**Open Geospatial Consortium**

Date: 2016-02-15

External identifier of this OGC® document:

Internal reference number of this OGC® document: 15-108r1

Version: 0.1

Category: OGC® Draft Implementation Specification

Editor: Peter Trevelyan

**­**

**MetOcean GetCorridor Extension for WCS2.0**

**Copyright notice**

Copyright © 2015 Open Geospatial Consortium  
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

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

Document type: OGC® Draft Implementation Specification

Document subtype:

Document stage: Draft

Document language: English

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.

Contents

[**Open Geospatial Consortium** 1](#_Toc427587341)

[**MetOcean GetCorridor Extension for WCS2.0** 1](#_Toc427587342)

[**Copyright notice** 1](#_Toc427587343)

[**Warning** 1](#_Toc427587344)

[License Agreement 2](#_Toc427587345)

[Contents 3](#_Toc427587346)

[Tables 6](#_Toc427587347)

[1. Scope 7](#_Toc427587348)

[2. Conformance 7](#_Toc427587349)

[3. References 9](#_Toc427587350)

[OGC MetOcean metadata profile for WCS2.0 Document 15-45 9](#_Toc427587351)

[4. Terms and Definitions 11](#_Toc427587352)

[4.1 numerical weather prediction model 11](#_Toc427587353)

[4.2 reference time 11](#_Toc427587354)

[4.3 verification time 11](#_Toc427587355)

[4.4 GRIB 11](#_Toc427587356)

[4.5 Web Coverage Service 2.0 (WCS2.0) 11](#_Toc427587357)

[4.6 GetCapabilities operation 12](#_Toc427587358)

[4.7 GetMetOCeanCorridor operation 12](#_Toc427587359)

[4.8 path 12](#_Toc427587360)

[4.9 corridor 12](#_Toc427587361)

[5. Conventions 13](#_Toc427587362)

[5.1 Abbreviated terms 13](#_Toc427587363)

[5.2 Schema language 13](#_Toc427587364)

[5.3 UML notation 13](#_Toc427587365)

[5.4 Namespace prefix conventions 14](#_Toc427587366)

[6. Non-Normative (Informative) Material 14](#_Toc427587367)

[6.1 WCS2.0 14](#_Toc427587368)

[6.1.1 A Short NWP (Numerical Weather Prediction) Primer 14](#_Toc427587369)

[6.1.2 Coverages 15](#_Toc427587370)

[Figure 1 UML Diagram representing the coverage model. 15](#_Toc427587371)

[6.1.3 Trajectory coverage 16](#_Toc427587372)

[Figure 2 UML Diagram representing the trajectory observation model. 16](#_Toc427587373)

[6.1.4 4D Coverages 16](#_Toc427587374)

[6.2 Trajectories and corridors: 17](#_Toc427587375)

[6.2.1 Surface Route Analysis: 18](#_Toc427587376)

[6.2.2 Forecast of weather conditions along an aircraft flight path, in time and space:- 25](#_Toc427587377)

[6.2.3 The effect of the vertical interpolation to the trajectory: 30](#_Toc427587378)

[7. UML Model (Normative) 32](#_Toc427587379)

[7.1 Requirements Class: GetMetOceanCorridor 32](#_Toc427587380)

[7.1.1 Requirements class overview 33](#_Toc427587381)

[7.1.2 GetMetOceanCorridor 34](#_Toc427587382)

[7.1.3 PathDescription. 34](#_Toc427587383)

[7.1.4 ReferenceableGridByArray. 35](#_Toc427587384)

[7.1.5 CorridorDescription. 35](#_Toc427587385)

[7.1.6 AxisDetails 36](#_Toc427587386)

[7.1.7 VerticalOrientationMode. 36](#_Toc427587387)

[7.1.8 PointSelectionMode. 37](#_Toc427587388)

[7.1.9 AxisName. 37](#_Toc427587389)

[7.1.10 InterpolationMethod. 37](#_Toc427587390)

[8. Amendments to WCS service model to support to support the GetMetOCeanCorridor: 38](#_Toc427587391)

[8.1 Requirements Class: Get capabilities 40](#_Toc427587392)

[8.1.1 Requirements class overview 41](#_Toc427587393)

Figures

[Figure 1 UML Diagram representing the coverage model. 15](#_Toc427587305)

[Figure 2 UML Diagram representing the trajectory observation model. 16](#_Toc427587306)

[Figure 3 A simple example of a Trajectory observation 17](#_Toc427587307)

[Figure 4 — Extraction of the trajectory using grid-box semantics. 18](#_Toc427587308)

[Figure 5 — Extraction of the corridor using grid-box semantics. 19](#_Toc427587309)

[Figure 6— Extracting of a corridor, using point based semantics. 20](#_Toc427587310)

[Figure 7— Extraction of the trajectory using grid-point semantics, horizontal interpolation, but no path sampling. 21](#_Toc427587311)

[Figure 8 — Extraction of the trajectory using grid-point semantics, horizontal interpolation, and path sampling. 22](#_Toc427587312)

[Figure 9 — Extraction of the corridor using grid-point semantics, horizontal interpolation, no path sampling, and a corridor extent 23](#_Toc427587313)

[Figure 10 — Extraction of the corridor using grid-point semantics, horizontal interpolation, path sampling, and a corridor extent 24](#_Toc427587314)

[Figure 11 Extraction of the trajectory using voxel-in-centre semantics.. 25](#_Toc427587315)

[Figure 12 Extraction of a corridor using voxel-in-centre semantics. 26](#_Toc427587316)

[Figure 13 Extraction of a corridor, using point based semantics. 27](#_Toc427587317)

[Figure 14 Extraction of a corridor, using point based semantics. Vertical and horizontal interpolation. 28](#_Toc427587318)

[Figure 15 Vertical and horizontal interpolation. 29](#_Toc427587319)

[Figure -- 16 vertical orientation in the vertical dimension 30](#_Toc427587320)

[Figure 17 -- vertical orientation perpendicular to the corridor 31](#_Toc427587321)

[Figure 18 The UML model for GetMetOceanCorridor 33](#_Toc427587322)

Tables

[Table 1 UML Namespace prefix conventions 15](#_Toc427587134)

[Table 2 GetMetOceanCorridor properties 35](#_Toc427587135)

[Table 3 PathDescription properties 35](#_Toc427587136)

[Table 4 ReferenceableGridByArray Properties 36](#_Toc427587137)

[Table 5 Corridor Description Properties 36](#_Toc427587138)

[Table 6 Axis Details Properties 37](#_Toc427587139)

[Table 7 – Vertical Orientation Mode code items 37](#_Toc427587140)

[Table 8 – PointSelectionMode code items 38](#_Toc427587141)

[Table 9 – AxisName code items 38](#_Toc427587142)

[Table 10 – InterpolationMethod code items 39](#_Toc427587143)

1. Abstract

This document defines a MetOcean profile consisting of an information model and an XML encoding for the following WCS2.0 operations:

1. *-* a WCS server extracts from a multidimensional coverage a corridor (i.e. a corridor based on a trajectory with an optional boundary).

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

1. Keywords

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

WCS, coverage, meteorology, oceanography, NWP, trajectory, corridor, analysis, observation, measurement, O&M, GetMetOCeanCorridor:

1. Preface

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

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

1. Submitting organizations

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

Met Office, UK

1. Submitters

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

|  |  |
| --- | --- |
| Name | Company |
| Peter Trevelyan | Met Office, UK |

1. Scope

The purpose of the GetMetOCeanCorridor operation is to extract a corridor based on a trajectory from a multidimensional coverage.

The usefulness of these extensions outside of the MetOcean community may well result in this extension being incorporated into the WCS2.0 core family of extensions.

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

1. Conformance

This standard defines:

* An additional operation, *GetMetOCeanCorridor*, whose request enables a client application to access a multidimensional corridor from a MetOcean coverage.
* An amended *GetCapabilities* operation whose response provides summary information of offered *GetMetOCeanCorridor* resources.
* XML/POST protocol bindings for the new *GetMetOCeanCorridor* operation.

Requirements are considered for

* XML/POST protocol binding.

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

In order to conform to this OGC™interface standard, a software implementation shall choose to implement any one of the conformance levels specified in Annex B (normative).

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

1. References

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

OGC 08-131r3 – The Specification Model – A Standard for Modular Specification

ISO 19103:2005 – Geographic information - Conceptual schema language

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

OGC Abstract Specification Topic 1 – Feature geometry (aka ISO 19107)

OGC Abstract Specification Topic 2 – Spatial Referencing by Coordinates (aka ISO 19111:2007)

OGC Abstract Specification Topic 6 – Schema for Coverage geometry and functions (aka ISO 19123:2005)

OGC Abstract Specification Topic 11 – Geographic information — Metadata (aka ISO 19115:2014)

OGC Abstract Specification Topic 20 – Observations and Measurements (aka ISO 19156:2011)

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

OGC® Web Coverage Service 2.0 Interface Standard - Core OGC Document 09-110r4 <http://www.opengeospatial.org/standards/wcs>

OGC MetOcean metadata profile for WCS2.0 Document 15-45

<https://portal.opengeospatial.org/files/?artifact_id=63323&version=1>

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

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

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

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

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

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

1. Terms and Definitions

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

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

* 1. numerical weather prediction model

[mathematical model](http://en.wikipedia.org/wiki/Mathematical_model) of the atmosphere and oceans used to [predict the weather](http://en.wikipedia.org/wiki/Weather_forecasting) based on current weather conditions and are normally run at set times each day.

Synonyms: forecast model, NWP Model.

EXAMPLE The ECMWF model that runs twice per day and creates a ten day prediction of the global atmosphere.

* 1. reference time

nominal start time at the beginning of a specific forecast model run.

Synonym: model run time.

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

* 1. verification time

time at which a forecast becomes verifiable.

Synonym: validity time.

NOTE: 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.

* 1. GRIB

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

* 1. [Web Coverage Service](http://www.opengeospatial.org/standards/wcs) 2.0 (WCS2.0)

standard created by the OGC that refers to the exchange of geospatial information as ‘coverages’: digital geospatial information representing space-varying phenomena.

* 1. GetCapabilities operation

request to a WCS server for a list of what operations and services (“capabilities”) are being offered by that server.

* 1. GetMetOCeanCorridor operation

request to a WCS server for a corridor coverage based on a trajectory path with a lateral and vertical extent.

* 1. path

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

* 1. corridor

a corridor is defined as a passageway. The spatial extent is defined in one or more dimensions with reference to the path .

1. Conventions
   1. Abbreviated terms

GML Geography Mark-up Language

O&M Observations and Measurements

OGC Open Geospatial Consortium

MetOcean Meteorological/Oceanographic

NWP Numerical Weather Prediction

UML Unified Modelling Language

WCS2.0 OGC Web Coverage Service version 2.0

WMO World Meteorological Organisation

XML W3C Extensible Markup Language

XSD W3C XML Schema Definition Language

* 1. Schema language

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

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

* 1. UML notation

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

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

 Tan: Defined within this standard

 Yellow: GMLCOV

 Red: ISO19156 – Observations & Measurements

 Green: ISO19115 – The MetOcean Metadata

 Blue WCS 2.0

* 1. Namespace prefix conventions

The following namespaces are used in this document. The prefix abbreviations used constitute conventions used here, but are **not** normative. The namespaces to which the prefixes refer are normative.

Table 1 UML Namespace prefix conventions

|  |  |  |
| --- | --- | --- |
| **Prefix** | **Namespace URI** | **Description** |
| xsd | <http://www.w3.org/2001/XMLSchema> | XML Schema namespace |
| gml | <http://www.opengis.net/gml/3.2> | GML 3.2.1 |
| gmlcov | <http://www.opengis.net/gmlcov/1.0> | GML Application Schema for Coverages 1.0 |
| wcs | <http://www.opengis.net/wcs/2.0> | WCS 2.0 Core |
| covcoll | [http://www.opengis.net/wcs/coveragecollection /1.0](http://www.opengis.net/wcs/coveragecollection%20/1.0) | WCS Coveragecollection Extension |
| metoceancorr | [http://www.opengis.net/wcs/metoceancorr /1.0](http://www.opengis.net/wcs/metoceancorr%20/1.0) | WCS MetOceanGetCorridor Extension |

* 1. Multiple representations

When multiple representations of the same information are given in a specification document these are consistent. Should this not be the case then this is considered an error, and the XML schema shall take precedence.

1. Non-Normative (Informative) Material
   1. WCS2.0

The WCS2.0 core standard and core extensions (see below) cover most of the operations (specifically GetCapabilities, DescribeCoverage and GetCoverage.

WCS Core Extensions

* WCS CoverageCollections, version 1.0.0, OGC 15-044
* WCS MetOcean Metadata, version 1.0.0, OGC 15-045
* WCS Range Subsetting Extension, version 1.0.0, OGC 12-040
* WCS Scaling Extension, version 1.0.0, OGC 12-039
* WCS Range Subsetting version 1.0 OGC 12-040
* WCS Interpolation Extension, version 1.0.0, OGC 12-049
* WCS CRS Extension version 1.0 OGC 11-053
  + 1. A Short NWP (Numerical Weather Prediction) Primer

The term “NWP model” refers to a computer model 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; this “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.

* + 1. Coverages

Coverages represent digital geospatial information representing space/time-varying phenomena. OGC Abstract Topic 6 [OGC 07-011] – which is identical to ISO 19123 – defines an abstract model of coverages. A typical NWP forecast model data may expressed as a set of coverages typically, but not exclusively rectified grid coverages, i.e. coverages whose horizontal domain is a rectified Grid. A typical model run contains literally thousands of 2D coverages each with a unique identifier. The metadata to describe this soon becomes unmanageable and the problem can be simplified by identifying, where possible, “4D Coverages”.

As mentioned in the ISO definition, the concept of “coverage” (see Figure 1) is central to the representation of many common weather observations and forecasts. Weather datasets that fall into the category of coverages include point measurements, wind profiles, model grids, and time series measurements at a single point. Of particular interest to aviation are weather properties observed or forecast along a trajectory, which can also be represented as a “coverage” (see Figure 2)



Figure 1 UML Diagram representing the coverage model.

Figure 2 UML Diagram representing the trajectory observation model.

* 1. Trajectories and corridors:

To help with this discussion it will be instructive to look at some possible methods that can be used to extract a trajectory/corridor. These examples will be heavily based on MetOcean use cases, but can be easily extended by other communities of interest. It is not expected that all these methods will be available on the server, but it is important to at least look at the range of options and try and extract the key variables that will need to be specified by the WCS trajectory API. It will be necessary to advertise different levels of server capability via the GetCapabilities response document.

Encoding of the trajectory in GML may be done for a simple coverage, but this is likely to be very verbose and in many cases the ending will be done in a binary format such as NetCDF. For a simple trajectory/corridor an alternative to GML such as JSON will provide a very accessible encoding that may be used in client software such as JavaScript and HTML5. The encoding of the trajectory in GML will be covered in another document, but the UML diagram (see Figure 3 ) illustrates a typical trajectory observation pattern.

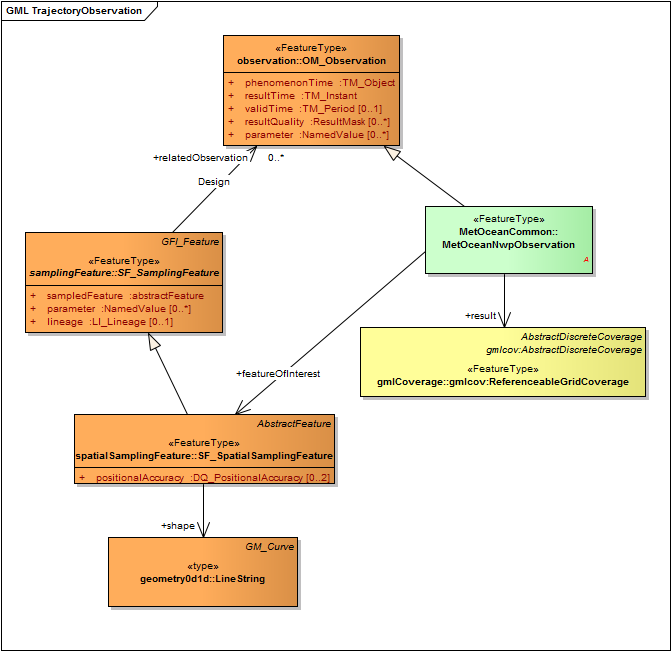


Figure 3 A simple example of a Trajectory observation

* + 1. Surface Route Analysis:

Extracting a weather data along a surface corridor; the route is defined by a series of “way points” with an optional time dimension.

* + - 1. Method 1 (trajectory with grid-box semantics)

With reference to Figure 4; the path is defined by a set of “way points” (in green), and are un-equally spaced. The coverage returned in response to a successful GetMetOceanCorridor request would have as its “domain set” the centre 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 (see Figure 4).

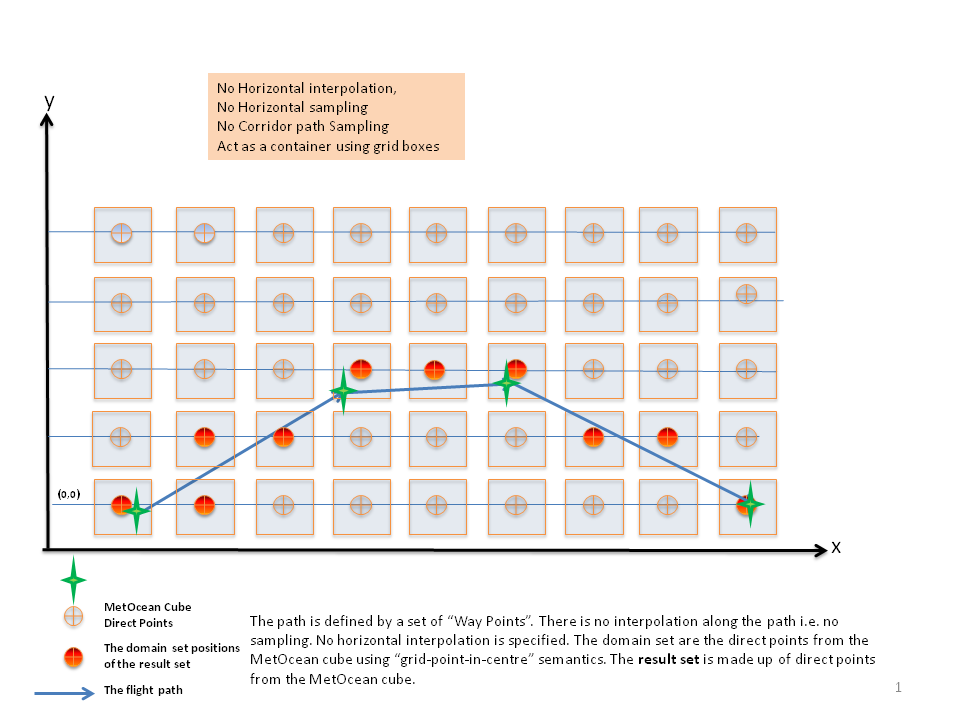


Figure 4 — Extraction of the trajectory using grid-box semantics.

* + - 1. Method 2 (corridor with grid-box semantics)

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 GetMetOceanCorridor 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 5). No horizontal interpolation is performed.

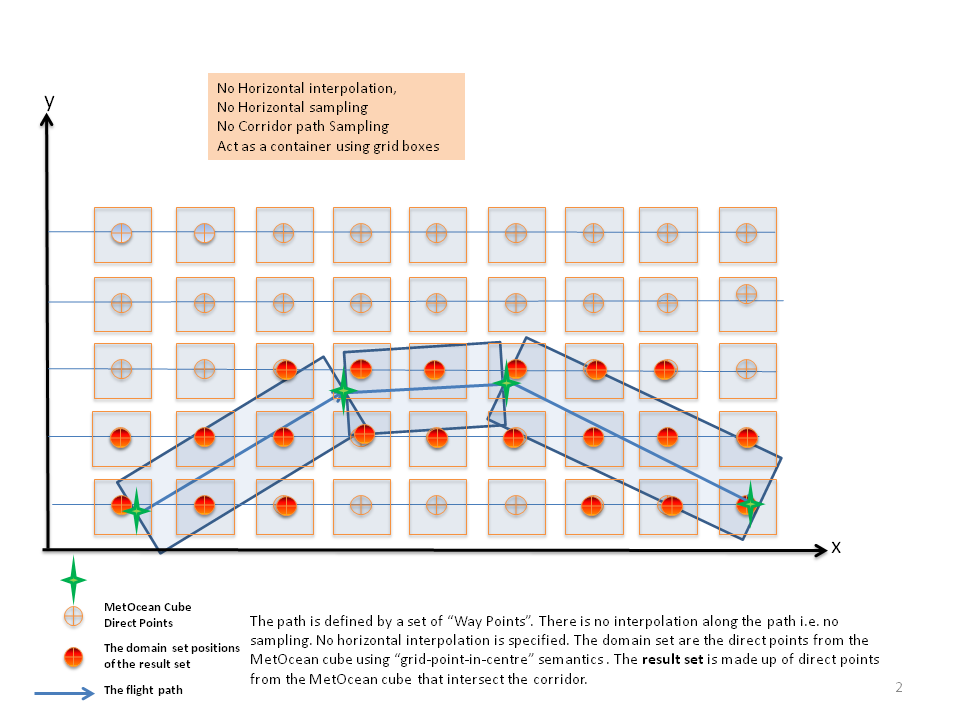


Figure 5 — Extraction of the corridor using grid-box semantics.

* + - 1. Method 3 (corridor with grid point based semantics)

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 GetMetOCeanCorridor 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 6). No horizontal interpolation is performed.

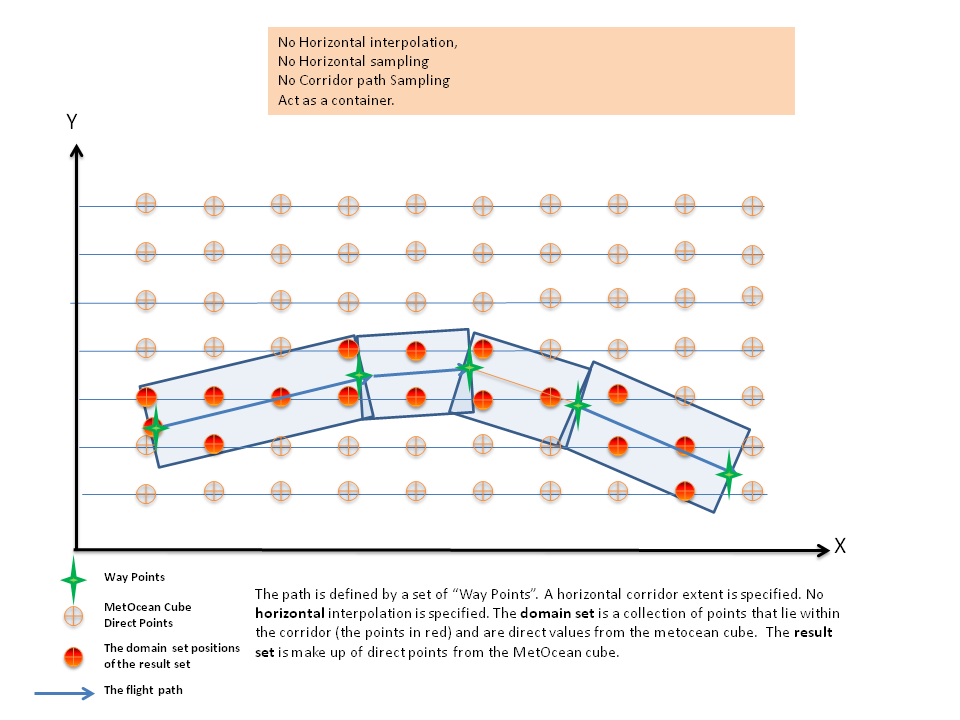


Figure 6— Extracting of a corridor, using point based semantics.

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

* + - 1. Method 4 (trajectory with point based semantics, horizontal interpolation, but no path sampling)

The path is defined by a set of “way points” (in green) that are un-equally spaced and each “way point” may have a different time. The coverage returned in response to a successful GetMetOCeanCorridor 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 7).

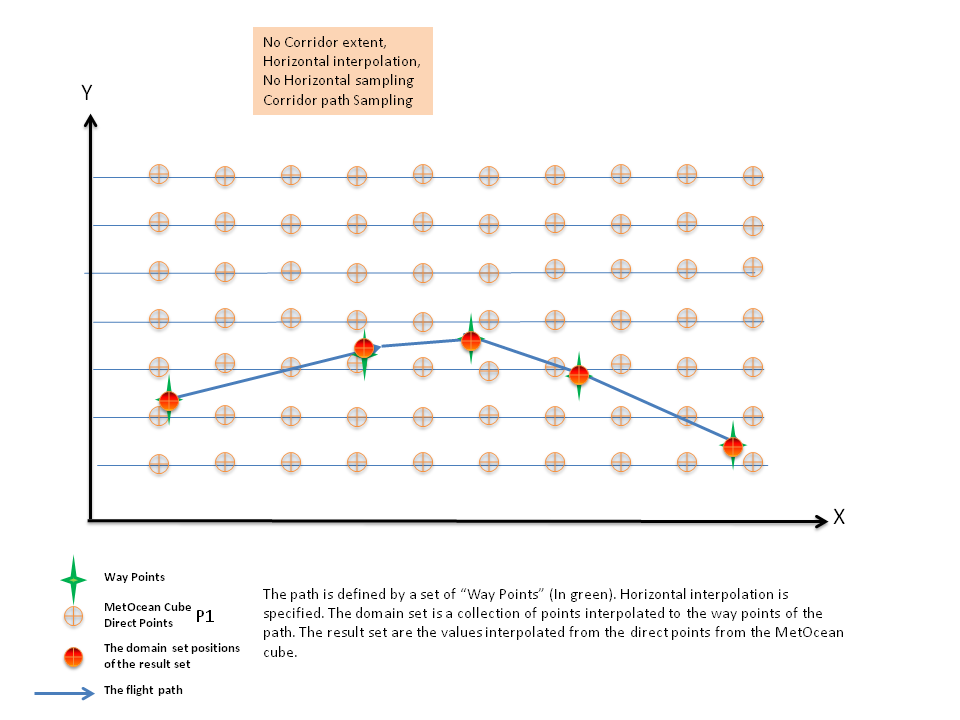
****

Figure 7— Extraction of the trajectory using grid-point semantics, horizontal interpolation, but no path sampling.

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.

* + - 1. Method 5 (trajectory extraction with point based semantics, horizontal interpolation, path sampling)

The path is defined by a set of “way points” (in green) that are un-equally spaced and each “way point” may have a different time. The trajectory is divided into “n” segments (given by “i.e number of path segments”), denoted in the diagram by the blue stars. The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set” each sample point (the points in red). The result set is calculated by interpolating from the MetOcean coverage to each of the sampling points (see Figure 8).

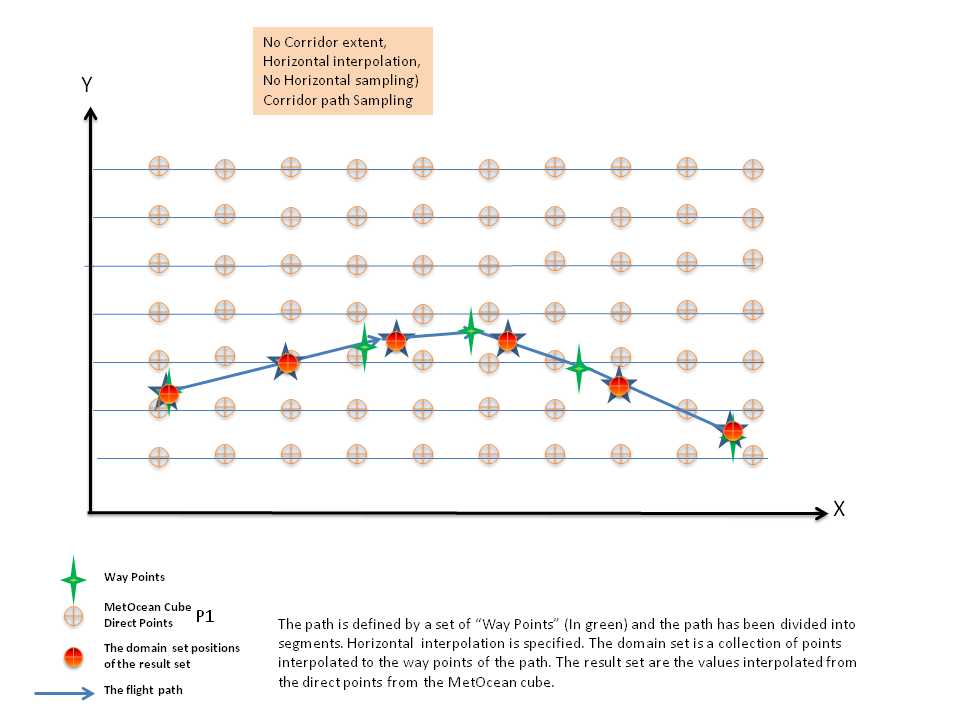


Figure 8 — Extraction of the trajectory using grid-point semantics, horizontal interpolation, and path sampling.

If time interpolation is used then the trajectory is divided up into equal time segments using the “path sampling size” and the start and end times. In this case that may or may not be equal in space as the speed may vary between points. Thus the lat/long of each returned point would be dependent on the average speed for each time segment.

* + - 1. Method 6 (corridor with point based semantics, horizontal interpolation, no path sampling, and a corridor extent)

The path is defined by a set of “way points” that are un-equally spaced (in green), with a horizontal buffer region around the path (defined by a “horizontal buffer Extent”). The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set” each sample point (the points in red). The result set is calculated by interpolating from the MetOcean coverage to each of the sampling points (see Figure 9).

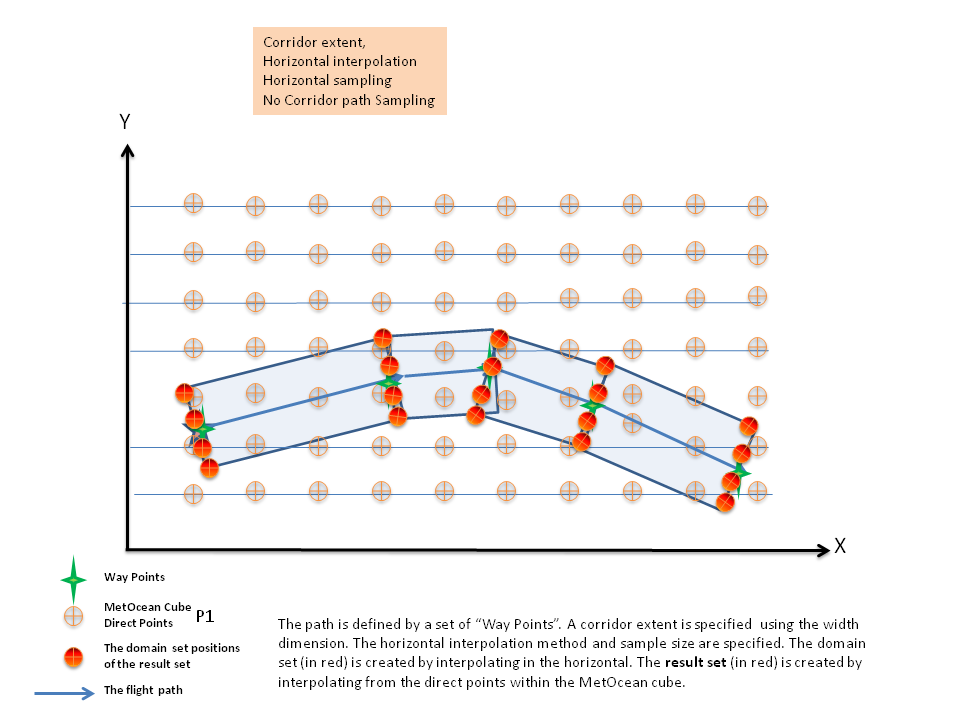


Figure 9 — Extraction of the corridor using grid-point semantics, horizontal interpolation, no path sampling, and a corridor extent

If there a different time for each way point then the data will be interpolated to the correct time for each way point as long as the coverage has a time dimension. As no sampling is done the results set will exactly match the way point values.

* + - 1. Method 7 (corridor with point based semantics, horizontal interpolation, path sampling, and a corridor extent)

The path is defined by a set of “way points” (in green) that are un-equally spaced and each “way point” may have different times. There is a horizontal buffer region around the path (defined by a “horizontal buffer Extent”). The trajectory is divided into “n” segments (given by “number of path segments”), denoted in the diagram by the blue stars. The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set” each sample point (the points in red). The result set is calculated by interpolating from the MetOcean coverage to each of the sampling points (see Figure 10).

.

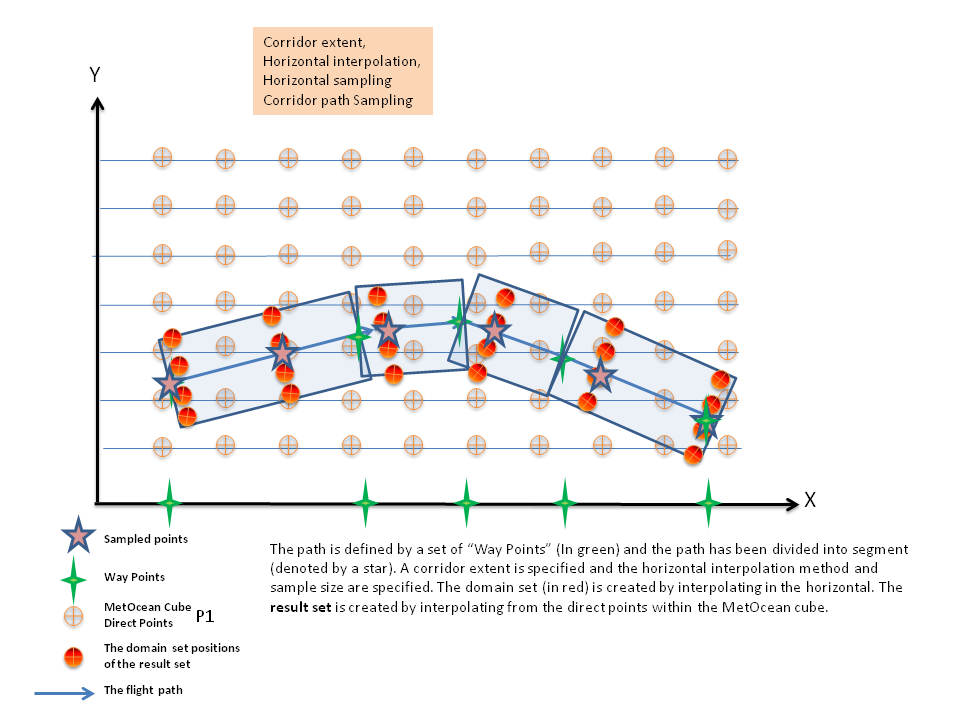


Figure 10 — Extraction of the corridor using grid-point semantics, horizontal interpolation, path sampling, and a corridor extent

If time interpolation is used then the trajectory is divided up into equal time segments using the” path sampling size” and the start and end times. In this case that may or may not be equal in space as the speed may vary between points. In this example each group of four points would have the same time.

* + 1. Forecast of weather conditions along an aircraft flight path, in time and space:-

An aircraft controller wishes to extract forecast winds for a flight corridor. The corridor will be defined by, a series of way points defined in space (aka way points that includes time and elevation).The data needs to be encoded in a number of formats to include JSON and NetCDF. The width and height of the corridor may be specified.

* + - 1. Method 1 (trajectory using voxel-in-centre semantics);

The trajectory is defined by a set of “Way Points” (P1 to P4). The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set” the grid points contained using the “voxel-in-centre” semantics i.e. the voxels intersected by the trajectory. The result set is made up of direct points, i.e. no interpolation, from the MetOcean coverage (in red) extracted from each voxel that the path intersects (see Figure 11).

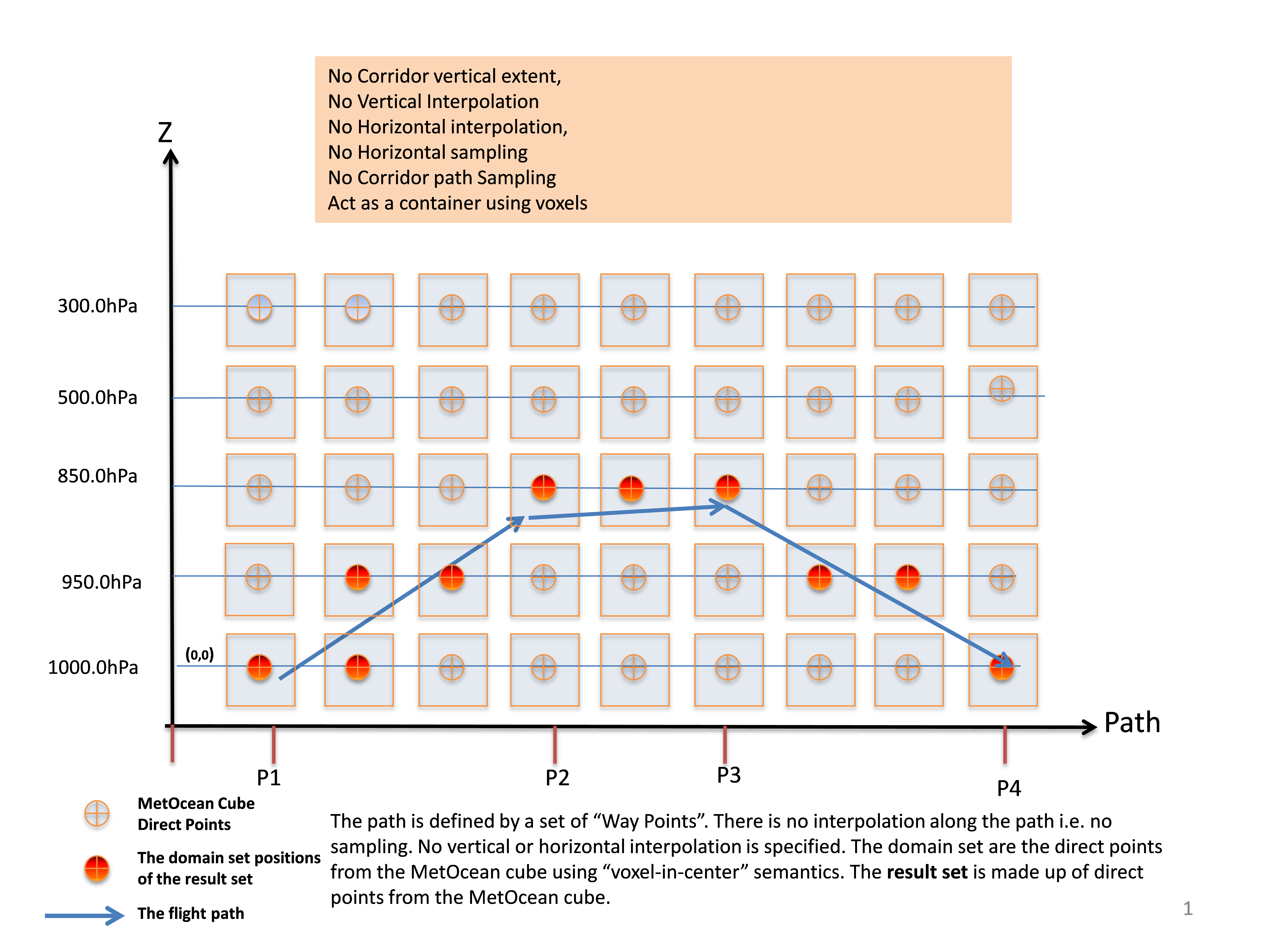


Figure 11 Extraction of the trajectory using voxel-in-centre semantics..

* + - 1. Method 2 (corridor extraction using voxel-in-centre semantics);

The corridor is defined by a set of “Way Points” (P1 to P4) and the vertical extent. The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set the grid points contained using the “voxel-in-centre” semantics i.e. the voxels overlapped by the corridor. The result set is made up of direct points (in red), i.e. no interpolation, from the MetOcean coverage extracted from each voxel that the corridor intersects (see Figure 12).

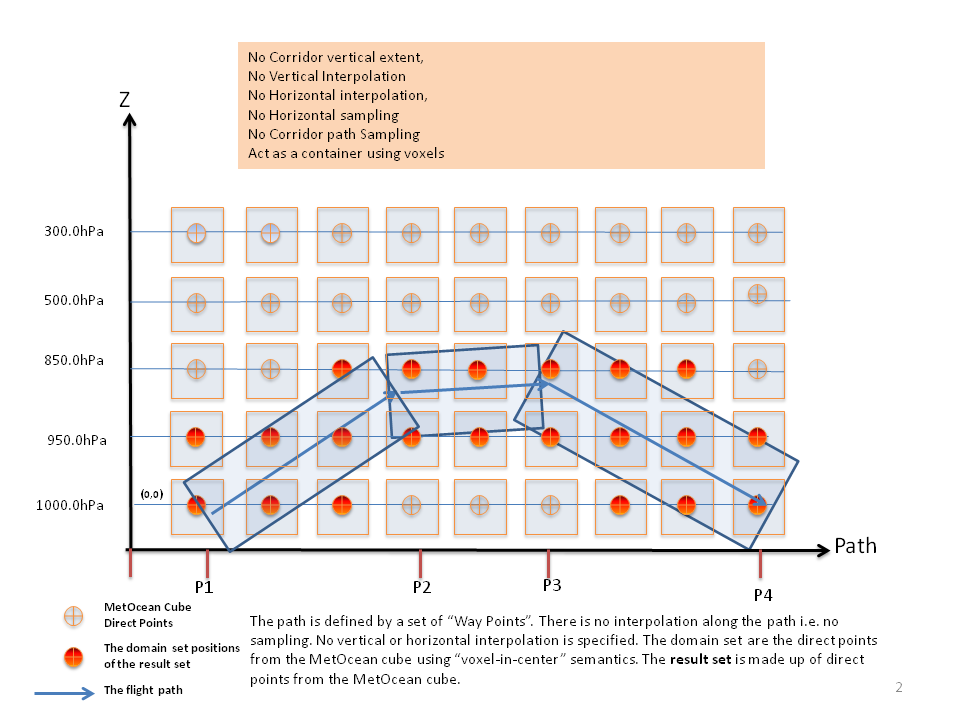


Figure 12 Extraction of a corridor using voxel-in-centre semantics.

**.**

* + - 1. Method 3 (corridor extraction using grid point semantics)

The corridor is defined by a set of “way points”(P1 to P4) and by a vertical buffer region around the path. The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set” all the points that lie within the corridor (the points in red). The result setare the direct point values taken from the MetOcean coverage as there is not horizontal or vertical interpolation (see Figure 13). Note how this differs from the voxel semantics.

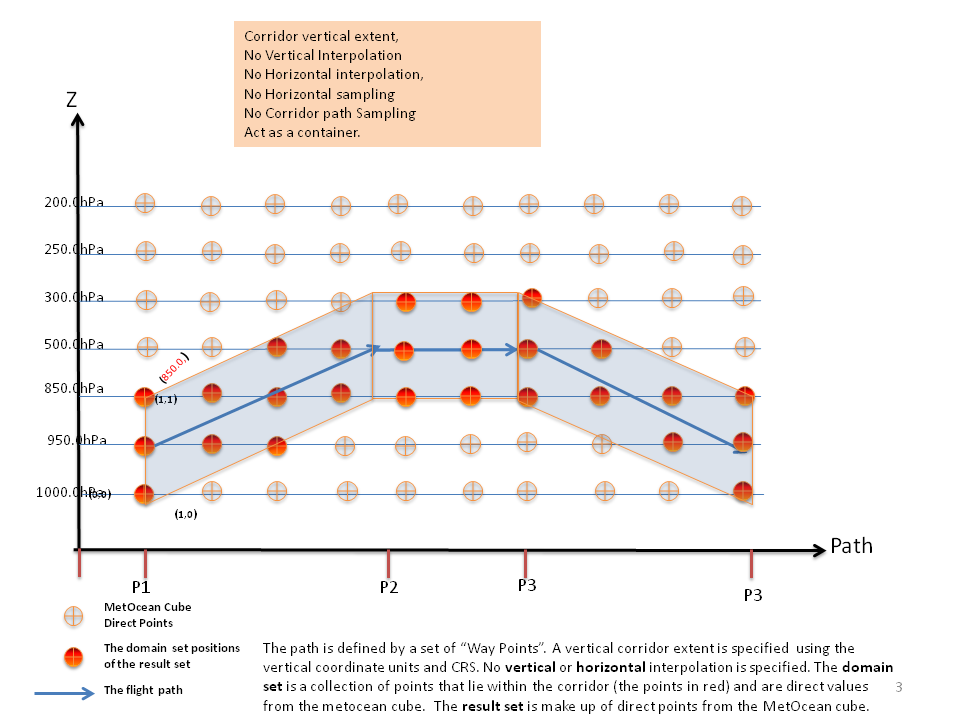


Figure 13 Extraction of a corridor, using point based semantics.

* + - 1. Method 4 (trajectory extraction using grid point semantics, no path sampling, vertical and horizontal interpolation)

The trajectory is defined by a set of “Way Points” (P1 to p4). The coverage returned in response to a successful GetMetOCeanCorridor request would have as its “domain set” a collection of points (in red) i.e. the way points. The result setare the values, vertically and horizontally interpolated from the direct point values within the MetOcean coverage (see Figure 14).

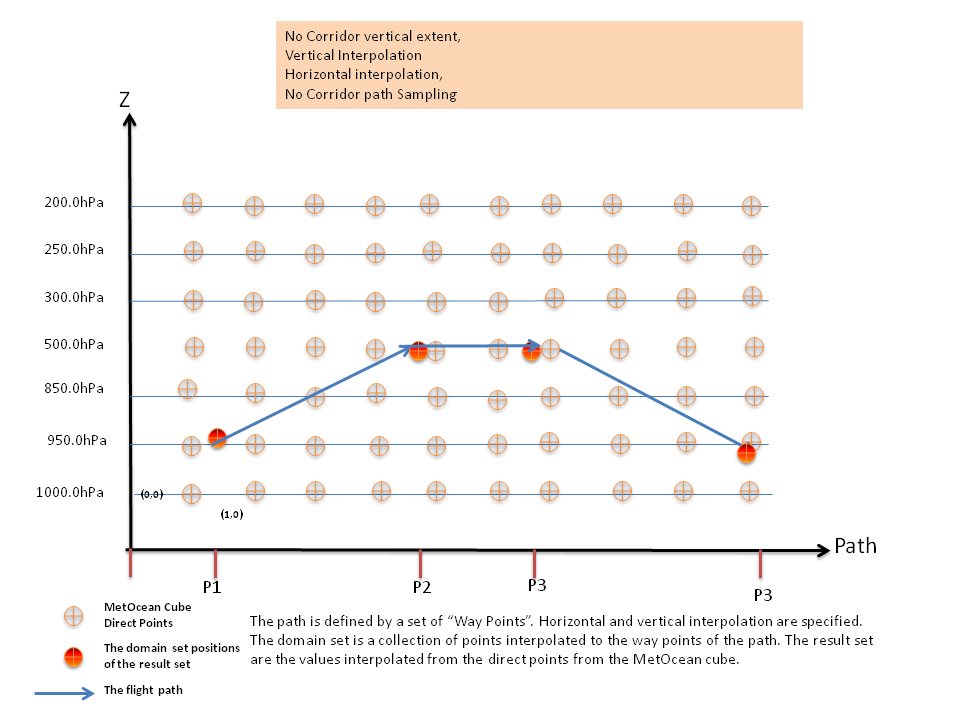


Figure 14 Extraction of a corridor, using point based semantics. Vertical and horizontal interpolation.

If there a different time for each way point then the data will be interpolated to the correct time for each way point as long as the coverage has a time dimension. As no sampling is done the results set will exactly match the way point values.

* + - 1. Method 5 (corridor extraction using grid point semantics, vertical and horizontal interpolation)

The path is defined by a set of “Way Points” (P1 to P4) and a vertical extent. The coverage returned in response to a successful GetMetOCeanCorridor request would have as its domain set a collection of points denoted in red. The result setare the direct point values taken from the MetOcean coverage and interpolated to each point using horizontal and vertical interpolation (see Figure 15).



Figure 15 Vertical and horizontal interpolation.

If there a different time for each way point then the data will be interpolated to the correct time for each way point as long as the coverage has a time dimension. As no sampling is done the results set will exactly match the way point values.

* + 1. The effect of the vertical interpolation to the trajectory:

The following examples illustrate the use of the vertical orientation parameter. For most use cases the effect of the vertical coordinate not being perpendicular to the corridor is not important as typical aircraft trajectories are not steep. For other uses cases the orientation it is important e.g. line of sight.

* + - 1. Example 1: vertical orientation in the vertical dimension at an angle to the trajectory:

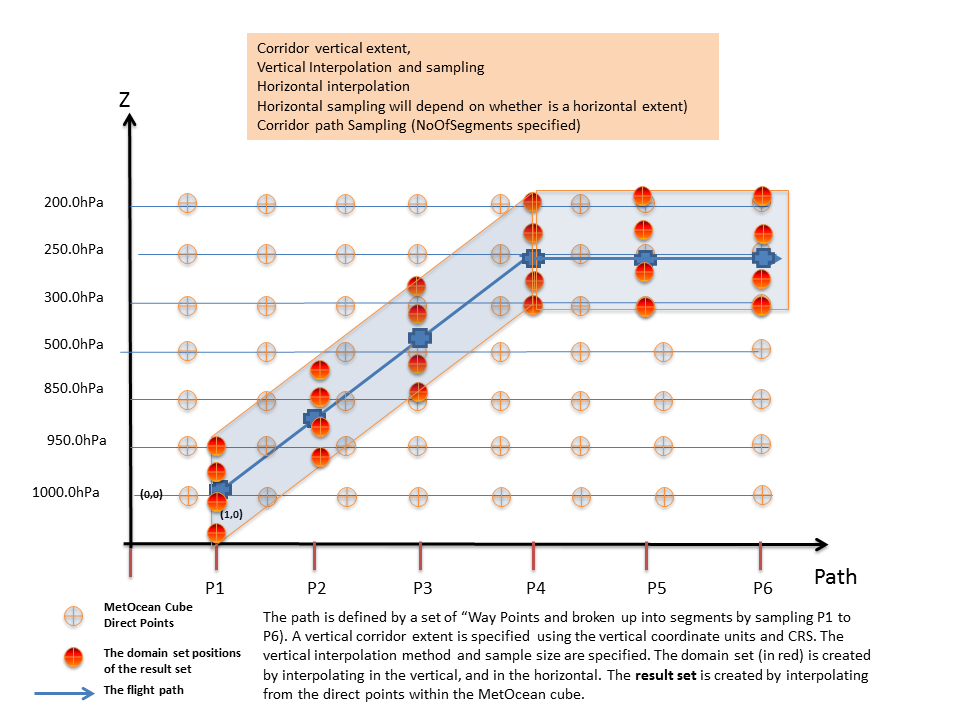
The path is defined by a set of “way points (P1 to P6). The domain set is a collection of points (in red) that extends in the vertical (using vertical sampling) for each segment point. The result setare the direct point values taken from the MetOcean coverage and interpolated to each point using horizontal/vertical interpolation (see Figure -- 16)

Figure -- 16 vertical orientation in the vertical dimension

* + - 1. Example 2: vertical orientation perpendicular to the trajectory

This is the same as example 1 except for the orientation of the vertical interpolation..

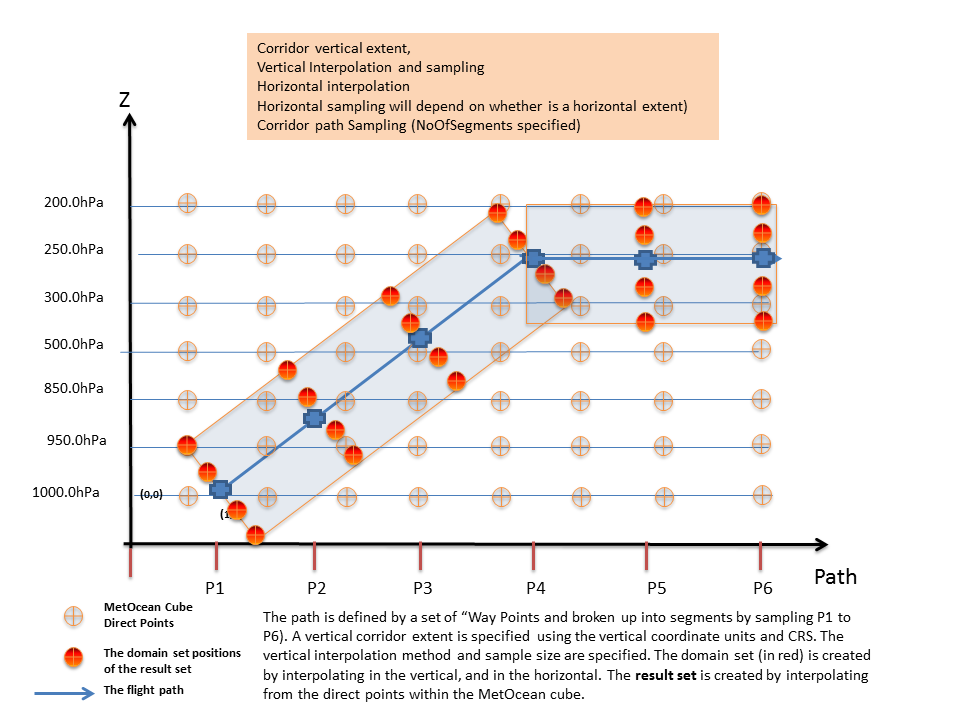


Figure 17 -- vertical orientation perpendicular to the corridor

1. GetMetOCeanCorridor Model (Normative)
   1. Requirements Class: GetMetOCeanCorridor

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/metocean/1.0/req/GetMetOceanCorridor** | |
| **Requirement** | **/request-structure**  The GetMetOceanCorridor request shall consist of a structure as defined in Figure 18 and Table 2, Table 3, Table 4 and Table 5 |
| **Requirement** | **/pathDescription**  The PathDescripton property of the GetMetOceanCorridor type shall be a type of gml:ReferenceableGridByArray. |
| **Requirement** | **/request-identifier**  The requested coverage must be a valid coverage offered by the server addressed. |
| **Requirement** | **/axis-name**  The value of the axis name shall be equal to one of the axisName names specified in the AxisName code list. |
| **Requirement** | **/vertical-orientation-mode**  The value of the vertical-orientation-mode name shall be equal to one of the vertical-orientation-mode names specified in the vertical-orientation-mode code list. |
| **Requirement** | **/point-selection-mode**  The value of the point-selection-mode name shall be equal to one of the point-selection-mode names specified in the point-selection-mode code list. |

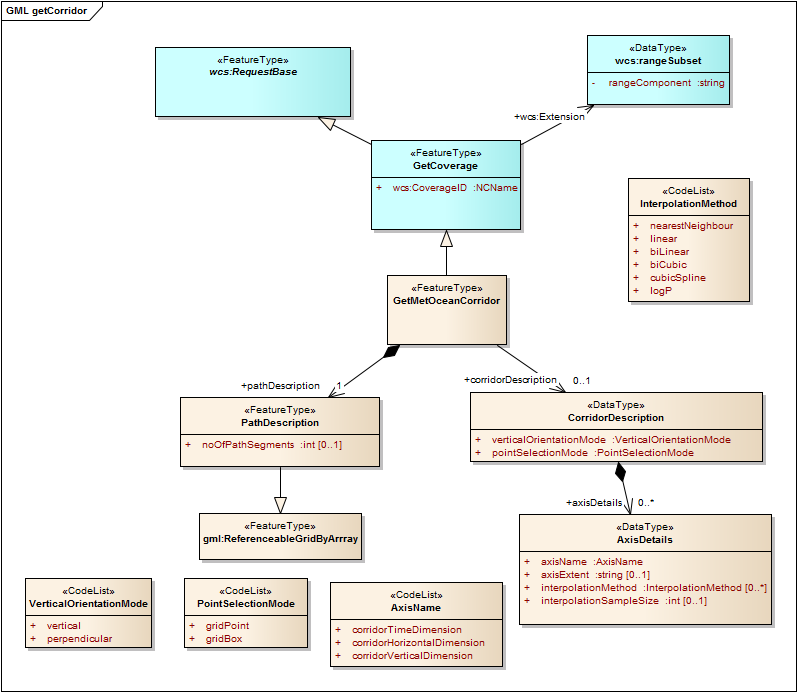


Figure 18 The UML model for GetMetOceanCorridor

* + 1. Requirements class overview

The MetOceanGetCorridor class defines the top level MetOceanGetCorridor that allows the extraction of a corridor from a multi-dimensional coverage

* + 1. GetMetOceanCorridor

This GetMetOceanCorridor extension extends the WCS2.0 core specification to allow for the extraction of data within a corridor or along a trajectory, from up to a 4D (x,y,z,t) coverage.

Table 2 GetMetOceanCorridor properties

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type** | **Multiplicity** |
| pathDescription | The route or course along which something travels or moves, for example the path of an aeroplane. | PathDescription | 1 |
| corridorDescription | A description of the corridor’s properties. | CorridorDescription | 0,\* |

* + 1. PathDescription.

This path description defines the trajectory as a set of way points defined using a referenceable grid. The trajectory may be multidimensional depending on the capabilities of the server.

Table 3 PathDescription properties

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type** | **Multiplicity** |
| noOfSegments | The route may be divided up into “n” segments. | int | 0,1 |
| GML:ReferenceableGridByArray | The sequence of points that defines the path | GML:ReferenceableGridByArray | 1 |

* + 1. GML:ReferenceableGridByArray.

The element group gml:geometricPositionListGroup specifies the array of grid point locations in the external coordinate reference system (through either a gml:posList element, or a sequence of gml:pos elements or gml:Point objects). The gml:sequenceRule element specifies the sequence order of these grid point locations over the grid.

Table 4 ReferenceableGridByArray Properties

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type** | **Multiplicity** |
| gml:posList | The route may be divided up into “n” segments. | int | 0,1 |
| Gml:sequenceRule | The sequenceRule element specifies the sequence order of these grid point locations over the grid. | :SequenceRuleType | 1 |
| gml:axisLabels | list of labels of the axes of the grid | gml:NCNameList | 1 |
| gml:limits | gml:limits element contains a single gml:GridEnvelope | Gml:GridEnvelope | 1 |

* + 1. CorridorDescription.

A corridor description defines the properties of the corridor

Table 5 Corridor Description Properties

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type** | **Multiplicity** |
| verticalOrientationMode | The orientation of the vertical coordinate relative to the path. | VerticalOrientationMode  (see | 1 |
| pointSelectionMode | Selection based on grid boxes or points | PointSelectionMode  (see | 1 |
| axisDetails | A description of each axis used by the corridor specification | AxisDetails | 0,\* |

* + 1. AxisDetails

This path description defines the trajectory as a set of way points defined using a referenceable grid. The trajectory may be multidimensional depending on the capabilities of the server.

Table 6 Axis Details Properties

|  |  |  |  |
| --- | --- | --- | --- |
| **Name** | **Definition** | **Data type** | **Multiplicity** |
| axisName | The name of the axis. | AxisName | 0,1 |
| axisExtent | The width of the named axis. | string | 0,1 |
| interpolationMethod | The method to be used by the interpolation scheme. |  | 0,\* |
| interPolationSampleSize | The sampling sixe to be used by the interpolation. | int | 0,1 |

* + 1. VerticalOrientationMode.

This path description defines the trajectory as a set of way points defined using a referenceable grid. The trajectory may be multidimensional depending on the capabilities of the server.

Table 7 – Vertical Orientation Mode code items

| **Code item** | **Definition** | **URL** |
| --- | --- | --- |
| vertical | The conversion relationship is the working version. This is the currently active conversion. | http://www.opengis.net/def/GetMetOceanCorridor/1.0/OrientationMoode/vertical |
| perpendicular | The conversion is under review. | http://www.opengis.net/def/GetMetOceanCorridor/1.0/OrientationMoode/perpendicular |

* + 1. PointSelectionMode.

The point selection mode determines how the grid points in the trajectory are selected on the basis of grid box boundary or the grid point is used. The examples in section 6 illustrate this in Figure 5 and Figure 6.

Table 8 – PointSelectionMode code items

| **Code item** | **Definition** | **URL** |
| --- | --- | --- |
| gridPoint | The “domain set” is made up of all the points that lie within the corridor (see Figure 6). | http://www.opengis.net/def/GetMetOceanCorridor/1.0/ PointSelectionMode/gridPoint |
| gridBox | The “domain set” uses the centre points of the “grid boxes” intersected by the trajectory (see Figure 5) | http://www.opengis.net/def/GetMetOceanCorridor/1.0/ PointSelectionMode/gridBox |

* + 1. AxisName.

This path description defines the trajectory as a set of way points defined using a referenceable grid. The trajectory may be multidimensional depending on the capabilities of the server.

Table 9 – AxisName code items

| **Code item** | **Definition** | **URL** |
| --- | --- | --- |
| corrridorTimeDimension | A reference to the time axis of the corridor | http://www.opengis.net/def/GetMetOceanCorridor/1.0/AxisName/corrridorTimeDimension |
| corrridorHorizontalDimension | A reference to the horizontal axis relative to the corridor. | http://www.opengis.net/def/GetMetOceanCorridor/1.0/AxisName/corrridorHorizontalDimension |
| corridorVerticalDimension | A reference to the vertical axis relative to the corridor. | http://www.opengis.net/def/GetMetOceanCorridor/1.0/AxisName/corridorVerticalDimension |

* + 1. InterpolationMethod.

This path description defines the trajectory as a set of way points defined using a referenceable grid. The trajectory may be multidimensional depending on the capabilities of the server.

Table 10 – InterpolationMethod code items

| **Code item** | **Definition** | **URL** |
| --- | --- | --- |
| nearestNeighbour | Choose the nearest grid point to the interpolated position | http://www.opengis.net/def/interpolation/OGC/1/nearest-neighbor |
| linear | Interpolate using a simple linear function. | http://www.opengis.net/def/interpolation/OGC/1/linear |
| biLinear | Interpolate using a simple bi-linear function. | http://www.opengis.net/def/interpolation/OGC/1/bi-Linear |
| biCubic | Interpolate using a simple bi-Cubic function. | http://www.opengis.net/def/interpolation/OGC/1/bi-Cubic |
| cubicSpline | Interpolate using a cubic spline function. | http://www.opengis.net/def/interpolation/OGC/1/cubic-Spline |
| linearLogP | Pressure varies logarithmically so this method interpolates linearly in log pressure. | http://www.opengis.net/def/interpolation/OGC/1/linear-LogP |

1. Amendments to WCS service model to support to support the GetMetOCeanCorridor:

The following list is returned in the Capabilities document of a server supporting the Metocean:GetCorridor extension.

Unordered list of identifiers of Application Profiles that are implemented by this server. This element should be included for each specified application profile implemented by this server. The identifier value should be specified by each Application Profile. If this element is omitted, no meaning is implied.

<ows:ServiceIdentification>  
 <ows:Title>Title</ows:Title>  
 <ows:Abstract>Abstract</ows:Abstract>  
 <ows:Keywords>  
 <ows:Keyword>EO-WCS</ows:Keyword>  
 <ows:Keyword>WCS 2.0</ows:Keyword>  
 <ows:Keyword>Keyword</ows:Keyword>  
 </ows:Keywords>  
 <ows:ServiceType codeSpace="OGC">OGC WCS</ows:ServiceType>  
 <ows:ServiceTypeVersion>2.0.1</ows:ServiceTypeVersion>  
 <ows:Profile>[http://www.opengis.net/spec/WCS\_service-extension\_crs/1.0/conf/crs</ows:Profile](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs%3c/ows:Profile)>  
 <ows:Profile>[http://www.opengis.net/spec/WCS/2.0/conf/core</ows:Profile](http://www.opengis.net/spec/WCS/2.0/conf/core%3c/ows:Profile)>  
 <ows:Profile>http://www.opengis.net/spec/WCS\_protocol-binding\_get-kvp/1.0/conf/get-  
 kvp</ows:Profile>  
 <ows:Profile>http://www.opengis.net/spec/WCS\_protocol-binding\_post-xml/1.0/conf/post-  
 xml</ows:Profile>  
 <ows:Profile>http://www.opengis.net/spec/GMLCOV/1.0/conf/gml-coverage</ows:Profile>  
 <ows:Profile>[http://www.opengis.net/spec/GMLCOV/1.0/conf/multipart</ows:Profile](http://www.opengis.net/spec/GMLCOV/1.0/conf/multipart%3c/ows:Profile)>  
 <ows:Profile>http://www.opengis.net/spec/GMLCOV\_geotiff-coverages/1.0/conf/geotiff-  
 coverage</ows:Profile>  
 <ows:Profile>http://www.opengis.net/spec/WCS\_service-model\_crs-predefined/1.0/conf/crs-  
 predefined</ows:Profile>  
 <ows:Profile>http://www.opengis.net/spec/WCS\_service-  
 extension\_interpolation/1.0/conf/interpolation-bi-cubic</ows:Profile>

<ows:Profile>http://www.opengis.net/spec/WCS\_service-  
 extension\_interpolation/1.0/conf/interpolation-nearest-neighbor</ows:Profile>

<ows:Profile>http://www.opengis.net/spec/WCS\_service-  
 extension\_interpolation/1.0/conf/interpolation-linear</ows:Profile>

<ows:Profile>http://www.opengis.net/spec/WCS\_service-  
 extension\_interpolation/1.0/conf/interpolation-linear-LogP</ows:Profile>

<ows:Profile>http://www.opengis.net/spec/WCS\_service-  
 model\_scaling+interpolation/1.0/conf/scaling+interpolation</ows:Profile>

<ows:Profile>[http://www.opengis.net/spec/WCS\_service--  
 extension\_ Metocean:GetCorridor /1.0/conf/ Metocean:GetCorridor </ows:Profile](http://www.opengis.net/spec/WCS_service-extension_coveragecollection/1.0/conf/coveragecollection%3c/ows:Profile)>  
 <ows:Profile>[http://www.opengis.net/spec/WCS\_service--  
 extension\_coveragecollection/1.0/conf/coveragecollection</ows:Profile](http://www.opengis.net/spec/WCS_service-extension_coveragecollection/1.0/conf/coveragecollection%3c/ows:Profile)>   
</ows:ServiceIdentification>

<wcs:ServiceMetadata>  
 <wcs:formatSupported>application/gml+xml</wcs:formatSupported>  
 <wcs:formatSupported>image/tiff</wcs:formatSupported>  
 <wcs:Extension>  
 <covcoll:coveragecollectionProfileSupported>  
 http://www.opengis.net/def/coveragecollection/OGC/1/metOcean  
 </covcoll:coveragecollectionProfileSupported>

<metoceancorr:GetMetOceanCorridorProfileSupported>http://www.opengis.net/spec/   
 WCS\_service-extension\_GetMetOceanCorridor/1.0/def/verticalOrientation-vertical

</metoceancorr :GetMetOceanCorridorProfileSupported >

<metoceancorr:GetMetOceanCorridorProfileSupported>http://www.opengis.net/spec/   
 WCS\_service-extension\_GetMetOceanCorridor/1.0/def/verticalOrientation-perpendicular

</metoceancorr :GetMetOceanCorridorProfileSupported >

<metoceancorr:GetMetOceanCorridorProfileSupported>http://www.opengis.net/spec/   
 WCS\_service-extension\_GetMetOceanCorridor/1.0/def/pointSelectionMode-gridPoint

</metoceancorr GetMetOceanGetCorridorProfileSupported >

<metoceancorr:GetMetOceanGetCorridorProfileSupported>http://www.opengis.net/spec/   
 WCS\_service-extension\_GetMetOceanCorridor/1.0/def/pointSelectionMode-gridBox

</metoceancorr :GetMetOceanCorridorProfileSupported >

<metoceancorr:GetMetOceanCorridorProfileSupported>http://www.opengis.net/spec/   
 WCS\_service-extension\_GetMetOceanCorridor/1.0/def/dimensionsSupported-2D-3D

</metoceancorr :GetMetOceanCorridorProfileSupported >  
 </wcs:Extension>  
 </wcs:ServiceMetadata>

* 1. Requirements Class: Get capabilities

This requirements class specifies the amendments to the *Get Capabilities* operation.

Table 11 Requirements class http://www.opengis.net/spec/WCS-service-extension\_GetMetOceanCorridor /1.0/req/getCapabilities

|  |  |
| --- | --- |
| **Requirements Class** | |
| **http://www.opengis.net/spec/WCS\_service-extension\_GetMetOceanCorridor /1.0/req/getCapabilities** | |
| Dependency | **http://www.opengis.net/doc/IS/WCS/2.0#clause:8.2** |
| Dependency | **http:/www.opengis.net/spec/WCS/2.0/req/core/getCapabilities** |
| Dependency | **http:/www.opengis.net/spec/WCS/2.0/req/core/wcsServiceMetadata-structure** |
| Dependency | **http:/www.opengis.net/spec/WCS/2.0/req/core/wcsServiceMetadata-contents** |
| Dependency | **http://www.opengis.net/spec/WCS\_service-extension\_ metOceanGetCorridor /1.0/req/GetMetOceanCorridor** |
| **Requirement** | **http://www.opengis.net/spec/WCS\_service-extension\_GetMetOceanCorridor/1.0/req/getCapabilities/response-extension-identification**  *A WCS server implementing this extension shall include the following URI in the Profile element of the ServiceIdentification in a GetCapabilities response:* *<ows:Profile>http://www.opengis.net/spec/WCS\_service--  extension\_ Metocean:GetCorridor /1.0/conf/ Metocean:GetCorridor </ows:Profile>* |
| **Requirement** | **http://www.opengis.net/spec/WCS\_service-extension\_ GetMetOceanCorridor /1.0/req/getCapabilities/response-supported-profiles**  *If a WCS server that implements this extension provides explicit support for a registered Coveragecollection profile, the URI of that profile shall be provided in the GetMetOceanCorridorProfileSupported element within the Extension element of the ServiceMetadata:*  *ServiceMetadata/Extension/metOceanGetCorridorProfileSupported*  *A WCS server must declare support for at least one of the following profiles:-*  *http://www.opengis.net/spec/   WCS\_service-extension\_GetMetOceanCorridor/1.0/def/ verticalOrientation*  *http://www.opengis.net/spec/   WCS\_service-extension\_GetMetOceanCorridor/1.0/def/pointSelectionMode http://www.opengis.net/spec/   WCS\_service-extension\_GetMetOceanCorridor/1.0/def/ dimensionsSupported* |

* + 1. Requirements class overview

The Get Capabilities operation of a WCS server implementing this extension provides summary information about its available GetMetOCeanCorridor resources. Please refer to section 6 for more details about the elements VerticalOrientationMode . PointSelectionMode and dimensionSupported.

In order to help client applications mitigate issues relating to parsing very large capabilities documents (e.g. due to the WCS server offering hundreds, if not thousands, of coverages) an additional term is added to the controlled vocabulary that is used in the sections element of the Get Capabilities request: offeredCollections. A client application may, therefore, request only the summary information about the Coveragecollection resources, omitting the summary information about the offered coverages themselves.

A WCS server implementing this extension shall also declare support for any registered Coveragecollection profiles.

1. [www.opengeospatial.org/cite](http://www.opengeospatial.org/cite) [↑](#footnote-ref-1)