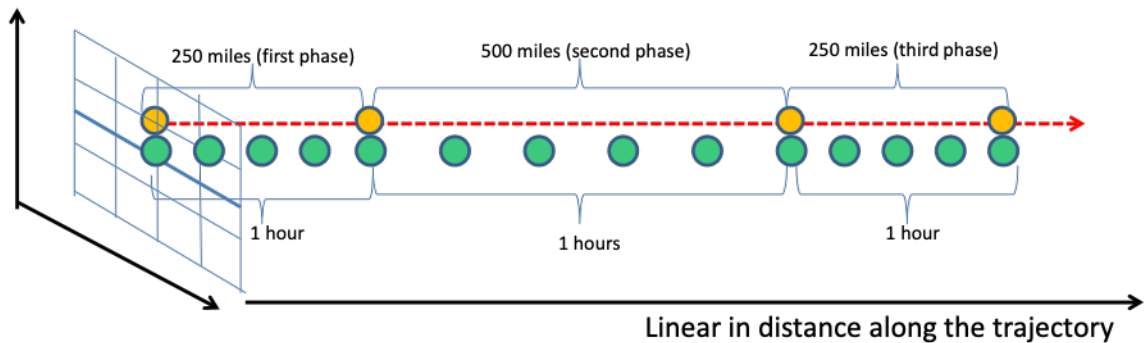
Figure 6 – Segments equal in Time



### 7.5.5. Scenario 5: Segments equal in distance for each “phase” broken into a discrete segments.

Figure 7 shows a trajectory broken up into 13 segments. For phases 1 and 3, this equates to a distance of 62.5 miles. For the middle phase, the distance between segments is 100 miles. For the first and last phases, this equates to 15 minute intervals. For the second phase, the time interval is 12 mins. Note that the segment points and the “way points” will always coincide.

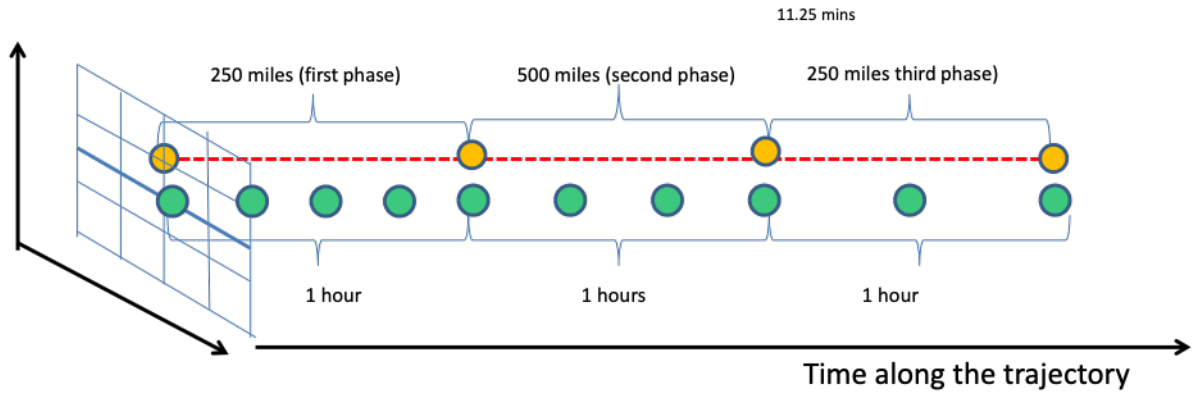
Figure 7 – Discrete Segments equal in distance



### 7.5.6. Scenario 6: Segments equal in time for each “phase” broken into a discrete segments,

Figure 8 shows a trajectory broken into an integral number of segments for each phase. For phase 1, this equates to 15 minute intervals or 62.5 miles. For the middle phase, the distance between segments is 33.3 minutes or 166.67 miles. For last phases, this equates to 30 minute intervals or 125 miles. Note that the segment points and the “way points” will always coincide.

Figure 8 – Discrete Segments equal in time



### 7.5.7. Scenario 7: Describing the corridor:

Corridor vertical aligned with the local vertical coordinate

The depiction of the corridor seen in Figure 9 shows the trajectory, shown in red, based on the “way points”, shown in yellow. The corridor, shown in blue, is based on a simple grid that is orthogonal to the trajectory. An alternative view is shown in Figure 10. Note the vertical orientation of the corridor is aligned with the vertical coordinate.

Figure 9 – Corridor aligned with trajectory axis

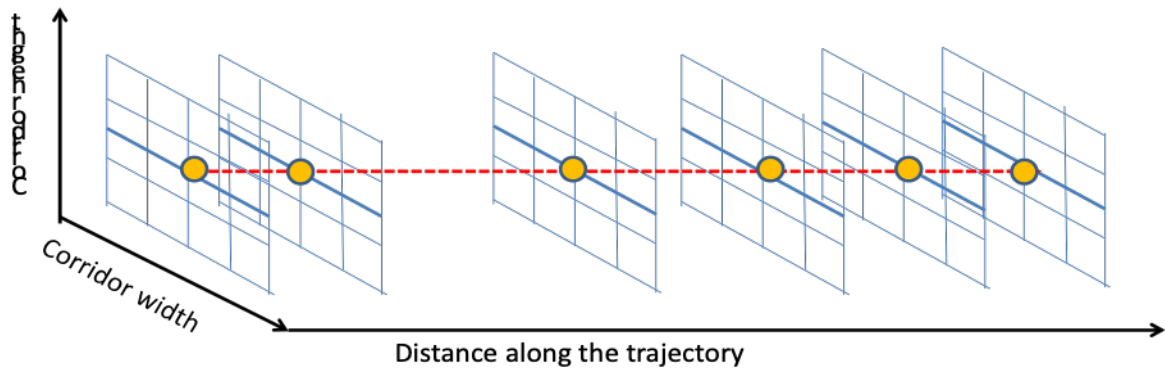
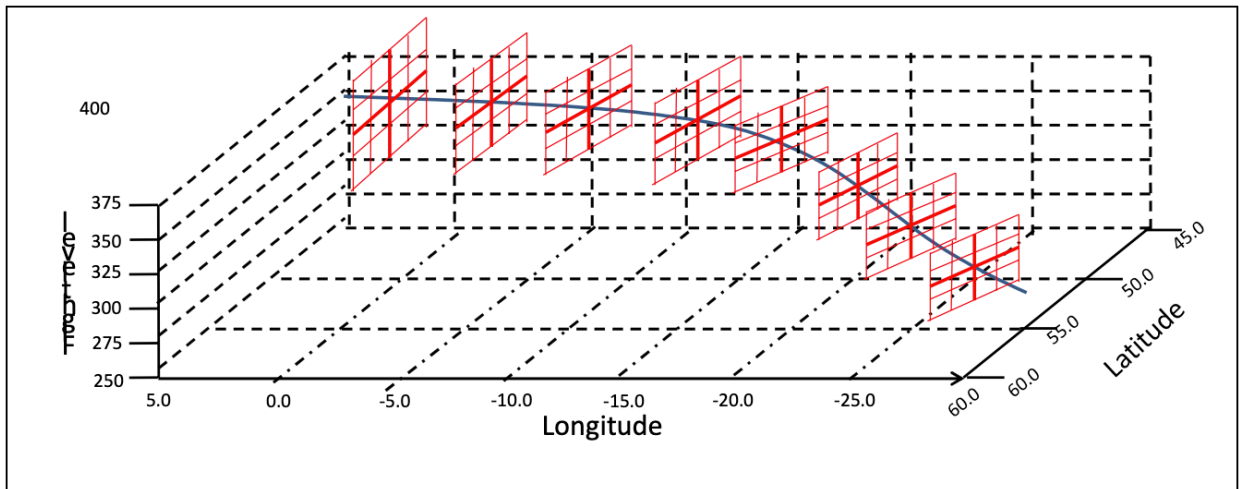


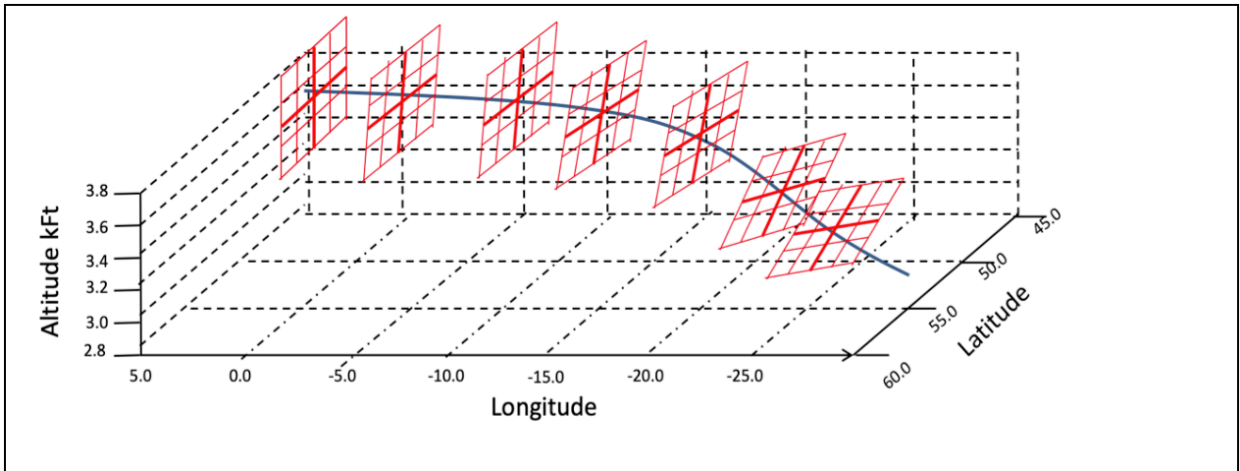
Figure 10 – Alternative view of the corridor aligned with trajectory axis



### 7.5.8. Corridor vertically aligned perpendicular to the trajectory.

In this example the vertical alignment of the corridor may be required to be perpendicular to the flight path. This is particularly important where the trajectory is very steep (see Figure 11)

Figure 11 – Corridor vertical aligned perpendicular to the trajectory



## 7.6. METHODS OF EXTRACTION

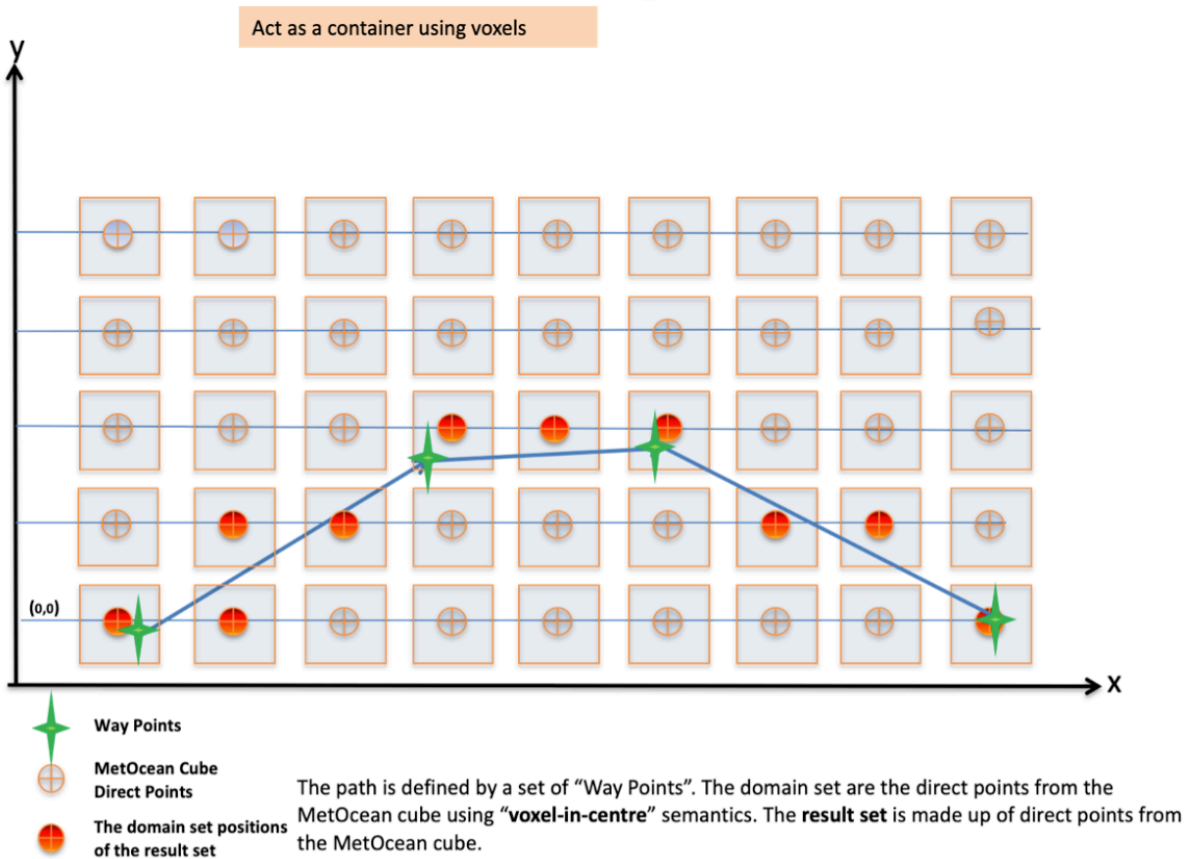
There are a number of extraction patterns and those that are considered to be most relevant are listed below. The diagrams are mainly in 2D, but are detailed enough to convey the concepts being described.



### 7.6.1. Method: 1 Trajectory Intersection using Voxel semantics

With reference to Figure 12, the path is defined by a set of “way points” (shown in green). The coverage returned in response to a successful GetCorridor request would have as its “domain set” the center points of the “grid boxes” intersected by the trajectory. The “result set” is made up of direct points from the MetOcean coverage (in red) extracted from each grid box that the path intersects.

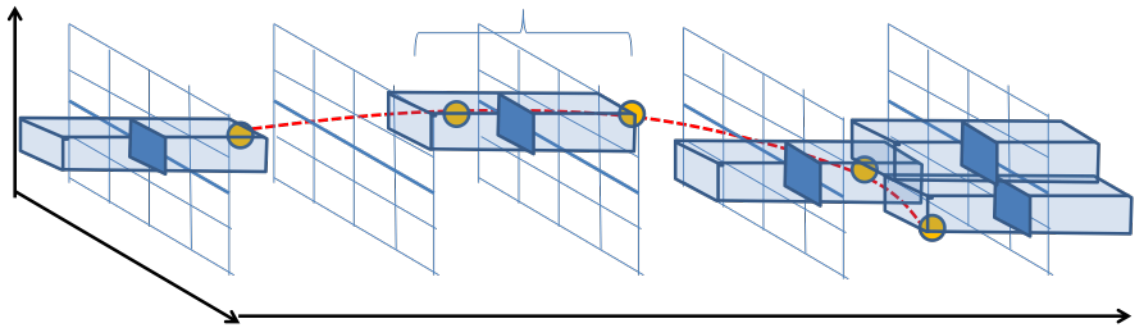
Figure 12 – Extraction of the trajectory using grid-box semantics



An alternative view:

In Figure 12 there is an alternative depiction. Note that there may not be a continuous set of Voxels intercepted.

Figure 13 – A ‘Voxel’ view of the trajectory



### 7.6.2. Method 2: (Corridor with “Voxel” semantics)

In method 2 the path is defined by a set of “way points” with a horizontal buffer region around the path. The coverage returned in response to a successful GetCorridor request would have as its “domain set” the points contained within the buffer using “grid-box -in-centre” semantics. The “result set” is made up of direct points from the MetOcean coverage (in red) extracted from each grid box within the boundary box (see Figure 14). No horizontal interpolation is performed.

Figure 14 – Extraction of the corridor using grid-box semantics

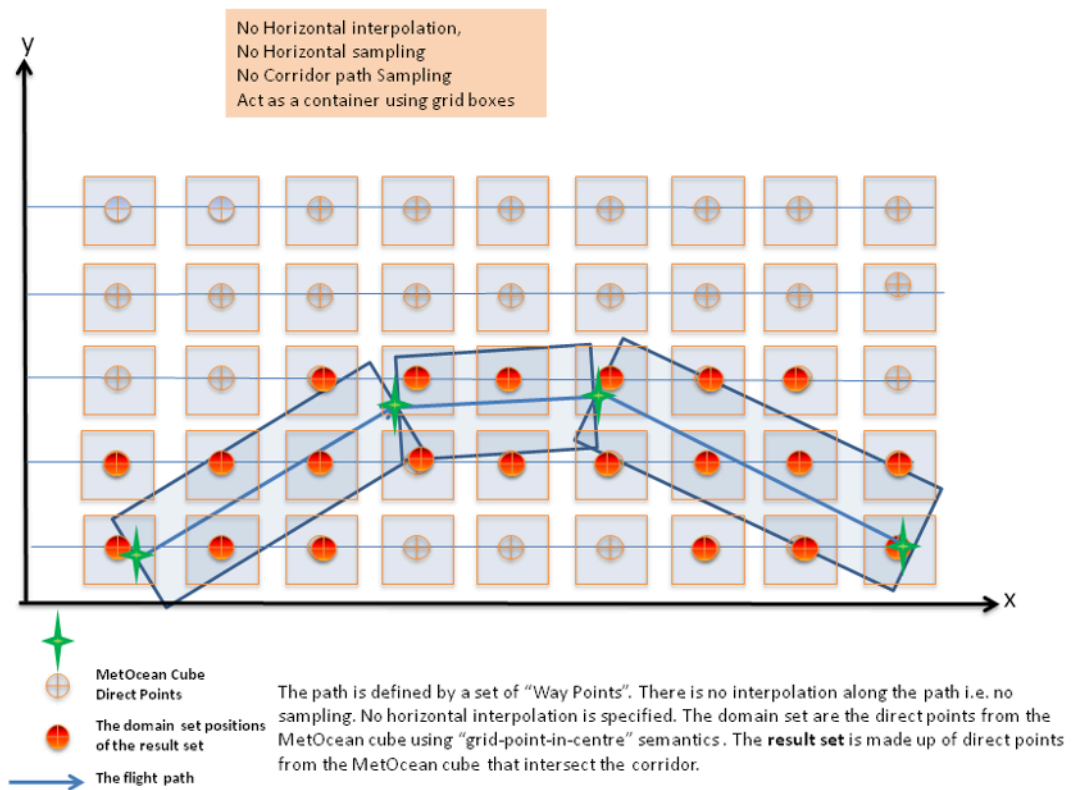


Figure 15 – A "Voxel" view of the corridor

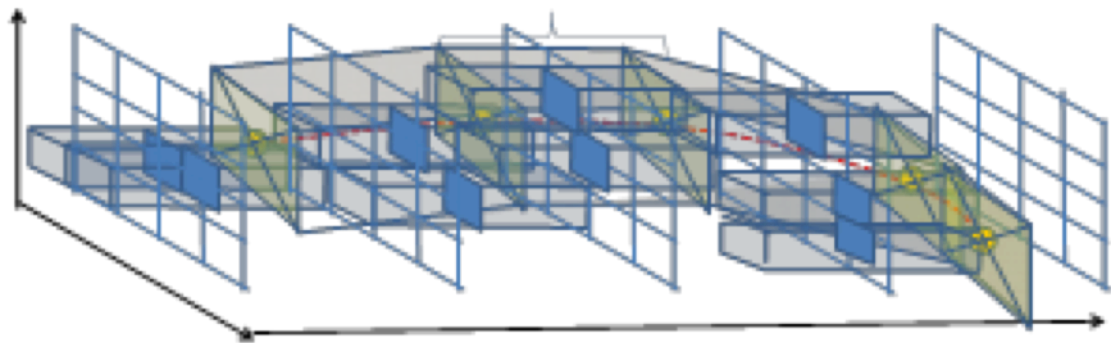


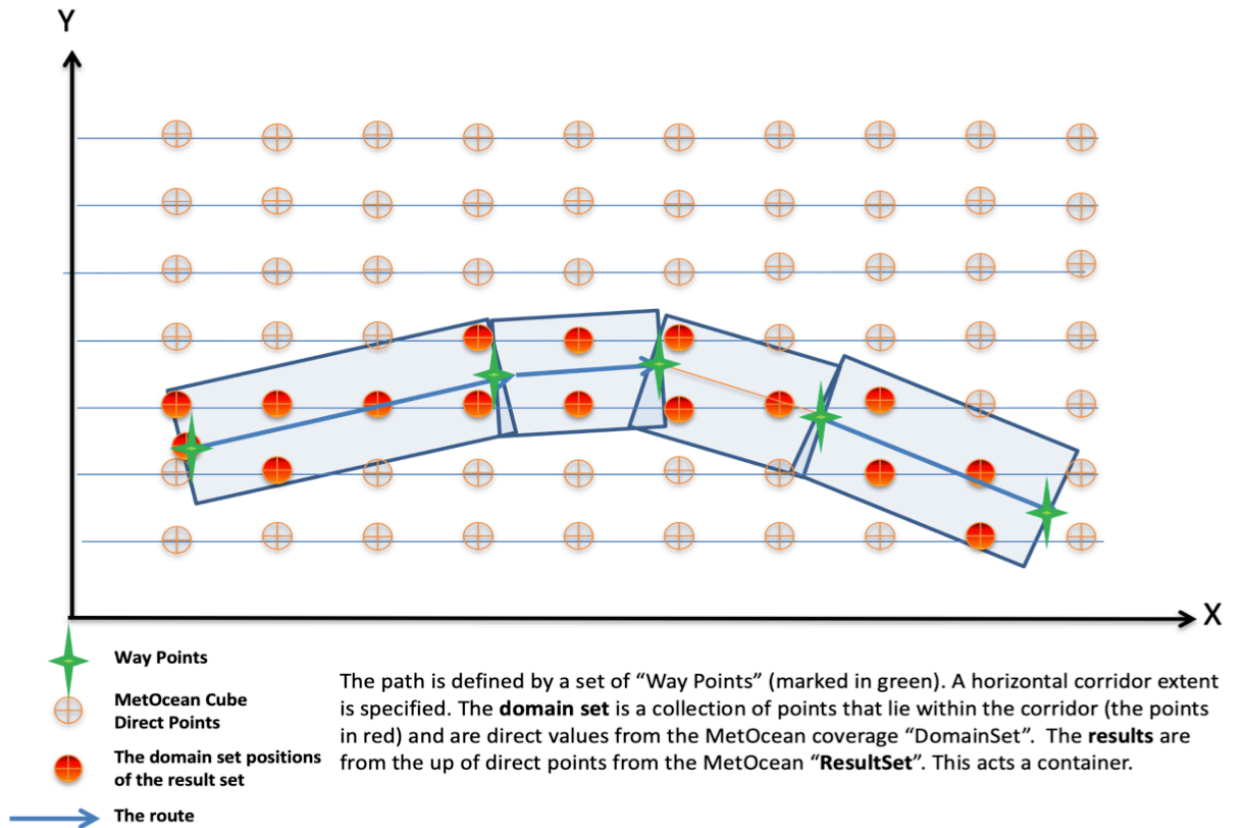
Figure 15 is an attempt to show how a corridor with "Voxels" would look like in 3D.

### 7.6.3. Method 3: (Corridor point collection)

In this method all the grid points contained within the corridor form a *collection*. The path is defined by a set of "way points" (in green) with a horizontal buffer region around the path (defined by a "horizontal buffer extent"). The coverage returned in response to a successful

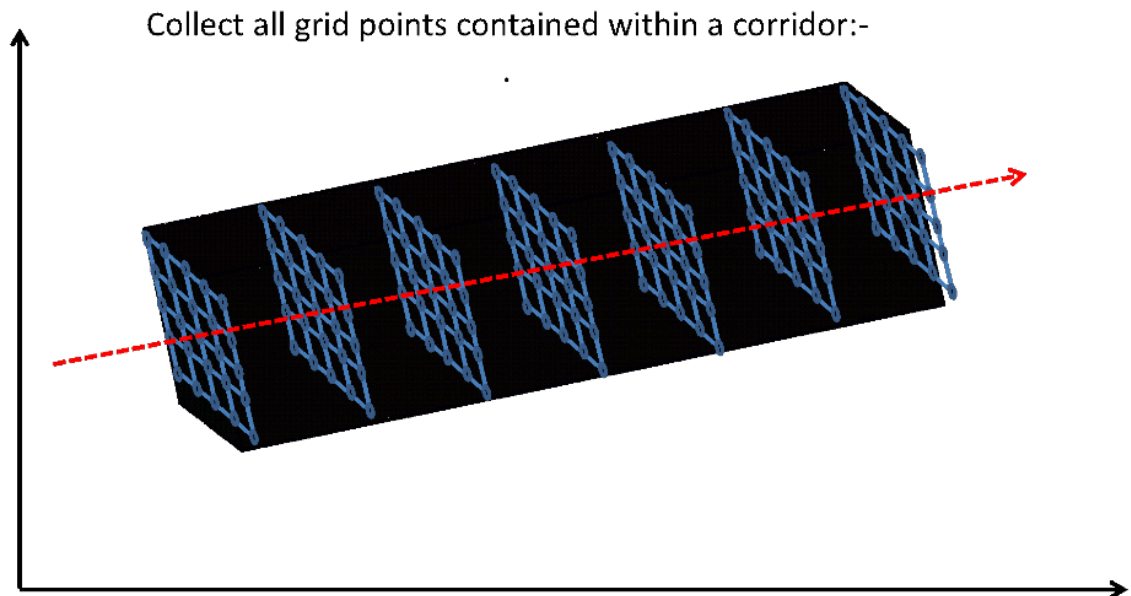
GetCorridor request would have as its “domain set” all the points that lie within the corridor (the points are in red) The “result set” is made up of direct points from the MetOcean coverage (in red) extracted from each point that lies within the boundary box (see Figure 16).

Figure 16 – Extraction of a corridor, using point based semantics



Note that this is different from the first two examples as it based on “grid point” semantics, not “grid-box” semantics.

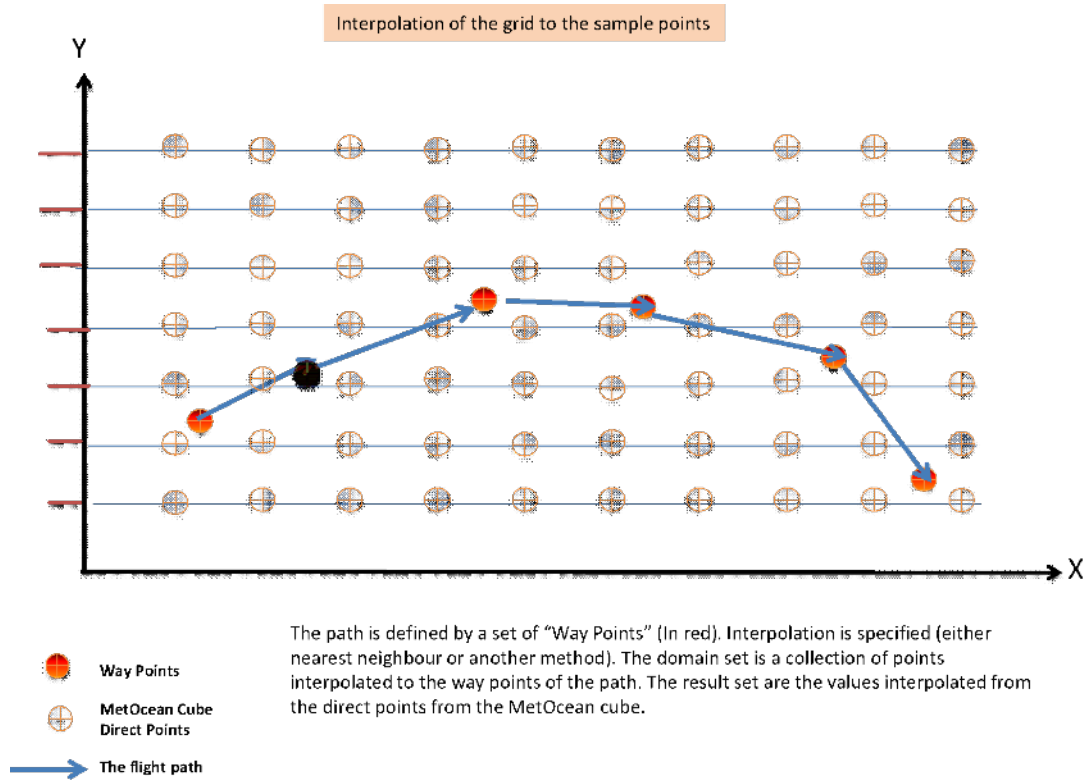
Figure 17 – Extraction of a corridor, using point based semantics



#### 7.6.4. Method: 4 (Trajectory with point based semantics)

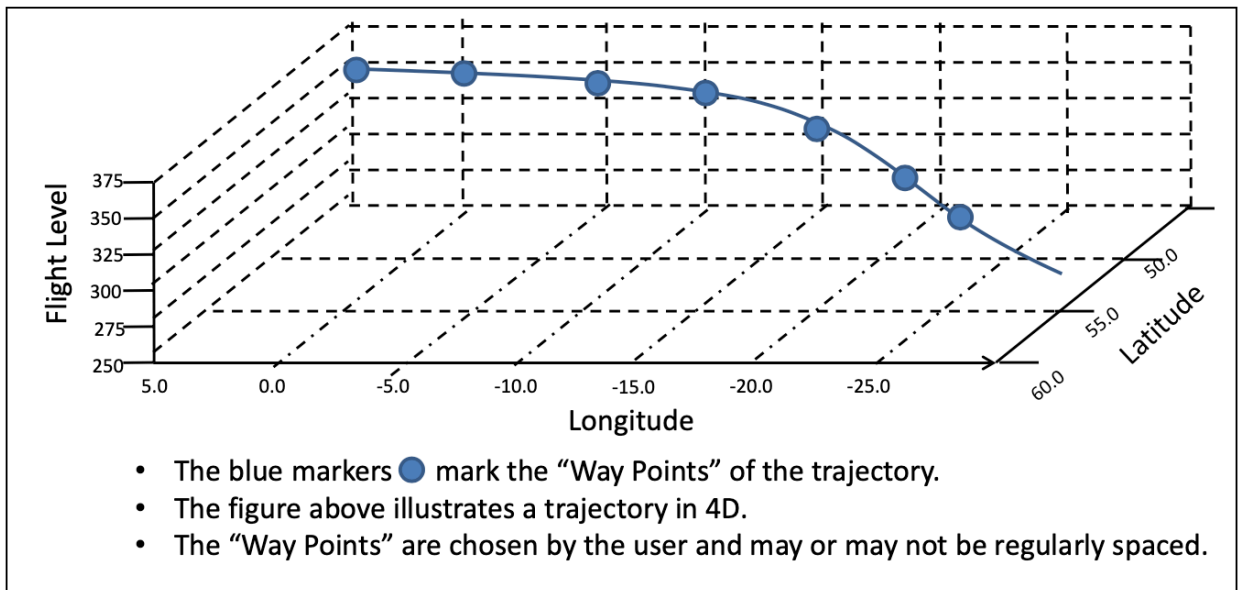
In method 4 the path is defined by a set of “way points” (in red) that are un-equally spaced. The coverage returned in response to a successful GetCorridor request would have as its “domain set” the way points that lie along the trajectory (the points in red). The “result set” is calculated by interpolating horizontally and if required, temporally from the MetOcean coverage (see Figure 18)

Figure 18 – Trajectory with point based semantics



If there a different time for each way point then the data will be interpolated to the correct time for each point given the coverage has a time dimension. Thus each returned data point would have a different time.

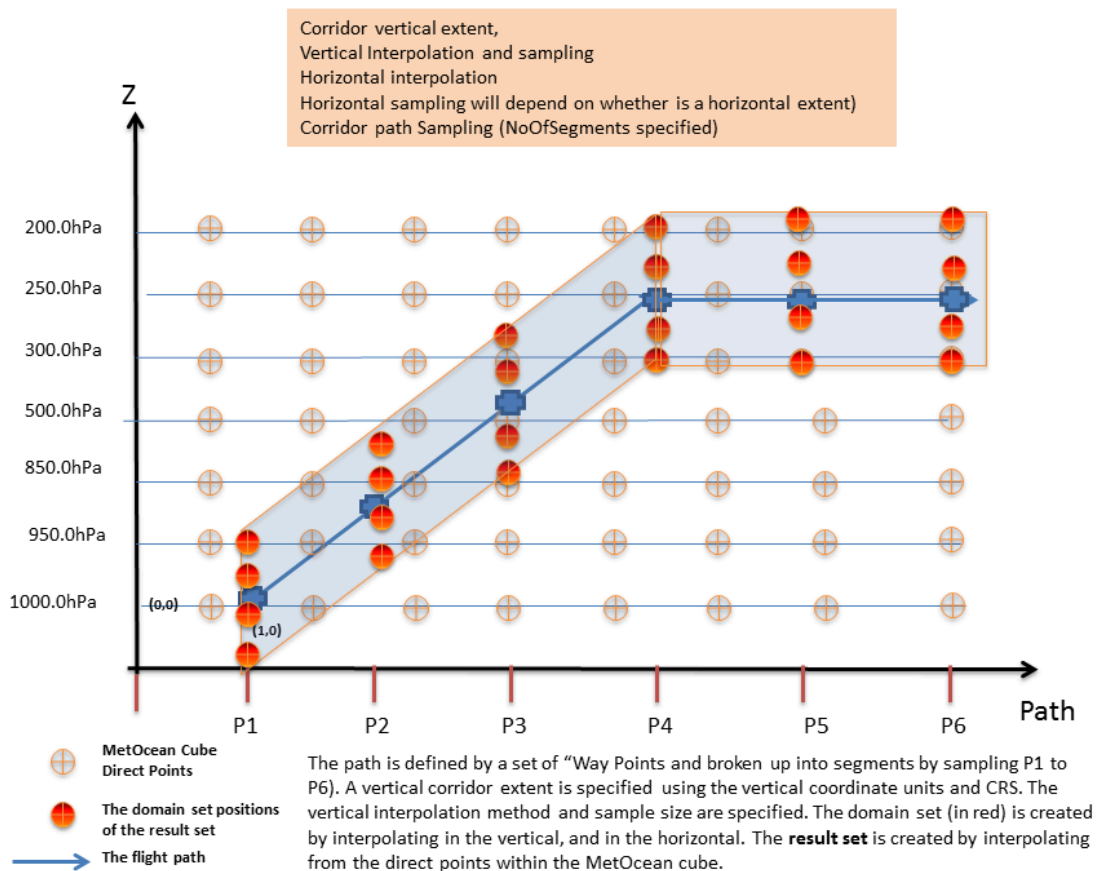
Figure 19 – Trajectory with point based semantics



### 7.6.5. Method: 5 (Corridor grid with vertical aligned with grid)

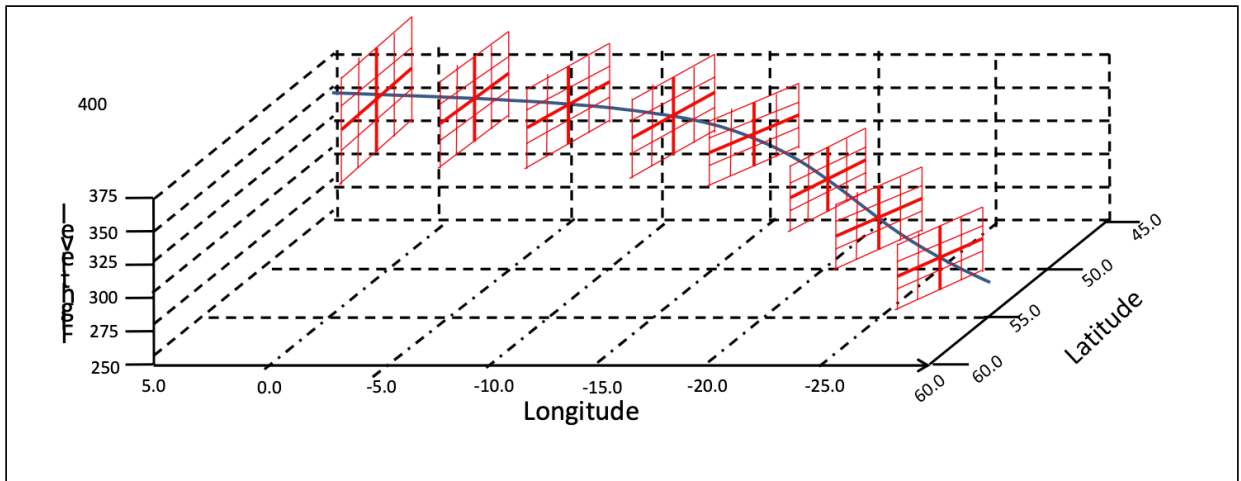
In method 5 the path, denoted in the diagram by blue stars is shown in figure 5. The coverage returned in response to a successful GetCorridor request would have as its “domain set” each sample point (the points are shown in red). The “result set” is calculated by interpolating from the MetOcean coverage to each of the sampling points (see Figure 20). Note that the orientation of the vertical axis and its relationship to the path. Compare this with the next method.

Figure 20 – Corridor grid with vertical aligned with grid



The following Figure 21 shows an alternative view of Figure 20.

Figure 21 – Corridor grid with vertical aligned with grid



### 7.6.6. Method 6 (Corridor grid perpendicular to the trajectory)

In method 6 the path is defined by a set of “way points” that are un-equally spaced, with a horizontal buffer region around the path (defined by a “horizontal buffer Extent”). The coverage returned in response to a successful GetCorridor request would have as its “domain set” each sample point (the points are shown in red). The difference in this example is the orientation of the points relative to the path, i.e. perpendicular to the path. The “result set” is calculated by interpolating from the MetOcean coverage to each of the sampling points (see Figure 22 and Figure 23).



Figure 22 – Corridor grid perpendicular to the trajectory

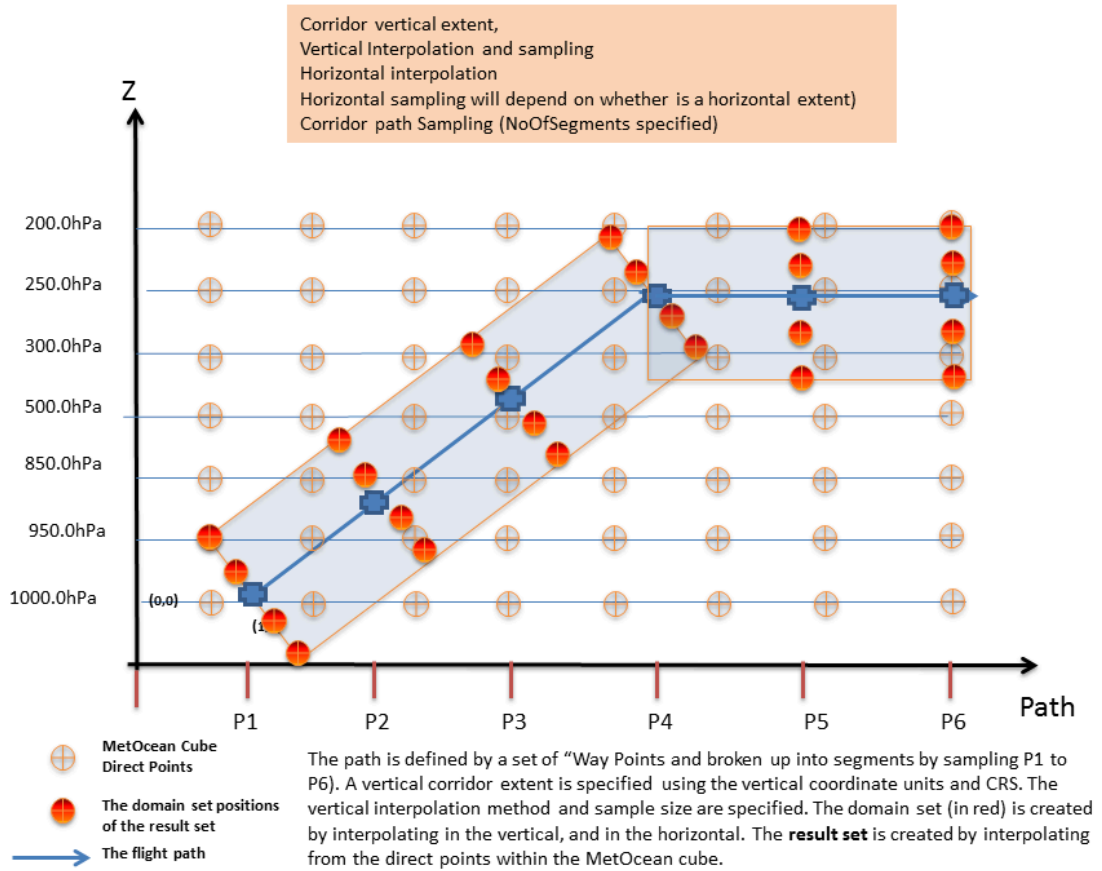
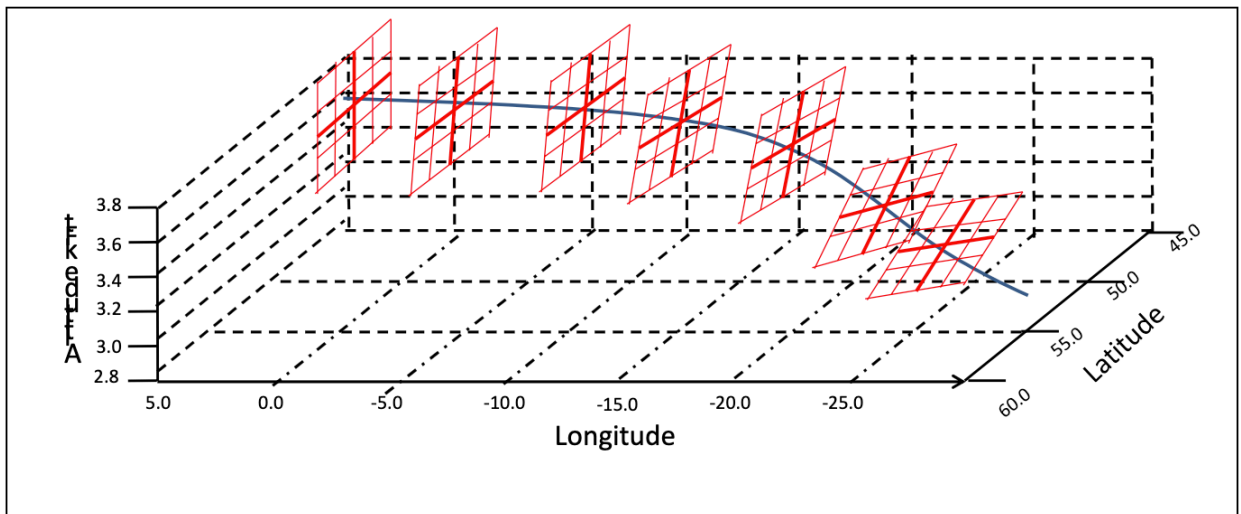


Figure 23 – Corridor grid perpendicular to the trajectory



This figure illustrates the orientation of the corridor axis relative to the path, i.e. it is perpendicular.

### 7.6.7. Encoding the trajectory

Encoding of the trajectory using the [CIS1.1](#) coverage model is quite straight forward and some examples have been included with this document. The alternative encodings, e.g. [covJson](#) are more limited, but will work for simple trajectories. Binary encoding e.g. [NetCDF](#) have been done, but until recently there was no standard way of organising the data. The latest CF standard for sampling may address the issue. The alternative encoding i.e. GRIB2 is not straightforward at all and not easy to interpret.



8

# THE CORE GETCORRIDOR REQUIREMENT (NORMATIVE)

---

# THE CORE GETCORRIDOR REQUIREMENT (NORMATIVE)

## 8.1. REQUIREMENTS CLASS: GETCORRIDOR

This clause establishes the GetCorridor extension core for conformance class `get_Corridor`. Clients and servers supporting the requirements' class support the extraction of a trajectory/corridor from a multidimensional data cube.

### REQUIREMENTS CLASS

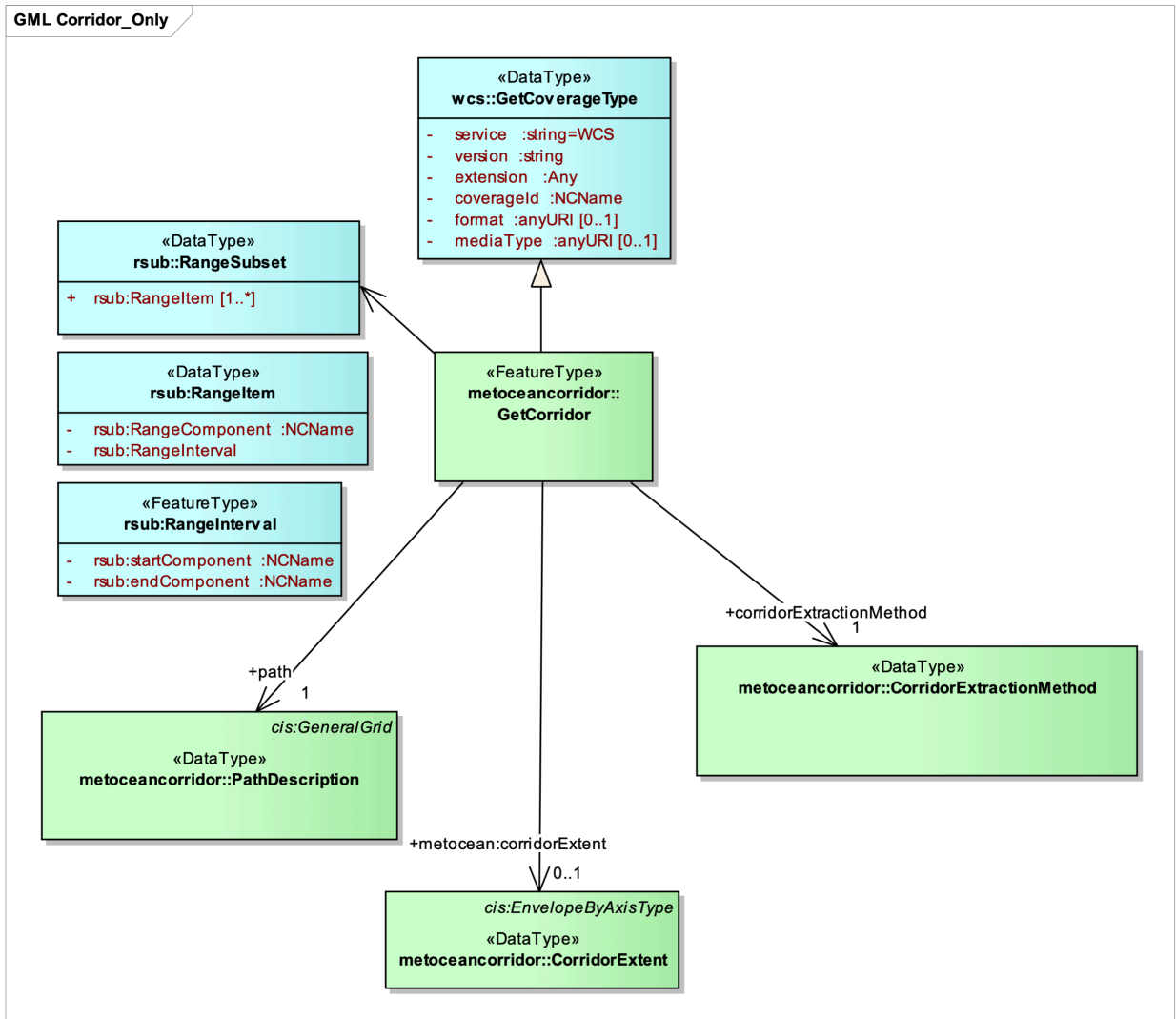
[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor)

Dependency	<a href="http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage">http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage</a>
Dependency	<a href="http://www.opengis.net/spec/WCS_service-extension_range-subsetting/1.0/conf/record-subsetting">http://www.opengis.net/spec/WCS_service-extension_range-subsetting/1.0/conf/record-subsetting</a> ;
Requirement 1	<a href="#">/req/getCorridor/structure</a> A <code>metoceanCorridor:GetCorridor</code> instance <b>shall</b> conform to Figure 24, Table 2 & Table 3
Requirement 2	<a href="#">/req/getCorridor/getCapabilities-response-conformance-class-in-profile</a> A WCS service implementing this extension <b>shall</b> include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor</a>
Requirement 3	<a href="#">/req/getCorridor/request-valid-identifier</a> The <code>coverageId</code> parameter value in a GetCorridor request <b>shall</b> be equal to the identifier of one of the coverages offered by the server addressed.
Requirement 4	<a href="#">/req/getCorridor/acceptable-format</a> If a <code>GetCorridor</code> request contains a <code>format</code> parameter then this parameter <b>shall</b> contain a MIME type identifier occurring in some <code>WCS::formatSupported</code> element of the response to a successful GetCapabilities request to this server
Requirement 5	<a href="#">/req/getCorridor/acceptable-mediaType</a> If a <code>GetCorridor</code> request contains a <code>mediaType</code> parameter then this parameter <b>shall</b> contain a MIME type identifier of fixed value "multipart/related".
Requirement 6	<a href="#">/req/getCorridor/path</a> The GetCorridor request <b>shall</b> contain a valid PathDescription element within the path element.
Requirement 7	<a href="#">/req/getCorridor/range-component</a> The parameter value of the RangeComponent of the <code>wcs:RangeItem</code> element <b>shall</b> contain a parameter that is part of the requested coverage.
Requirement 8	<a href="#">/req/getCorridor/response-encoding</a>

## REQUIREMENTS CLASS

The contents of the response to a successful *GetCorridor* 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 24 – GetCorridor UML class diagram



### 8.1.1. Requirements class overview

The *Get\_Corridor* requirements class defines the structure of the *GetCorridor* operation.

### 8.1.2. Metoceancorridor::GetCorridor

The new operation *GetCorridor* allows for the extraction of a trajectory/corridor and has a number of options that have been described in section 7. The extra conformance classes are used to further define the possible options outlined here in the *getCorridor* conformance class.

The GetCorridor service is derived from wcs:GetCoverage and inherits the version and service elements.

Table 2 – METOCEANCORRIDOR::GetCorridor properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
wcs::coverageId	Identifier of a coverage offered by the service on hand	NCName	one (mandatory)
wcs:mediaType	Optional element indicating the MimeType of the response of a GetCoverage request. Only currently allowed valued is "multipart/related".	anyURI,	wcs:mediaType
wcs: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.	Directly referred to be the GerCorridor element.	zero or one
wcs:format	MIME type identifier of the format in which the coverage returned is encoded	anyURI	zero or one
service	The service name.	string=wcs	required
version	The version number	String (value="2\.\0\.\d+")	required
wcs:extension	Extension element used to hook in additional content e.g. in extensions or application profiles.</documentation>	Any	one or more
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
metoceancorridor:corridorExtractionMethod	The definition of the extraction pattern to be used by the corridor	metoceancorridor:CorridorExtractionMethod	one
metoceancorridor:corridorExtent	The description of the corridor to be used. The corridor is described in section 7 of this document.	metoceancorridor:CorridorExtent	zero or one

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
metoceanCorridor:path	Of type cis:GeneralGridCoverage This defines the path of the trajectory or in the case of a corridor the centre line. Some examples are given (see 8.2.7).	metoceanCorridor:PathDescription	one (mandatory)

### 8.1.3. rsub::RangeSubset

Table 3 – RSub::RangeSubset properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
rsub:RangeItem	List of range components to be extracted	RangeComponent or RangeInterval	one or more (mandatory)
rsub:RangeComponent	Range component name	RangeComponent	one (mandatory)
rsub:RangeInterval	Pair of range interval lower and upper bound	Pair of RangeComponent	one (mandatory)

### 8.1.4. rsub::RangeSubset

Table 4 – RSub::RangeItem properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
rsub:RangeComponent	Range component name	RangeComponent	one (mandatory)

## 8.2. REQUIREMENTS CLASS: PATHDESCRIPTION

### 8.2.1. Requirements class overview

This clause establishes the PathDescription conformance. Clients and servers supporting the requirements class support the extraction of a trajectory/corridor from a multidimensional data cube. This PathDescription is mandatory.

#### REQUIREMENTS CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/PathDescription](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/PathDescription)

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-regular/conf>

Dependency <http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf>

Requirement 9 [/req/PathDescription/structure](#)  
A metoceanCorridor:PathDescription instance **shall** conform to Figure 3, Figure 25 and Table 5, Table 6, Table 7, Table 8, Table 9, and Table 10

Requirement 9b [/req/PathDescription/definition](#)  
The PathDescription **shall** be a derived from of cis:GeneralGrid

Requirement 10 [/req/PathDescription/segment-Definition](#)  
If the SegmentDefinition element is present then either of the two elements segmentPerSector or segmenPerPath **shall** be present.

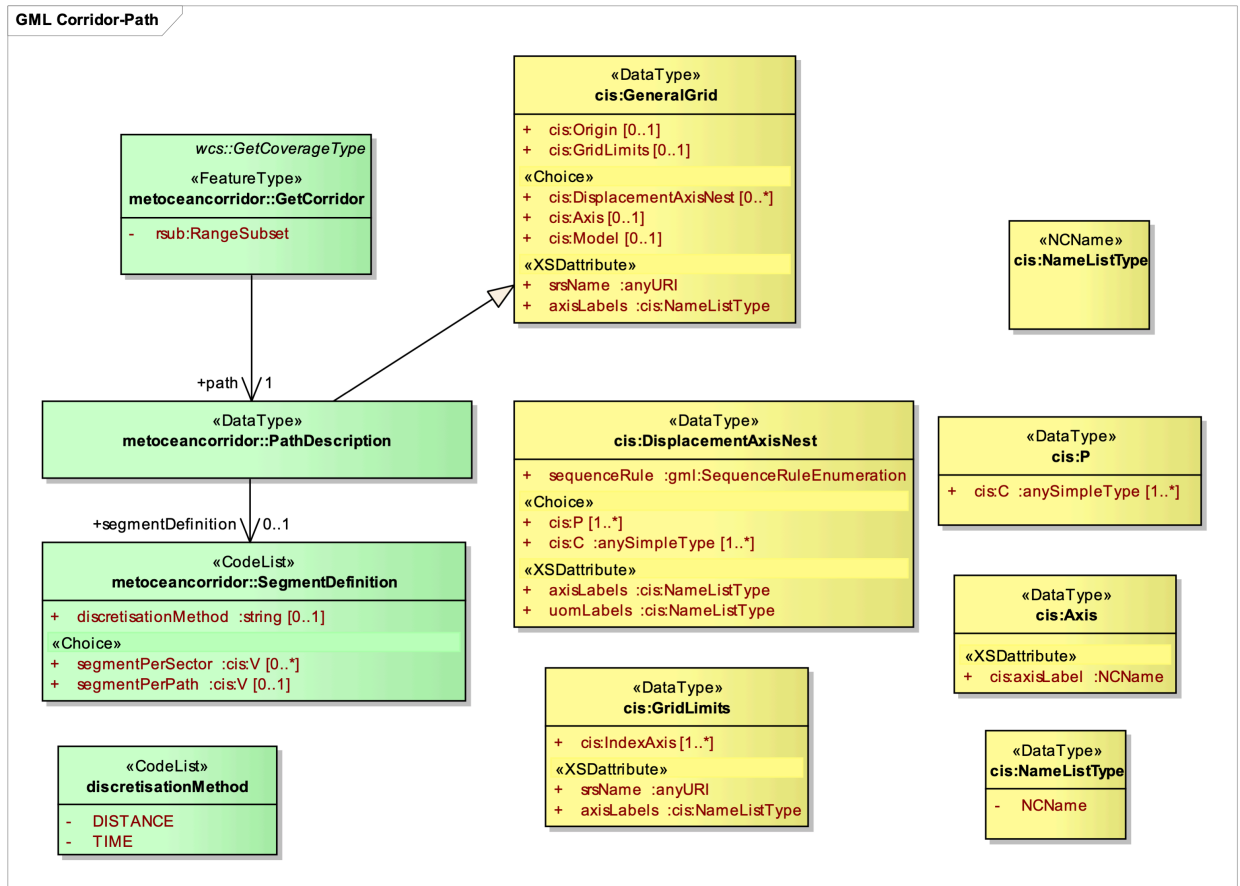
Requirement 11 [/req/PathDescription/getCapabilities-response-segment-definition](#)  
A GetCorridor service implementing the segment extension shall include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response:  
[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/segment-definition](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/segment-definition)

Requirement 12 [/req/PathDescription/segmentPerPath](#)  
The number of the value in the element segmentPerPath **shall** have a value of that is at least one more than the number of "way points".

Requirement 13 [/req/PathDescription/segmentPerSector](#)  
The number of the value in the element segmentPerSector **shall** have a value of two or more.



Figure 25 – CorridorPath UML class diagram



## 8.2.2. PathDescription Properties

The path description is derived from CIS:GeneralGrid and is extended to include information regarding segmentation. Note that only those elements used by this conformance class are included.

Table 5 – METOCEANCORRIDOR::PathDescription properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
metoceanCorridor: segmentDefinition	The definition of how the path may, if required, be broken up into segments.	metoceanCorridor:: SegmentDefinition	zero or one (optional)
cis:DisplacementAxisNest	DisplacementAxisNest combines several axes to a single “nest” where the coordinates are enumerated individually for each direct position.	cis:DisplacementAxisNestType	One (mandatory)

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
cis:GridLimits	the Grid limits in the CIS::Axis structure contains information about the grid boundaries in the coverage's CRS.	cis:GridLimitsType	One (mandatory)
srsName	URL identifying the CRS of the coordinates in this coverage	anyURI	One
uomLabels	units of measure in which values along the axes are expressed	cis:NameListType	One or more
axisLabels	Axes involved in the "nest" of displaced direct positions; these axes shall form a subset of the CIS::General-Grid axisLabels	cis:NameListType	One or more

### 8.2.3. SegmentDefinition Properties

The path (aka trajectory) may be divided into segments (see Figure 4, Figure 5, Figure 6, Figure 7 and Figure 8). The description is given in section 7.5.

Table 6 – METOCEANCORRIDOR::SegmentDefinition properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
metoceanCorridor:discretisationMethod	The mode by which the path is broken up into segments, if required.	metoceanCorridor::DiscretisationMethodValue	One (mandatory)
metoceanCorridor:segmentPerSector	If the path is divided by sector (i.e. a sector is the part of the path from one "way point" to another, then for each sector this value specifies the number of segments per sector	cis:V	Zero or many (optional)
metoceanCorridor:segmentPerPath	If the whole path is divided	cis:V	Zero or one (optional)

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
	equally across its domain then this specifies the number of segments from start to finish.		

Table 7 – METOCEANCORRIDOR::cis:V properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
cis:V	Atomic Range Values consist of single, atomic values; these can be of any kind, such as numeric, date, or categorical (e. g., “true” and “false”).	anySimpleType	one

### 8.2.4. DiscretisationMethodValue Properties

The path may be divided up into segments using one of two methods. The first by distance, the second method is similar, but the division is done equally in time.

Table 8 – METOCEANCORRIDOR::DiscretisationMethodValue Properties

CODE ITEM	DEFINITION
DISTANCE	The path/sector is divided equally by distance
TIME	The path/ sector is divided equally by time

### 8.2.5. DisplacementAxisNest Properties

The CIS::DisplacementAxisNest combines several axes to a single “nest” where the coordinates are enumerated individually for each direct position (Note this come from CIS1.1).

Table 9 – CIS::DisplacementAxisNest

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
axisLabels	Axes involved in the “nest” of displaced direct positions; these axes shall form a subset of the CIS::GeneralGrid axisLabels	cis:NameListType	One or more
uomLabels		cis:NameListType	One or more

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
directPositions	Array of direct positions along this axis, linearized according to the sequence rule or, if missing, along the GML 3.2.1 default	string	One or more (mandatory)
sequenceRule	Description of the array linearization in directPositions, according to the GML 3.2.1 sequence rule	GML:: sequenceRule	Zero or one (optional)

## 8.2.6. GridLimits Properties

The limits of the underlying array are given by the CIS::gridLimits component (Note this come from CIS1.1)

Table 10 – CIS::GridLimits

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)
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	cis:NameListType	One or more
axisExtent	Sequence of extents of the grid along a specific axis, exactly one for each axis defined in the CRS referenced in srsName	CIS:: AxisExtent	One or more (mandatory)

## 8.2.7. An example of a Path encoding

```
<metoceanCorridor:path
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:metoceanCorridor="http://www.opengis.net/wcs/metoceanProfile_getCorridor/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:scal="http://www.opengis.net/WCS_service-extension_scaling/1.0"
xmlns:int="http://www.opengis.net/WCS_service-extension_interpolation/1.0"
xmlns:cis="http://www.opengis.net/cis/1.1/gml"
xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<metoceanCorridor:PathDescription srsName="http://www.opengis.net/def/crs-compound?
```

```

1=http://www.opengis.net/def/crs/EPSG/0/4326&
2=http://http://www.opengis.net/def/crs/OGC/0/AnsiDate&
3= http://codes.wmo.int/grib2/codeflag/4.5/_102/Specific_altitude_above_mean_sea_level"
axisLabels="Lat Lon Time Specific_altitude_above_mean_sea_level">
<cis:DisplacementAxisNest axisLabels="Lat Lon Time Specific_altitude_above_mean_sea_level"
uomLabels="deg deg ISO8601 ft">
<cis:P><cis:C>45.0</cis:C><cis:C>5.0</cis:C><cis:C>2015-05-15T00:00:00Z</cis:C><cis:C>3600</cis:
C></cis:P>
<cis:P><cis:C>46.0</cis:C><cis:C>0.0</cis:C><cis:C>2015-05-15T00:15:00Z</cis:C><cis:C>3500</cis:
C></cis:P>
<cis:P><cis:C>47.0</cis:C><cis:C>-5.0</cis:C><cis:C>2015-05-15T00:30:00Z</cis:C><cis:C>34200</cis:
C></cis:P>
<cis:P><cis:C>48.0</cis:C><cis:C>-10.0</cis:C><cis:C>2015-05-15T00:45:00Z</cis:C><cis:C>3400</cis:
C></cis:P>
<cis:P><cis:C>49.0</cis:C><cis:C>-15.0</cis:C><cis:C>2015-05-15T01:00:00Z</cis:C><cis:C>3200</cis:
C></cis:P>
<cis:P><cis:C>51.0</cis:C><cis:C>-20.0</cis:C><cis:C>2015-05-15T01:15:00Z</cis:C><cis:C>3000</cis:
C></cis:P>
<cis:P><cis:C>52.0</cis:C><cis:C>-25.0</cis:C><cis:C>2015-05-15T01:30:00Z</cis:C><cis:C>2000</cis:
C></cis:P>
</cis:DisplacementAxisNest>
<cis:GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index1D" axisLabels="i">
<cis:IndexAxis axisLabel="i" lowerBound="0" upperBound="6"/>
</cis:GridLimits>

<metoceanCorridor:SegmentDefinition>
<metoceanCorridor:discretisationMethod>DISTANCE</metoceanCorridor:discretisationMethod>
<metoceanCorridor:segmentsPerSector>
<cis:V>3</cis:V>
<cis:V>5</cis:V>
<cis:V>4</cis:V>
<cis:V>6</cis:V>
<cis:V>4</cis:V>
<cis:V>5</cis:V>
</metoceanCorridor:segmentsPerSector>
</metoceanCorridor:SegmentDefinition>
</metoceanCorridor:PathDescription>
</metoceanCorridor:path>

```

## 8.3. REQUIREMENTS CLASS: CORRIDOREXTENT

This clause establishes the CorridorExtent conformance class. This class describes the corridor, that may be specified in addition to the simple trajectory. The corridor is multi-dimensional and typically has a width and an optional height.

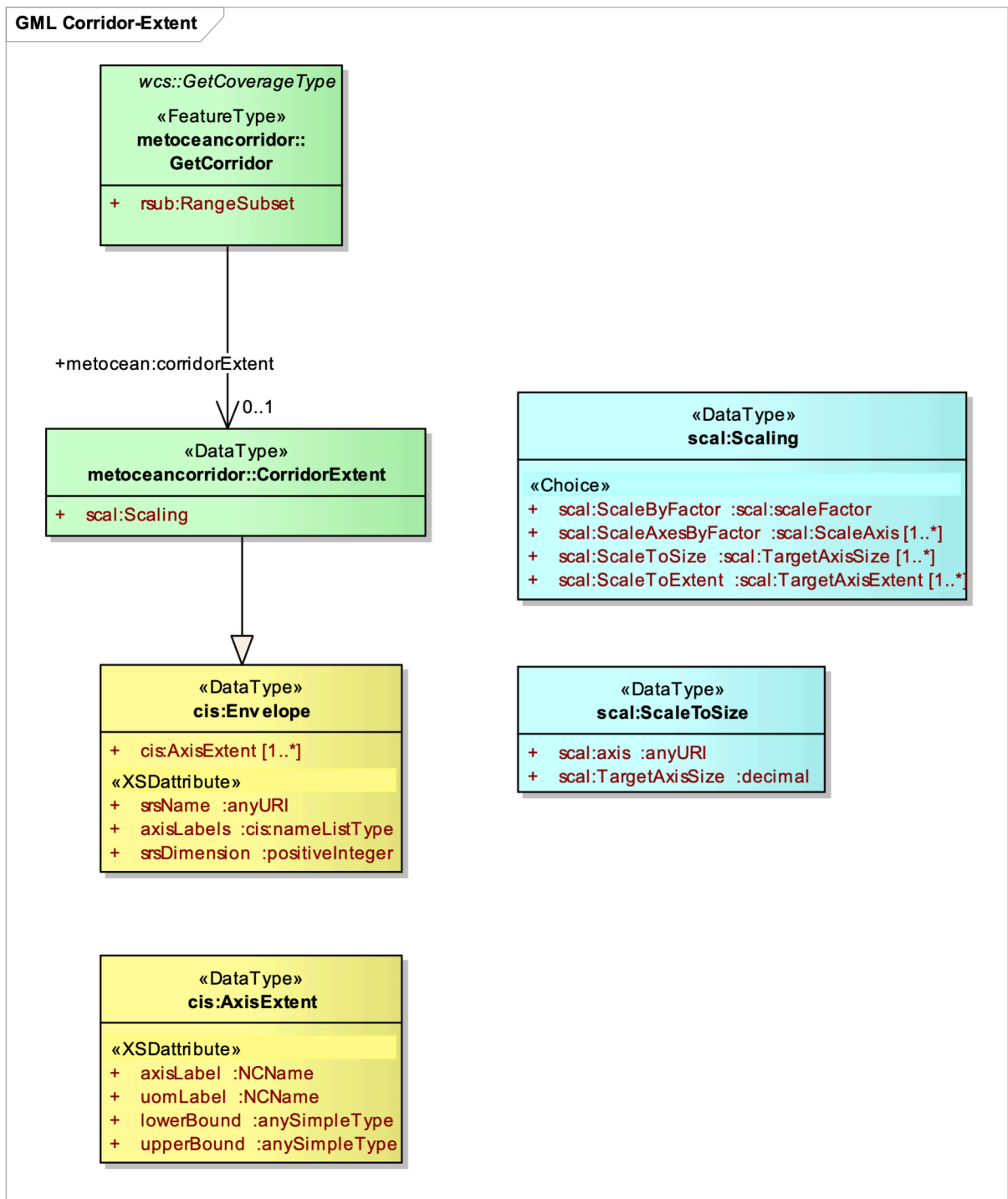
### REQUIREMENTS CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/CorridorExtent](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtent)

## REQUIREMENTS CLASS

Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Dependency	<a href="http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage">http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf">http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/grid-regular/conf">http://www.opengis.net/spec/CIS/1.1/conf/grid-regular/conf</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf">http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf</a>
Requirement 14	<b>req/CorridorExtent/structure</b> A metoceanCorridor:CorridorExtent instance <b>shall</b> conform to Figure 26, Table 11, Table 12, Table 13 and Table 14.
Requirement 15	<b>req/CorridorExtent/axis-names</b> The scal:axis names <b>shall</b> be one of "Corridor_Width", "Corridor_Height"
Requirement 16	<b>req/CorridorExtent/axis-names-duplicates</b> The CorridorExtent element <b>shall</b> not have duplicate scal:axis names.
Requirement 17	<b>req/CorridorExtent/TargetAxisSize</b> The element TargetAxisSize <b>shall</b> contain valid elements scal:axis and scal:targetSize
Requirement 18	<b>req/CorridorExtent/AxisExtent</b> The scal:Axis element, if present, <b>shall</b> have a corresponding every cis:AxisExtent axisLabel with the an attribute that has the same value as the scal:Axis element.

Figure 26 – CorridorExtent UML class diagram



### 8.3.1. Requirements class overview

In this document a trajectory with a lateral and (Optional) vertical extent is known as a corridor. The CorridorExtent Properties define the corridor axes and therefore dimensionality. The corridor axis known as “Corridor\_Width” is always in the horizontal whereas the

“Corridor\_Height” axis may either be perpendicular to the trajectory or aligned with the metocean cube vertical axis (see Figure 10 and Figure 11). The size of the corridor is specified by the AxisExtent element, see below for some examples.

### 8.3.2. Examples of CorridorExtent

**NOTE:** 5703 is vertical height of the US

#### Example 1

In this example the corridor has two dimensions; the width is 10 Kilometres wide and the height is 20 Kilometres both dimensions are specified relative to the trajectory. Note that the vertical coordinate system i.e. EPSG:5703 is the vertical coordinate system for Mexico and the States (USA).

```
<metoceanCorridor:corridorExtent
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:metoceanCorridor="http://www.opengis.net/wcs/metoceanProfile_getCorridor/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:scal="http://www.opengis.net/WCS_service-extension_scaling/1.0"
xmlns:int="http://www.opengis.net/WCS_service-extension_interpolation/1.0"
xmlns:cis="http://www.opengis.net/cis/1.1/gml"
xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<metoceanCorridor:CorridorExtent srsName="http://www.opengis.net/def/crs-compound?
1=http://www.opengis.net/def/crs/EPSSG/0/4326 &
2=http://http://www.opengis.net/def/crs/EPSSG/0/5703"
axisLabels="Lat Lon Corridor" srsDimension="3">
<cis:AxisExtent axisLabel="Corridor_Width" uomLabel="km" lowerBound="-5.0" upperBound="5.0"/
>
<cis:AxisExtent axisLabel="Corridor_Height" uomLabel="km" lowerBound="-10.0" upperBound="10.
0"/>
<scal:Scaling xmlns:scal="http://www.opengis.net/WCS_service-extension_scaling/1.0">
<scal:ScaleToSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Width</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Height</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
</scal:ScaleToSize>
</scal:Scaling>
</metoceanCorridor:CorridorExtent>
</metoceanCorridor:corridorExtent>
```

#### Example 2

In this example vertical reference system for “Corridor\_Height” uses the GRIB2 definition for isobaric surfaces. The unit of measure in is Hectopascals.

```
<metoceanCorridor:corridorExtent
```



```

xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:metoceanCorridor="http://www.opengis.net/wcs/metoceanProfile_getCorridor/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:scal="http://www.opengis.net/WCS_service-extension_scaling/1.0"
xmlns:int="http://www.opengis.net/WCS_service-extension_interpolation/1.0"
xmlns:cis="http://www.opengis.net/cis/1.1/gml"
xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<metoceanCorridor:CorridorExtent srsName="http://www.opengis.net/def/crs-compound?
1= http://www.opengis.net/def/crs/EPSSG/0/4326 &
2=http://www.codes.wmo.int/GRIB2/table4.5/IsobaricSurface"
axisLabels=" Corridor_Width Corridor_Height " srsDimension="2">
<cis:AxisExtent axisLabel="Corridor_Width" uomLabel="m" lowerBound="-5.0" upperBound="5.0"/>
<cis:AxisExtent axisLabel="Corridor_Height" uomLabel="hPa" lowerBound="-1000.0" upperBound=
"1000.0"/>
<scal:Scaling>
<scal:ScaleToSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Width</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Height</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
</scal:ScaleToSize>
</scal:Scaling>
</metoceanCorridor:CorridorExtent>
</metoceanCorridor:corridorExtent>

```

### Example 3

In this example vertical reference system for “Corridor\_Height” uses the GRIB2 definition for pressure altitude WMO GRIB2 table4.2-0-3. The unit of measure in this example is metres.

```

<metoceanCorridor:corridorExtent
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:metoceanCorridor="http://www.opengis.net/wcs/metoceanProfile_getCorridor/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:scal="http://www.opengis.net/WCS_service-extension_scaling/1.0"
xmlns:int="http://www.opengis.net/WCS_service-extension_interpolation/1.0"
xmlns:cis="http://www.opengis.net/cis/1.1/gml"
xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
<metoceanCorridor:CorridorExtent srsName="http://www.opengis.net/def/crs-compound?
1= http://www.opengis.net/def/crs/EPSSG/0/4326 &
2=http://www.codes.wmo.int/GRIB2/table4.2-0-3/Pressure Altitude"
axisLabels="Corridor_Width Corridor_Height" srsDimension="2">
<cis:AxisExtent axisLabel="Corridor_Width" uomLabel="m" lowerBound="-5.0" upperBound="5.0"/>
<cis:AxisExtent axisLabel="Corridor_Height" uomLabel="m" lowerBound="-300.0" upperBound=
"300.0"/>
<scal:Scaling>
<scal:ScaleToSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Width</scal:axis>

```

```

<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Height</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
</scal:ScaleToSize>
</scal:Scaling>
</metoceanCorridor:CorridorExtent>
</metoceanCorridor:corridorExtent>

```

### 8.3.3. CorridorExtent Properties

Table 11 – METOCEANCORRIDOR::CorridorExtent 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	string	One or more (mandatory)
srsDimension	Dimension (number of axes) of the grid	positiveInteger	One (mandatory)
axisExtent	Sequence of extents of the grid along a specific axis, exactly one for each axis defined in the CRS referenced in srsName	CIS::AxisExtent	One or more (mandatory)
scal:Scaling	Scaling can be expressed by indicating, per domain axis given, the target domain extent to which the coverage should be rescaled; this leaves the lower bound of the grid unchanged while the upper bound is adjusted accordingly. Axes not mentioned remain unaffected.	scal:ScaleToSize	One (mandatory)

### 8.3.4. Requirements class overview

The Axis Extent comes from the Coverage Implementation Schema (CIS).

Table 12 – CIS::AxisExtent Properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
cis:axisLabel	Shorthand axis identifier with scope given by the coverage document	NCName	One (mandatory)
cis:uomLabel	Shorthand identifier of the Unit of Measure used on this axis (as indicated in the CRS definition for this axis)	NCName	One (mandatory)
cis:lowerBound	Lowest coordinate along this axis	anySimpleType	One (mandatory)
cis:upperBound	Highest coordinate along this axis	anySimpleType	One (mandatory)

### 8.3.5. Requirements class overview

The number of points defined in the corridor axes is set by using the scal:ScaleByFactor, element value.

Table 13 – SCAL::Scaling

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
scal:ScaleByFactor	Scale factor for this axis	scal:scaleFactor	one (mandatory)
scal:ScaleAxisByFactor	ScaleAxes for the Scaling	scal:ScaleAxis	one (mandatory)
scal:ScaleToSize	Target size of the coverage in the axis indicated for the Scaling	scal:TargetAxisSize	one (mandatory)
scaleToExtent	ScaleToExtent for the Scaling	scal:TargetAxisExtent	one (mandatory)

### 8.3.6. Requirements class overview

The number of points defined in the corridor axes is set by the scal:targetSize.

Table 14 – SCAL::ScaleToSize Properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
axis	Axis to which scaling is to be applied	anyURI	one (mandatory)
scal:targetSize	Target size of the coverage in the axis indicated	int	one (mandatory)

## 8.4. REQUIREMENTS CLASS: CORRIDOREXTRACTIONMETHOD

This clause establishes the GetCorridor extraction methods that are described in section 7.6.

### REQUIREMENTS CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/CorridorExtractionMethod](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtractionMethod)

Dependency [http://www.opengis.net/spec/WCS\\_service-extension\\_interpolation/1.0/conf/interpolation](http://www.opengis.net/spec/WCS_service-extension_interpolation/1.0/conf/interpolation)

Dependency <http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage>

Dependency [http://www.opengis.net/spec/WCS\\_service-extension\\_interpolation/1.0/req/interpolation-per-axis](http://www.opengis.net/spec/WCS_service-extension_interpolation/1.0/req/interpolation-per-axis)

Requirement 19 </req/CorridorExtractionMethod/structure>  
A metoceanCorridor:CorridorExtractionMethod instance **shall** conform to Figure 26, Table 15, Table 16, Table 17, Table 18 and Table 19

Requirement 20 </req/CorridorExtractionMethod/Value>  
The CorridorExtractionMethod instance **shall** contain a valid metoceanCorridor:CorridorExtractionMethodValue element that conforms with Table 8.

Requirement 21 </req/CorridorExtractionMethod/interpolationMethod>  
In those methods that require interpolation i.e. where the CorridorExtractionValue is either Trajectory\_Point\_Interpolation or Corridor\_Point\_Interpolation then the request **shall** contain valid int:Interpolation parameter.

Requirement 22 </req/CorridorExtractionMethod/interpolation-axes>  
In those methods that require interpolation i.e. where the CorridorExtractionValue is either Trajectory\_Point\_Interpolation or Corridor\_Point\_Interpolation then the Interpolation:interpolationAxes element in a GetCorridor request, **shall** consist of a non-empty sequence of Interpolation::InterpolationPerAxis elements.

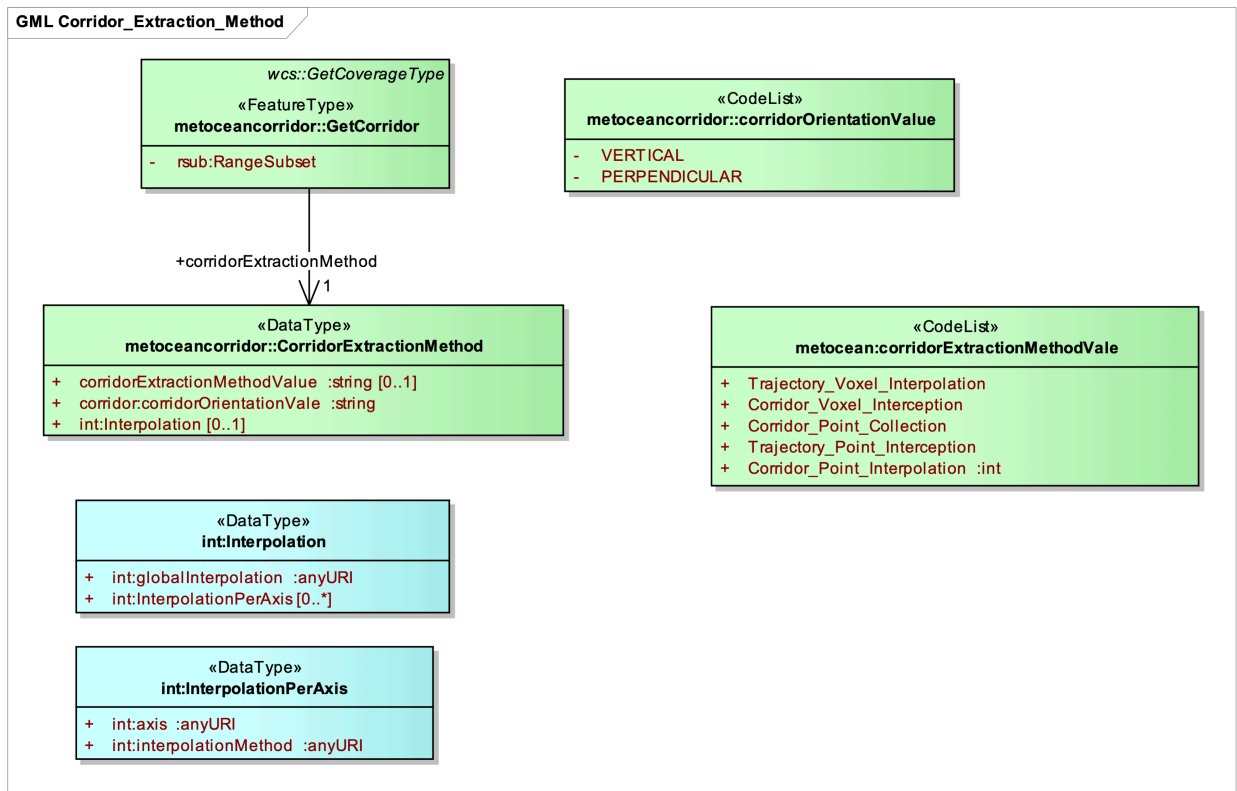
Requirement 23 </req/CorridorExtractionMethod/TrajectoryVoxelInterception> If CorridorExtractionValue is set to 'TrajectoryVoxelInterception' then the GetCoverage request containing **shall** be obtained by applying the method outlined in 7.6.1 during the preparation of the response.

Requirement 24 [/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/TrajectoryVoxel\\_Interception](/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/TrajectoryVoxel_Interception) A WCS service implementing requirements class [/req/CorridorExtractionMethod/TrajectoryVoxel\\_Interception](/req/CorridorExtractionMethod/TrajectoryVoxel_Interception) of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities

## REQUIREMENTS CLASS

Requirement 25	<p>response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/TrajectoryVoxel_Interception">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/TrajectoryVoxel_Interception</a></p> <p><b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Voxel_Interception</b> A WCS service implementing requirements class/req/CorridorExtractionMethod/Corridor_Voxel_Interception of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Voxel_Interception">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Voxel_Interception</a></p>
Requirement 26	<p><b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Point_Collection</b> A WCS service implementing requirements class/req/CorridorExtractionMethod/Corridor_Voxel_Interception of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Collection">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Collection</a></p>
Requirement 27	<p><b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Trajectory_Point_Interpolation</b> A WCS service implementing requirements class/req/CorridorExtractionMethod/Trajectory_Point_Interpolation of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Trajectory_Point_Interpolation">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Trajectory_Point_Interpolation</a></p>
Requirement 28	<p><b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Point_Interpolation</b> A WCS service implementing requirements class/req/CorridorExtractionMethod/Corridor_Point_Interpolation of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Interpolation">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Interpolation</a></p>

Figure 27 – CorridorExtractionMethod UML class diagram



### 8.4.1. Requirements class overview

The CorridorExtractionMethod requirements class defines the structure of the CorridorExtractionMethod.

### 8.4.2. CorridorExtractionMethod Properties

These properties detail how the extraction of the data within the corridor is defined. The detailed description is given in section 7.

Table 15 – METOCEANCORRIDOR::CorridorExtractionMethod properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
metocean:corridor:: CorridorExtractionMethod	The value of the CorridorExtractionValue will control the methods of extraction outlined in section 7	metocean:corridor:: CorridorExtractionValue	One (mandatory)

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
int:Interpolation	Interpolation method to be applied to the named axis GetCorridor result preparation	int:InterpolationPerAxis	Optional
metoceanCorridor::corridorOrientationValue	The corridor height axis may either be perpendicular to the trajectory or aligned with the metocean cube vertical axis and this is determined by the value of this element.	metoceanCorridor::CorridorOrientationValue	

### 8.4.3. The CorridorExtractionValue Properties

This table sets out the permitted values for the orientation of the corridor’s vertical axis with respect to the trajectory.

Table 16 – METOCEANCORRIDOR::CorridorOrientationValue Properties

CODE ITEM	DEFINITION
VERTICAL	Corridor vertical aligned with the local vertical coordinate
HORIZONTAL	Corridor vertical aligned with the perpendicular to the path.

### 8.4.4. metoceanCorridor::CorridorExtractionMethodValue Properties

There are five GetCorridor extraction patterns and these are listed in Table 17. The requirements’ for each are listed separately.

Table 17 – METOCEANCORRIDOR::CorridorExtractionMethodValue Properties

CODE ITEM	DEFINITION
Trajectory_Voxel_Interception	Defined as method 1 in section 7
Corridor_Voxel_Interception	Defined as method 2 in section 7
Corridor_Point_Collection	Defined as method 3 in section 7
Trajectory_Point_Interpolation	Defined as method 4 in section 7
Corridor_Point_Interpolation	Defined as method 5 and 6 in section 7

## 8.4.5. int:Interpolation Properties

Any method that requires interpolation must set the following properties

Table 18 – INT::Interpolation Properties

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
int:globalInterpolation	Interpolation method to be applied on all axes during getCorridor result preparation	anyURI	one
Int:InterpolationPerAxis	Interpolation method to be applied on specific as axis for the getCorridor operation	InterpolationPerAxis	Zero to many

## 8.4.6. int:InterpolationPerAxis

Table 19 – INT::InterpolationPerAxis

NAME	DEFINITION	DATA TYPES AND VALUES	MULTIPLICITY
axis	Coverage axis along which the interpolation method is to be applied	anyURI	one (mandatory)
InterpolationMethod	Interpolation method to be applied along the specified axis during GetCoverage result preparation	anyURI	one (mandatory)

```
<?xml version="1.0" encoding="UTF-8"?>
<metoceanCorridor:GetCorridor xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:wcs="http://www.opengis.net/wcs/2.0"
xmlns:metocean="http://www.opengis.net/wcs/metoceanProfile/1.0"
xmlns:metoceanCorridor="http://www.opengis.net/wcs/metoceanProfile_getCorridor/1.0"
xmlns:wcsCRS="http://www.opengis.net/wcs_service-extension_crs/1.0"
xmlns:scal="http://www.opengis.net/WCS_service-extension_scaling/1.0"
xmlns:int="http://www.opengis.net/WCS_service-extension_interpolation/1.0"
xmlns:gml="http://www.opengis.net/gml/3.2"
xmlns:cis="http://www.opengis.net/cis/1.1/gml"
xmlns:rsub="http://www.opengis.net/wcs/range-subsetting/1.0"
xmlns:gmlrgrid="http://www.opengis.net/gml/3.3/rgrid"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
service="WCS" version="2.1.0"
```





```

<cis:P><cis:C>51.0</cis:C><cis:C>-20.0</cis:C><cis:C>2015-05-15T01:15:00Z</cis:C><cis:C>3000</cis:
C></cis:P>
<cis:P><cis:C>52.0</cis:C><cis:C>-25.0</cis:C><cis:C>2015-05-15T01:30:00Z</cis:C><cis:C>2000</cis:
C></cis:P>
</cis:DisplacementAxisNest>
<cis:GridLimits srsName="http://www.opengis.net/def/crs/OGC/0/Index1D" axisLabels="i">
<cis:IndexAxis axisLabel="i" lowerBound="0" upperBound="6"/>
</cis:GridLimits>

<metoceanCorridor:SegmentDefinition>
<metoceanCorridor:discretisationMethod>DISTANCE</metoceanCorridor:discretisationMethod>
<metoceanCorridor:segmentsPerSector>
<cis:V>3</cis:V>
<cis:V>5</cis:V>
<cis:V>4</cis:V>
<cis:V>6</cis:V>
<cis:V>4</cis:V>
<cis:V>5</cis:V>
</metoceanCorridor:segmentsPerSector>
</metoceanCorridor:SegmentDefinition>
</metoceanCorridor:PathDescription>
</metoceanCorridor:path>

<metoceanCorridor:CorridorExtent>
<metoceanCorridor:CorridorExtent srsName="http://www.opengis.net/def/crs-compound?
1=http://www.opengis.net/def/crs/EPSSG/0/4326&
2=http://http://www.opengis.net/def/crs/Corridor_Height"
axisLabels="Lat Lon Corridor" srsDimension="3">
<cis:AxisExtent axisLabel="Corridor_Width" uomLabel="deg" lowerBound="-5.0" upperBound="5.0"/
>
<cis:AxisExtent axisLabel="Corridor_Height" uomLabel="ft" lowerBound="-1000.0" upperBound=
"1000.0"/>
<scal:Scaling>
<scal:ScaleToSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Width</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
<scal:TargetAxisSize>
<scal:axis>Corridor_Height</scal:axis>
<scal:targetSize>5</scal:targetSize>
</scal:TargetAxisSize>
</scal:ScaleToSize>
</scal:Scaling>
</metoceanCorridor:CorridorExtent>
</metoceanCorridor:CorridorExtent>

<metoceanCorridor:CorridorExtractionMethod>
<metoceanCorridor:CorridorExtractionMethod>
<metoceanCorridor:CorridorExtractionMethodValue>Trajectory_Point_Interpolation</
metoceanCorridor:CorridorExtractionMethodValue>
<metoceanCorridor:CorridorOrientationValue>VERTICAL</metoceanCorridor:
CorridorOrientationValue>
<int:Interpolation>

```

```

<int:globalInterpolation>http://www.opengis.net/def/interpolation/OGC/1/linear</int:
globalInterpolation>
<int:InterpolationPerAxis>
<int:axis>Corridor_Path</int:axis>
<int:interpolationMethod>http://www.opengis.net/def/interpolation/OGC/1/linear</int:
interpolationMethod>
</int:InterpolationPerAxis>
<int:InterpolationPerAxis>
<int:axis>Corridor_Width</int:axis>
<int:interpolationMethod>http://www.opengis.net/def/interpolation/OGC/1/cubic</int:
interpolationMethod>
</int:InterpolationPerAxis>
<int:InterpolationPerAxis>
<int:axis>Corridor_Height</int:axis>
<int:interpolationMethod>http://www.opengis.net/def/interpolation/OGC/1/barycentric</int:
interpolationMethod>
</int:InterpolationPerAxis>
</int:Interpolation>
</metoceanCorridor:CorridorExtractionMethod>
</metoceanCorridor:corridorExtractionMethod>
</metoceanCorridor:GetCorridor>

```

## 8.5. REQUIREMENTS CLASS: GETCORRIDOR-POST-XML

This requirements class specifies how the GetCorridor operation is provided in WCS servers that implement the HTTP/POST using XML request body protocol binding.

### REQUIREMENTS CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-post-xml](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-post-xml)

Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Dependency	WCS 2.1 protocol extension XML-POST (OGC 09-148r1)
Requirement 29	<a href="/req/getCorridor-post-xml/mandatory">/req/getCorridor-post-xml/mandatory</a> Implementations of this GetCorridor extension that support the GetCorridor post-xml requirements class <b>shall</b> support the WCS 2.1 protocol extension XML-POST [OGC 09-148r1].
Requirement 30	<a href="/req/getCorridor-post-xml/conformance-class-in-profile">/req/getCorridor-post-xml/conformance-class-in-profile</a> Implementations of this GetCorridor extension that support the GetCorridor-xml-post requirements class <b>shall</b> include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-post-xml">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-post-xml</a>
Requirement 31	<a href="/req/getCorridor-post-xml/getCorridor-request-structure">/req/getCorridor-post-xml/getCorridor-request-structure</a> A WCS server implementing the XML/POST protocol binding extension <b>shall</b> encode

## REQUIREMENTS CLASS

request body of the GetCorridor operation using an XML document of type metocean:GetCorridor and described in this document

## 8.6. REQUIREMENTS CLASS: GETCORRIDOR-SIMPLE

A KVP binding for a full GetCorridor is complex and in reality unlikely to be used. The majority of use cases will simply be for a trajectory, i.e. a corridor with not lateral or vertical extent. This requirements class will define what is, and what is not, to be included for a trajectory extraction.

## REQUIREMENTS CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-simple](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple)

Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Requirement 32	<a href="#">/req/getCorridor-simple/corridor-extent</a> The GetCorridor request <b>shall</b> not contain a metoceancorridor:corridorExtent element.
Requirement 33	<a href="#">/req/getCorridor-simple/extraction-method</a> The metoceancorridor:corridorExtractionMethodValue <b>shall</b> only contain the values Trajectory_Point_Interpolation or Trajectory_Voxel_Interpolation
Requirement 34	<a href="#">/req/getCorridor-simple/conformance-class-in-profile</a> A GetCorridor service implementing only this conformance class <b>getCorridor-simple</b> of this extension <b>shall</b> include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple</a>

## 8.7. REQUIREMENTS CLASS: GETCORRIDOR-SIMPLE-KVP

This requirements class specifies how the GetCorridor operation is provided in WCS servers implementing a KVP binding.

## REQUIREMENTS CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-simple-get-kvp](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple-get-kvp)

## REQUIREMENTS CLASS

Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Dependency	WCS 2.0 protocol extension GET/KVP (OGC 09-147r3).
Requirement 35	<p><b>/req/getCorridor-simple-get-kvp/conformance-class-in-profile</b></p> <p>Implementations of this getCorridor-simple-get-kvp extension that supports the DescribeCoverageCollection-get-kvp requirements class <b>shall</b> include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple-get-kvp">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple-get-kvp</a></p>
Requirement 36	<b>/req/getCorridor-simple-get-kvp/request-structure.</b> A WCS server implementing both this get-kvp protocol binding extension <b>shall</b> encode the DescribeCoverageCollection operation request as specified in Table 20 .

The *DescribeCoverageCollection* request for HTTP/GET using KVP is defined below.

Table 20 – METOCEANCORRIDOR::GetCorridor request URL encoding

Name	Definition	Data type	Multiplicity
service	Service identifier	String, fixed to "WCS"	one (mandatory)
version	WCS service version indicator	String	one (mandatory)
request	Request type name	String, fixed to "GetCorridor"	one (mandatory)
CoverageId	Coverage identifiers	Comma-separated NCName list	one (mandatory)
PathSpec	Description of the trajectory	Path Spec as defined in below	one (mandatory)
format	MIME type identifier of the format in which the coverage returned is encoded	anyURI	zero or one
mediaType	If present, enforces a multipart encoding	anyURI	zero or one
RangeComponent	The name(s) of the required parameters	NCName	One or more (mandatory)
segmentsPerSector		int	

Each PathSpec shall adhere to this EBNF syntax:

```

Pathspec: dimension []#( # PointSpec []#)#
dimension: _NCName_
PointSpec: AxisDefn
AxisDefn InterpolationMethod[]#,#{point}
InterpolationMethod: NCName
point _number_ | " token " // " = double quote = ASCII 0x42

```



# ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)



# ANNEX A (NORMATIVE) CONFORMANCE CLASS ABSTRACT TEST SUITE (NORMATIVE)

---

## A.1. CONFORMANCE CLASS: GETCORRIDOR

---

CONFORMANCE CLASS	
<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor</a>	
Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Dependency	<a href="http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage">http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage</a>
Dependency	<a href="http://www.opengis.net/spec/WCS_service-extension_range-subsetting/1.0/conf/record-subsetting">http://www.opengis.net/spec/WCS_service-extension_range-subsetting/1.0/conf/record-subsetting</a>
Test	<a href="#">/conf/getCorridor/structure</a>
Requirement 1	<a href="#">/req/getCorridor/structure</a>
Test purpose	A metoceanCorridor:GetCorridor instance <b>shall</b> conform to Figure 24, Table 2 & Table 3
Test method	Send a valid <i>GetCorridor</i> request to server under test which conforms to the references in the requirement. Check that the response is not an exception.
Test Type	Conformance
Test	<a href="#">/conf/getCorridor/getCapabilities-response-conformance-class-in-profile</a>
Requirement 2	<a href="#">/req/getCorridor/getCapabilities-response-conformance-class-in-profile</a>
Test purpose	A WCS service implementing this extension <b>shall</b> include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor</a>

## CONFORMANCE CLASS

	<b>Test method</b>	Send a valid <i>GetCorridor</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
	<b>Test type</b>	Conformance
<b>Test</b>	<b><i>/conf/getCorridor/request-valid-identifier</i></b>	
	<b>Requirement 3</b>	<b><i>/req/getCorridor/request-valid-identifier</i></b>
	<b>Test purpose</b>	The coverageId parameter value in a <i>GetCorridor</i> request <b>shall</b> be equal to the identifier of one of the coverages offered by the server addressed.
	<b>Test method</b>	Send a <i>GetCorridor</i> request to the service under test. For every listed CoverageId (either in the <i>GetCapabilities</i> response or <i>GetCorridor</i> response ) then send, for each coverage identifier listed, a valid <i>GetCorridor</i> request. Check that none of these requests results in an exception.
	<b>Test type</b>	Conformance
<b>Test</b>	<b><i>/conf/getCorridor/acceptable-format</i></b>	
	<b>Requirement 4</b>	<b><i>/req/getCorridor/acceptable-format</i></b>
	<b>Test purpose</b>	If a <i>GetCorridor</i> request contains a format parameter then this parameter <b>shall</b> contain a MIME type identifier occurring in some WCS::formatSupported element of the response to a successful <i>GetCapabilities</i> request to this server
	<b>Test method</b>	Send <i>GetCapabilities</i> request to server under test, remember Capabilities document returned. Send <i>GetCorridor</i> requests containing valid coverage identifiers to server under test. Vary the format parameter: <ul style="list-style-type: none"> <li>• Send request with a format parameter value containing one of the MIME type identifiers reported in the Capabilities document. Request must be valid in the sense that the format requested can encode the coverage addressed. Verify that request succeeds.</li> <li>• Send request with a format parameter value not occurring in the in the Capabilities document. Verify that request fails.</li> </ul>



## CONFORMANCE CLASS

		Pass test if all checks succeed.
	Test type	Conformance
Test	/conf/getCorridor/acceptable-mediaType	
	Requirement 5	/req/getCorridor/acceptable-mediaType
	Test purpose	If a <i>GetCorridor</i> request contains a <i>mediaType</i> parameter then this parameter <b>shall</b> contain a MIME type identifier of fixed value "multipart/related".
	Test method	Send a <i>GetCorridor</i> request containing a <i>mediaType</i> parameter. Vary this parameter value: <ul style="list-style-type: none"> <li>Send request with a <i>mediaType</i> parameter value as required. Verify that request succeeds.</li> <li>Send request with an illegal <i>mediaType</i> parameter. Verify that request fails.</li> </ul>
		Pass test if all checks succeed.
	Test type	Conformance
Test	/conf/getCorridor/path	
	Requirement 6	/req/getCorridor/path
	Test purpose	The <i>GetCorridor</i> request <b>shall</b> contain a valid <i>PathDescription</i> element within the path element i.e. it conforms to the conformance class:- <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/PathDescription">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/PathDescription</a>
	Test method	Send a <i>GetCorridor</i> request containing and invalid <i>PathDescription</i> and Verify that request fails.
		Pass test if all checks succeed.
	Test type	Conformance
Test	/conf/getCorridor/range-component	
	Requirement 7	/req/getCorridor/range-component
	Test purpose	The parameter value of the <i>RangeComponent</i> of the <i>wcs:Rangeltem</i> element <b>shall</b> contain a parameter that is part of the requested coverage.
	Test method	Send <i>Describe Coverage</i> request to server under test, for a valid coverage and for each coverage note the <i>RangeType</i> items returned in the response document. Send <i>GetCorridor</i> requests containing the <i>Rangeltems</i> and check for a valid response.
		Pass test if all checks succeed.

CONFORMANCE CLASS	
Test type	Conformance
Test	<b>/conf/getCorridor/response-encoding</b>
Requirement 8	<b>/req/getCorridor/response-encoding</b>
Test purpose	The contents of the response to a successful <i>GetCorridor</i> 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.
Test method	For each coverage encoding format (i.e., format encoding extension) supported by the server under test: Send a valid <i>GetCorridor</i> request to retrieve a coverage in this format. Check that the result is a valid instance of the format indicated. Pass test if all checks succeed.
Test type	Conformance

## A.2. CONFORMANCE CLASS: PATHDESCRIPTION

CONFORMANCE CLASS	
<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/PathDescription">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/PathDescription</a>	
Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/PathDescription">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/PathDescription</a>
Dependency	<a href="http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage">http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf">http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/grid-regular/conf">http://www.opengis.net/spec/CIS/1.1/conf/grid-regular/conf</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf">http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf</a>
Test	<b>/conf/PathDescription/structure</b>
Requirement 9	<b>/req/PathDescription/structure</b>
Test purpose	A metoceanCorridor:PathDescription instance <b>shall</b> conform to Figure 3, Figure 25 and Table 5, Table 6, Table 7, Table 8, Table 9, and Table 10

CONFORMANCE CLASS	
Test method	Send <i>GetCorridor</i> requests with valid and PathDescription structure. Pass test if appropriate valid results or exceptions, resp., are delivered.
Test Type	Conformance
Test	<b>/conf/PathDescription/segment-Definition</b>
Requirement 10	<b>/req/PathDescription/segment-Definition</b>
Test purpose	If the SegmentDefinition element is present then either of the two elements segmentPerSector or segmenPerPath <b>shall</b> be present.
Test method	Send <i>GetCorridor</i> requests with a SegmentDefinition element. Pass test if appropriate valid results or exceptions, resp., are delivered.
Test type	Conformance
Test	<b>/conf/PathDescription/getCapabilities-response-segment-definition</b>
Requirement 11	<b>/req/PathDescription/getCapabilities-response-segment-definition</b>
Test purpose	A GetCorridor service implementing the segment extension shall include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/segment-definition">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/segment-definition</a>
Test method	Send a valid <i>GetCorridor</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
Test type	Conformance
Test	<b>/conf/PathDescription/segmentPerPath</b>
Requirement 12	<b>/req/PathDescription/segmentPerPath</b>
Test purpose	The number of the value in the element segmentPerPath <b>shall</b> have a value of that is at least one more than the number of “way points”.
Test method	Send a <i>GetCorridor</i> request with a segmentPerPath value of less than the number of “way points” and check for the error.
Test type	Conformance

## CONFORMANCE CLASS

Test	/conf/PathDescription/segmentPerSector
Requirement 13	/req/PathDescription/segmentPerSector
Test purpose	The number of the value in the element segmentPerSector <b>shall</b> have a value of two or more.
Test method	Send a GetCorridor request with a segmentPerSector value of less than two and check for the error.
Test type	Conformance

## A.3. CONFORMANCE CLASS: CORRIDOREXTENT

### CONFORMANCE CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/CorridorExtent](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtent)

Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtent">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtent</a>
Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Dependency	<a href="http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage">http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf">http://www.opengis.net/spec/CIS/1.1/conf/coverage/conf</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/grid-regular/conf">http://www.opengis.net/spec/CIS/1.1/conf/grid-regular/conf</a>
Dependency	<a href="http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf">http://www.opengis.net/spec/CIS/1.1/conf/grid-irregular/conf</a>
Test	/conf/CorridorExtent/structure
Requirement 14	/req/CorridorExtent/structure
Test purpose	A metoceanCorridor:CorridorExtent instance <b>shall</b> conform to Figure 26, Table 11, Table 12, Table 13 and Table 14
Test method	Send GetCorridor requests with valid and invalid CorridorExtent structure. Pass test if appropriate valid results or exceptions, resp., are delivered
Test Type	Conformance
Test	/conf/CorridorExtent/axis-names

## CONFORMANCE CLASS

	Requirement 15	<b>/req/CorridorExtent/axis-names</b>
	<b>Test purpose</b>	The scal:axis names <b>shall</b> be one of “Corridor_Width”, “Corridor_Height”
	<b>Test method</b>	Send GetCorridor requests containing CorridorExtent element with an invalid axis name content. Pass test if appropriate valid results or exceptions, resp., are delivered
	<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtent/axis-names-duplicates</b>	
	Requirement 16	<b>req/CorridorExtent/axis-names-duplicates</b>
	<b>Test purpose</b>	The CorridorExtent element <b>shall</b> not have duplicate scal:axis names.
	<b>Test method</b>	Send GetCorridor requests containing CorridorExtent element with duplicate scal:axis names. Pass test if appropriate valid results or exceptions, resp., are delivered
	<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtent/TargetAxisSize</b>	
	Requirement 17	<b>/req/CorridorExtent/TargetAxisSize</b>
	<b>Test purpose</b>	The element TargetAxisSize <b>shall</b> contain valid elements scal:axis and scal:targetSize
	<b>Test method</b>	Send GetCorridor requests containing an TargetAxisSize elements with invalid components. Pass test if appropriate exceptions, resp., are delivered.
	<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtent/AxisExtent</b>	
	Requirement 18	<b>/req/CorridorExtent/AxisExtent</b>
	<b>Test purpose</b>	The scal:Axis element, if present, <b>shall</b> have a corresponding every cis:AxisExtent axisLabel with the an attribute that has the same value as the scal:Axis element.
	<b>Test method</b>	Send GetCorridor requests containing an invalid cis:AxisExtent axisLabel element. Pass test if appropriate exceptions, resp., are delivered.
	<b>Test type</b>	Conformance

## A.4. CONFORMANCE CLASS: CORRIDOREXTRACTIONMETHOD

### CONFORMANCE CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/CorridorExtractionMethod](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod)

Depender [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/CorridorExtractionMethod](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/CorridorExtractionMethod)

Dependent <http://www.opengis.net/spec/WCS/2.1/conf/core/getCoverage>

Depender [http://www.opengis.net/spec/WCS\\_service-extension\\_interpolation/1.0/conf/interpolation](http://www.opengis.net/spec/WCS_service-extension_interpolation/1.0/conf/interpolation)

Dependent [http://www.opengis.net/spec/WCS\\_service-extension\\_interpolation/1.0/req/interpolation-per-axis](http://www.opengis.net/spec/WCS_service-extension_interpolation/1.0/req/interpolation-per-axis)

Test /conf/CorridorExtractionMethod/structure

Requirement 19 /req/CorridorExtractionMethod/structure

**Test purpose** A metoceanCorridor:CorridorExtractionMethod instance **shall** conform to Figure 26, Table 15, Table 16, Table 17, Table 18 and Table 19

**Test method** Send GetCorridor requests with valid and invalid request structure. Pass test if appropriate valid results or exceptions, resp., are delivered.

**Test Type** Conformance

Test /conf/CorridorExtractionMethod/Value

Requirement 20 /req/CorridorExtractionMethod/Value

**Test purpose** The CorridorExtractionMethod instance **shall** contain a valid metoceanCorridor:CorridorExtractionMethodValue element that conforms with Table 6.

**Test method** Send GetCorridor requests with valid and invalid CorridorExtractionMethodValue. Pass test if appropriate valid results or exceptions, resp., are delivered.

**Test type** Conformance

Test /conf/CorridorExtractionMethod/interpolationMethod

Requirement 21 /req/CorridorExtractionMethod/interpolationMethod

**Test purpose** In those methods that require interpolation i.e. where the CorridorExtractionValue is either Trajectory\_Point\_ Interpolation or

## CONFORMANCE CLASS

	Corridor_Point_Interpolation then the request <b>shall</b> contain valid int:Interpolation parameter.
<b>Test method</b>	Send GetCorridor request with where the CorridorExtractionValue is either Trajectory_Point_Interpolation or Corridor_Point_Interpolation and if the int:Interpolation element is missing, then throw an exception.
<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/interpolation-axes</b>
Requirement 22	<b>/req/CorridorExtractionMethod/interpolation-axes</b>
<b>Test purpose</b>	<b>/req/CorridorExtractionMethod/interpolation-axes</b> In those methods that require interpolation i.e. where the CorridorExtractionValue is either Trajectory_Point_Interpolation or Corridor_Point_Interpolation then the Interpolation:interpolationAxes element in a GetCorridor request, <b>shall</b> consist of a non-empty sequence of int:InterpolationPerAxis elements
<b>Test method</b>	Send GetCorridor request with where the CorridorExtractionValue is either Trajectory_Point_Interpolation or Corridor_Point_Interpolation and if the int:InterpolationPerAxis elements is empty, then throw an exception.
<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/TrajectoryVoxelInterception</b>
Requirement 23	<b>/req/CorridorExtractionMethod/TrajectoryVoxelInterception</b>
<b>Test purpose</b>	If CorridorExtractionValue is set to "TrajectoryVoxelInterception" then the GetCoverage request containing <b>shall</b> be obtained by applying the method outlined in 7.6.1 during the preparation of the response.
<b>Test method</b>	Send GetCorridor request with where the CorridorExtractionValue is set to TrajectoryVoxelInterception then check the response to ensure the method is applied correctly
<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/TrajectoryVoxel_Interception</b>
Requirement 24	<b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/TrajectoryVoxel_Interception</b>

## CONFORMANCE CLASS

<b>Test purpose</b>	A WCS service implementing requirements class /req/CorridorExtractionMethod/TrajectoryVoxel_Interception of this Extension shall include the following URI in the Profile element of the Servicelidentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/TrajectoryVoxel_Interception">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/TrajectoryVoxel_Interception</a>
<b>Test method</b>	Send a valid <i>GetCapabiliies</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Voxel_Interception</b>
Requirement 25	<b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Voxel_Interception</b>
<b>Test purpose</b>	A WCS service implementing requirements class/req/CorridorExtractionMethod/Corridor_Voxel_Interception of this Extension shall include the following URI in the Profile element of the Servicelidentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Voxel_Interception">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Voxel_Interception</a>
<b>Test method</b>	Send a valid <i>GetCapability r</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
<b>Test type</b>	Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Point_Collection</b>
Requirement 26	<b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Point_Collection</b>
<b>Test purpose</b>	A WCS service implementing requirements class/req/CorridorExtractionMethod/Corridor_Voxel_Interception of this Extension shall include the following URI in the Profile element of the Servicelidentification in a GetCapabilities response:



## CONFORMANCE CLASS

		<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Collection">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Collection</a>
<b>Test method</b>		Send a valid <i>GetCapability</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
<b>Test type</b>		Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Trajectory_Point_Interpolation</b>	
	Requirement 27	<b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Trajectory_Point_Interpolation</b>
<b>Test purpose</b>		A WCS service implementing requirements class/req/CorridorExtractionMethod/Trajectory_Point_Interpolation of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Trajectory_Point_Interpolation">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Trajectory_Point_Interpolation</a>
<b>Test method</b>		Send a valid <i>GetCapabilities</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
<b>Test type</b>		Conformance
<b>Test</b>	<b>/conf/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Point_Interpolation</b>	
	Requirement 28	<b>/req/CorridorExtractionMethod/getCapabilities-response-conformance-class-in-profile/Corridor_Point_Interpolation</b>
<b>Test purpose</b>		A WCS service implementing requirements class/req/CorridorExtractionMethod/Corridor_Point_Interpolation of this Extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Interpolation">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/CorridorExtractionMethod/Corridor_Point_Interpolation</a>

## CONFORMANCE CLASS

Test method	Send a valid <i>GetCapability</i> request to the service under test; for each ows:Profile element listed in the response, check that the corresponding conformance class exists and, if so, perform its conformance tests in completeness. Test passes if all conformance classes listed exist and each check succeeds completely.
Test type	Conformance

## A.5. CONFORMANCE CLASS: GETCORRIDOR-POST-XML

### CONFORMANCE CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/getCorridor-post-xml](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-post-xml)

Dependency [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor-post-xml](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-post-xml)

Dependency [http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/req/getCorridor](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor)

Dependency WCS 2.1 protocol extension XML-POST [OGC 09-148r1]

Test </conf/getCorridor-post-xml/mandatory>

Requirement 29 </req/getCorridor-post-xml/mandatory>

**Test purpose** Implementations of this GetCorridor extension that support the GetCorridor post-xml requirements class shall support the WCS 2.1 protocol extension XML-POST [OGC 09-148r1].

**Test method** Determine the list of supported extensions via a valid GetCapabilities request; check that the extension required is listed.

**Test type** Conformance

Test </conf/getCorridor-post-xml/conformance-class-in-profile>

Requirement 30 </req/getCorridor-post-xml/conformance-class-in-profile>

**Test purpose** Implementations of this GetCorridor extension that support the GetCorridor-post-xml requirements class shall include the following URI in a Profile element in the

CONFORMANCE CLASS	
	ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-post-xml">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-post-xml</a>
Test method	Determine the list of supported extensions via a valid GetCapabilities request; check that the extension required is listed.
Test type	Conformance
Test	<code>/conf/getCorridor-post-xml/getCorridor-request-structure</code>
Requirement 31	<code>/req/getCorridor-post-xml/getCorridor-request-structure</code>
Test purpose	A WCS server implementing the XML/POST protocol binding extension <b>shall</b> encode request body of the GetCorridor operation using an XML document of type metocean:GetCorridor and described in this document.
Test method	Send syntactically legal and illegal GetCoverage request to server under test, verify that the server responds appropriately.
Test type	Conformance

## A.6. CONFORMANCE CLASS: GETCORRIDOR-SIMPLE

CONFORMANCE CLASS	
	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple</a>
Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple</a>
Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Test	<code>/req/getCorridor-simple/corridor-extent</code>
Requirement 32	<code>/req/getCorridor-simple/corridor-extent</code>
Test purpose	The GetCorridor request shall not contain a metocean:corridorExtent element.

## CONFORMANCE CLASS

	Test method	If the GetCorridor only advertises through the Profile element contained in the GetCapabilities response that is supports this extension then fail if it accepts this element.
	Test type	Conformance
Test	<a href="/conf/getCorridor-simple/extraction-method">/conf/getCorridor-simple/extraction-method</a>	
	Requirement 33	<a href="/req/getCorridor-simple/extraction-method">/req/getCorridor-simple/extraction-method</a>
	Test purpose	The metoceancorridor:corridorExtractionMethodValue <b>shall</b> only contain the values:-Trajectory_Point_Interpolation or Trajectory_Voxel_Interpolation
	Test method	If the GetCorridor only advertises through the Profile element contained in the GetCapabilities response that is supports this extension then fail if it accepts any other values than those listed in this requirement.
	Test type	Conformance
Test	<a href="/conf/getCorridor-simple/conformance-class-in-profile">/conf/getCorridor-simple/conformance-class-in-profile</a>	
	Requirement 34	<a href="/req/getCorridor-simple/conformance-class-in-profile">/req/getCorridor-simple/conformance-class-in-profile</a>
	Test purpose	A GetCorridor service implementing only this conformance class <b>getCorridor-simple</b> of this extension <b>shall</b> include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple</a>
	Test method	Determine the list of supported extensions via a valid GetCapabilities request; check that the extension required is listed.
	Test type	Conformance

## A.7. CONFORMANCE CLASS: GETCORRIDOR-SIMPLE-GET-KVP

---

### CONFORMANCE CLASS

[http://www.opengis.net/spec/WCS\\_application-profile\\_metocean\\_corridor/1.0/conf/getCorridor-simple-get-kvp](http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple-get-kvp)

## CONFORMANCE CLASS

Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple-get-kvp">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor-simple-get-kvp</a>
Dependency	<a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/req/getCorridor</a>
Test	<b>/conf/getCorridor-get-kvp/conformance-class-in-profile</b>
Requirement 35	<b>/req/getCorridor-simple-get-kvp/conformance-class-in-profile</b>
<b>Test purpose</b>	Implementations of this getCorridor-simple-get-kvp extension that supports the getCorridor-simple-get-kvp requirements class shall include the following URI in a Profile element in the ServiceIdentification in a GetCapabilities response: <a href="http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple-get-kvp">http://www.opengis.net/spec/WCS_application-profile_metocean_corridor/1.0/conf/getCorridor-simple-get-kvp</a>
<b>Test method</b>	Determine the list of supported extensions via a valid GetCapabilities request; check that the extension required is listed.
<b>Test type</b>	Conformance
Test	<b>/conf/getCorridor-simple-get-kvp/request-structure</b>
Requirement 36	<b>/req/getCorridor-simple-get-kvp/request-structure</b>
<b>Test purpose</b>	A WCS server implementing both this get-kvp protocol binding extension <b>shall</b> encode the DescribeCoverageCollection operation request as specified in Table 20 .
<b>Test method</b>	Send a valid <i>GetCorridor</i> request to server under test which conforms to the references in the requirement. Check that the response is not an exception
<b>Test type</b>	Conformance



# ANNEX B (INFORMATIVE) REVISION HISTORY

---



# ANNEX B (INFORMATIVE) REVISION HISTORY

---

DATE	RELEASE	EDITOR	PRIMARY CLAUSES MODIFIED	DESCRIPTION
2017-08-21	0.1	Trevelyan/Hershberg/Olson	all	Created
2018-11-21	0.2	Trevelyan/Hershberg/Olson	all	



# BIBLIOGRAPHY







# BIBLIOGRAPHY

---