

Open Geospatial Consortium Inc.

Date: 2006-10-17

Reference number of this OGC® Document: 06-083r8

Version: 1.1.0

Category: OpenGIS® Implementation Specification

Editors: Arliss Whiteside, John D. Evans

Web Coverage Service (WCS) Implementation Specification

Copyright © 2007 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Warnings

OGC official documents use a triple decimal-dot notation (i.e. MM.xx.ss). This document may be identified as MM.xx (Major.minor) and may include increments to the third dot series (schema changes) without any modification to this document, or the version displayed on the document. This means, for example, that a document labeled with versions 1.1.0 and 1.1.1 or even 1.1.9 are exactly the same except for modifications to the official schemas that are maintained and perpetually located at: <http://schemas.opengis.net/>

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	OpenGIS® Implementation Specification
Document subtype:	(none)
Document stage:	Approved Revision 1.1
Document language:	English

Contents	Page
i. Preface.....	x
ii. Document terms and definitions	x
iii. Submitting organizations.....	x
iv. Document contributors	x
v. Revision history	xi
vi. Changes to the OGC Abstract Specification.....	xi
vii. Future work	xi
1 Scope	1
2 Conformance	1
3 Normative references	1
4 Terms and definitions.....	2
5 Conventions.....	3
5.1 Symbols (and abbreviated terms).....	3
5.2 UML notation.....	3
5.3 Used parts of other documents.....	3
5.4 Platform-neutral and platform-specific specifications	3
5.5 Data dictionary tables.....	4
6 WCS overview	4
7 Shared aspects	6
7.1 Introduction	6
7.2 DomainType.....	6
7.3 Operation request encoding.....	6
7.4 XML Schema notes.....	7
7.5 Operation response encoding and reserved characters.....	7
7.6 CRS definitions and references	7
7.6.1 WCS use of CRSs.....	7
7.6.2 CRS definitions.....	7
7.6.3 CRS references	8
7.7 BoundingBox use in WCS	9
7.7.1 General terms.....	9
7.7.2 Spanning the antimeridian of a geographic CRS.....	9
7.7.3 Treatment of edge grid points.....	10
8 GetCapabilities operation (mandatory).....	10
8.1 Introduction	10
8.2 GetCapabilities operation request	11
8.2.1 Request contents	11
8.2.2 KVP encoding.....	11
8.2.3 XML encoding.....	11
8.2.4 SOAP encoding	12
8.3 GetCapabilities operation response.....	12

8.3.1	Normal response	12
8.3.2	ServiceIdentification section contents	13
8.3.3	OperationsMetadata section contents	14
8.3.3.1	Overview	14
8.3.3.2	Indicating allowed encodings	14
8.3.3.3	Indicating support for the “store” parameter.....	15
8.3.4	Contents section.....	15
8.3.4.1	Overview	15
8.3.4.2	CoverageSummary.....	17
8.3.4.3	Reserved characters in CoverageSummary parameter values	18
8.3.4.4	CoverageSummary hierarchy.....	19
8.3.4.5	OtherSource	20
8.3.4.6	Implementation requirements	20
8.3.4.7	XML encoding	21
8.3.5	Capabilities document XML encoding.....	21
8.3.6	GetCapabilities document SOAP encoding.....	21
8.3.7	Exceptions.....	21
8.4	GetCapabilities operation examples.....	21
9	DescribeCoverage operation (mandatory)	22
9.1	Introduction	22
9.2	DescribeCoverage operation request.....	23
9.2.1	DescribeCoverage request parameters.....	23
9.2.2	DescribeCoverage request KVP encoding.....	24
9.2.3	DescribeCoverage request XML encoding.....	24
9.2.4	DescribeCoverage request SOAP encoding	25
9.3	DescribeCoverage operation response	25
9.3.1	Normal response parameters.....	25
9.3.1.1	Coverage descriptions.....	25
9.3.1.2	Domain of coverage.....	26
9.3.1.3	Range of coverage.....	29
9.3.1.4	Implementation requirements	33
9.3.2	Supporting information.....	34
9.3.2.1	Scalar and vector fields.....	34
9.3.2.2	SupportedFormat.....	35
9.3.2.3	Reserved characters in CoverageDescription parameter values	36
9.3.2.4	Summary of ISO 8601 syntax for time positions and time periods	36
9.3.3	Normal response XML encoding.....	37
9.3.4	Normal response SOAP encoding	37
9.4	DescribeCoverage exceptions	37
9.5	DescribeCoverage operation examples	38
10	GetCoverage operation (mandatory).....	40
10.1	Introduction	40
10.2	GetCoverage operation request.....	40
10.2.1	GetCoverage request parameters	40
10.2.1.1	Overview.....	40
10.2.1.2	RangeSubset.....	43
10.2.1.3	Implementation requirements	45
10.2.2	GetCoverage request KVP encoding.....	46
10.2.2.1	Overview	46

10.2.2.2	RangeSubset KVP syntax	48
10.2.3	GetCoverage request XML encoding	48
10.2.4	GetCoverage request SOAP encoding	48
10.3	GetCoverage operation response	48
10.3.1	Introduction	48
10.3.2	GetCoverage exceptions	48
10.3.3	Coverage processing	49
10.3.4	Output coverage CRS	50
10.3.5	SupportedFormat and SupportedCRS dependencies	50
10.3.6	Output coverage spatial extent	50
10.3.7	Output coverage temporal extent	51
10.3.8	Output coverage range	51
10.3.9	Output coverage metadata	52
10.3.10	Output coverage formats	52
10.3.11	Response encodings	53
10.3.11.1	Overview	53
10.3.11.2	Coverages data structure	53
10.3.11.3	MIME multipart response	54
10.3.11.4	SOAP response	54
10.3.11.5	SOAP with Attachments response	54
10.3.12	Coverages XML encoding	55
10.4	GetCoverage operation examples	55
10.4.1	A minimal GetCoverage request	55
10.4.2	A longer GetCoverage request	56
10.4.3	Another GetCoverage request	57
10.4.4	Example responses	58
Annex A (normative)	Conformance	59
Annex B (informative)	WCS XML Schemas	60
Annex C (informative)	XML document examples	62
C.1	Introduction	62
C.2	GetCapabilities operation XML encoding examples	62
C.3	DescribeCoverage operation XML encoding examples	68
C.4	GetCoverage XML operation XML encoding examples	68
Annex D (informative)	UML model	69
D.1	Introduction	69
D.2	UML packages	69
D.3	WCS Service package	70
D.4	Get Coverage package	71
D.5	Range Subset package	73
D.6	Describe Coverage package	73
D.7	Domain package	75
D.8	Range package	75
D.9	WCS Get Capabilities package	78
D.10	WCS Contents package	79
Annex E (normative)	DomainType data structure	80
E.1	Overview	80
E.2	Domain typed parameter encoding	85

Annex F (normative) SOAP transfer	87
Annex G (normative) GridCRS	88
G.1 Introduction	88
G.2 GridCRS	88
G.2.1 GridCRS contents	88
G.2.2 Default values	89
G.2.3 XML encoding	90
G.2.4 KVP encoding	90
G.3 Operation methods for rectangular grids	92
G.3.1 Introduction	92
G.3.2 2D grid in 2D CRS (urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS)	92
G.3.3 2D grid in 3D CRS (urn:ogc:def:method:WCS:1.1:2dGridIn3dCrS)	93
G.3.4 2D simple grid (urn:ogc:def:method:WCS:1.1:2dSimpleGrid)	95
Annex H (informative) GetCoverage use cases	97
H.1 Introduction	97
H.2 Use cases	97
H.2.1 Overview	97
H.2.2 Use case 1: No resampling of georectified coverage, defining extent in ground CRS	98
H.2.3 Use case 2: No resampling of georectified image, defining extent in image CRS	98
H.2.4 Use case 3: No resampling of unrectified and un-georeferenced image	98
H.2.5 Use case 4: No resampling of unrectified but georeferenced image, defining extent in image CRS	99
H.2.6 Use case 5: No resampling of unrectified but georeferenced image, defining extent in ground CRS	99
H.2.7 Use case 6: Resampling of georectified coverage, defining extent in ground CRS	99
H.2.8 Use case 7: Resampling of georectified image, defining extent in image CRS	100
H.2.9 Use case 8: Resampling of unrectified and un-georeferenced image	100
H.2.10 Use case 9: Resampling of unrectified but georeferenced image, defining extent in image CRS	100
H.2.11 Use case 10: Resampling of unrectified but georeferenced image, defining extent in ground CRS	101
H.2.12 Addition A	101
H.3 Handling of use cases	102
H.3.1 Overview	102
H.3.2 Use case 1: No resampling of georectified coverage, defining extent in ground CRS	106
H.3.3 Use case 2: No resampling of georectified image, defining extent in image CRS	107
H.3.4 Use case 3: No resampling of unrectified and un-georeferenced image	108
H.3.5 Use case 4: No resampling of unrectified but georeferenced image, defining extent in image CRS	109
H.3.6 Use case 5: No resampling of unrectified but georeferenced image, defining extent in ground CRS	109
H.3.7 Use case 6: Resampling of georectified coverage, defining extent in ground CRS	110

H.3.8	Use case 7: Resampling of georectified image, defining extent in image CRS.....	111
H.3.9	Use case 8: Resampling of unrectified and un-georeferenced image.....	113
H.3.10	Use case 9: Resampling of unrectified but georeferenced image, defining extent in image CRS.....	114
H.3.11	Use case 10: Resampling of unrectified but georeferenced image, defining extent in ground CRS.....	115
H.3.12	Addition B.....	116
Annex I (normative)	Other data structures.....	118
I.1	Overview.....	118
I.2	Manifest.....	118
I.2.1	Manifest contents.....	118
I.2.2	XML encoding.....	122
I.3	Coverages data structure.....	123
I.3.1	Introduction.....	123
I.3.2	XML encoding.....	125
I.4	Interpolation methods.....	125
I.4.1	Interpolation methods contents.....	125
I.4.2	Interpolation methods XML encoding.....	127
I.4.3	Interpolation method KVP encoding.....	128
Bibliography	129

Figures	Page
Figure 1 — WCS interface UML class diagram.....	6
Figure 2 — Grid points, grid cells, and sample spaces.....	9
Figure 3 — WCS Service Metadata UML class diagram.....	12
Figure 4 — Contents UML class diagram.....	16
Figure 5 — DescribeCoverage operation request UML class diagram.....	23
Figure 6 — DescribeCoverage operation response UML class diagram.....	25
Figure 7 — Domain UML class diagram.....	27
Figure 8 — Range UML class diagram, part 1.....	30
Figure 9 — Range UML class diagram, part 2.....	31
Figure 10 — Example vector fields.....	34
Figure 11 — GetCoverage request UML class diagram.....	41
Figure 12 — RangeSubset UML class diagram.....	44
Figure 13 — Conceptual GetCoverage request evaluation sequence.....	50
Figure D.1 — WCS interface UML class diagram.....	69
Figure D.2 — WCS interface package diagram.....	70
Figure D.3 — WCS Service package class diagram.....	71
Figure D.4 — Get Coverage package class diagram.....	72
Figure D.5 — Range Subset package class diagram.....	73
Figure D.6 — Describe Coverage package class diagram.....	74
Figure D.7 — Domain package class diagram.....	75
Figure D.8 — Range package class diagram, part 1.....	76
Figure D.9 — Range package class diagram, part 2.....	77
Figure D.10 — WCS Get Capabilities package class diagram.....	78
Figure D.11 — WCS Contents package class diagram.....	79
Figure E.1 — Domain package class diagram.....	81
Figure G.1 — GridCRS class diagram.....	88
Figure H.1 — Parallelogram grid within a 2D X-Y space.....	101
Figure I.1 — Manifest UML package.....	119
Figure I.2 — Coverage Group package partial UML diagram.....	124
Figure I.3 — Interpolation Methods class diagram.....	126

Tables	Page
Table 1 — Contents of data dictionary tables.....	4
Table 2 — Implementation of parameters in GetCapabilities operation request.....	11
Table 3 — Section name values and contents	13
Table 4 — Additional row in ServiceIdentification data structure.....	13
Table 5 — Required values of OperationsMetadata section attributes.....	14
Table 6 — Parts of Contents section.....	17
Table 7 —CoverageSummary data structure	18
Table 8 — Inheritance of parts of CoverageSummary data structure.....	19
Table 9 — Parameters in DescribeCoverage operation request.....	24
Table 10 — DescribeCoverage request URL encoding.....	24
Table 11 — Parts of CoverageDescriptions data structure	26
Table 12 — Parts of CoverageDescription data structure.....	26
Table 13 — Parts of Domain data structure.....	27
Table 14 — Parts of SpatialDomain data structure.....	28
Table 15 — Parts of TimeSequence data structure	28
Table 16 — Parts of TimePositionOrInterval data structure	28
Table 17 — Parts of TimePeriod data structure.....	29
Table 18 — Parts of Range data structure	31
Table 19 — Parts of Field data structure	32
Table 20 — Parts of Axis data structure	33
Table 21 — Parts of AvailableKeys data structure.....	33
Table 22 — Exception codes for DescribeCoverage operation.....	38
Table 23 — Parts of GetCoverage operation request	42
Table 24 — Parts of DomainSubset data structure	43
Table 25 — Parts of Output data structure	43
Table 26 — Parts of RangeSubset data structure.....	44
Table 27 — Parts of FieldSubset data structure.....	44
Table 28 — Parts of AxisSubset data structure	45
Table 29 — GetCoverage request expressed as Key Value Pairs.....	47
Table 30 — Exception codes for GetCoverage operation	49
Table 31 — Required output coverage metadata.....	52
Table 32 — GetCoverage response encodings	53
Table E.1 — Parts of DomainType data structure	82
Table E.2 — Parts of UnNamedDomainType data structure.....	83

Table E.3 — Parts of PossibleValues data structure.....	83
Table E.4 — Parts of ValuesUnit data structure.....	84
Table E.5 — Parts of AllowedValues data structure.....	84
Table E.6 — Parameters in Range data structure.....	84
Table E.7 — Parameters in DomainMetadata data structure.....	85
Table E.8 — Parameters in ValuesReference data structure.....	85
Table G.1 — Parameters in GridCRS data structure.....	89
Table G.2 — GridCRS default values.....	90
Table G.3 — GridCRS encoded as Key Value Pairs.....	91
Table H.1 — GetCoverage use case numbers for combinations of conditions.....	97
Table H.2 — GetCoverage request DomainSubset BoundingBox CRS for use cases.....	103
Table H.3 — GetCoverage request Output CRS for use cases.....	104
Table H.4 — GetCoverage response BoundingBox CRS for use cases.....	105
Table I.1 — Manifest data structure.....	120
Table I.2 — ReferenceGroup data structure.....	121
Table I.3 — ReferenceBase data structure.....	121
Table I.4 — Reference data structure.....	122
Table I.5 — Parts of the InterpolationMethods data structure.....	126
Table I.6 — Parts of the InterpolationMethod data structure.....	127
Table I.7 — Normal spatial interpolation types.....	127
Table I.8 — Effect of null values in spatial interpolation.....	127

i. Preface

This is WCS version 1.1, and builds on OWS Common [OGC 05-008 and OGC 05-009], including change proposal [OGC 04-105]. It also builds on the XML Schemas for WCS used in the IH4DS thread of OWS-2, recorded in [OGC 04-080]. It thus includes many of the proposed WCS changes summarized in [OGC 05-018].

ii. Document terms and definitions

This document uses the specification terms defined in Subclause 5.3 of [OGC 05-008], 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 specification.

iii. Submitting organizations

The following organizations have submitted this Implementation Specification to the Open Geospatial Consortium, Inc.

- BAE SYSTEMS E&IS
- Australia Commonwealth Scientific and Industrial Research Organisation (CSIRO)
- CubeWerx Inc.
- George Mason University
- International University Bremen / Rasdaman GmBH
- IONIC SOFTWARE s.a.
- U.S. National Aeronautics and Space Administration (NASA)
- U.S. National Geospatial-Intelligence Agency (NGA)
- Oracle Corp.
- PCI Geomatics

iv. Document contributors

OGC’s Web Coverage Service 1.1 Revision Working Group members are listed below.

Name	Organization
Baumann, Peter	Intl. University Bremen / Rasdaman GmBH
Cox, Simon	CSIRO
Di, Liping	George Mason University
Donéa, Luc	IONIC SOFTWARE
Evans, John	NASA / GST, Inc
Herring, John	Oracle Corporation
Keens, Steven	PCI Geomatics
Roswell, Charles	NGA
Vretanos, Panagiotis	CubeWerx, Inc.
Whiteside, Arliss	BAE Systems E&IS

v. Revision history

WCS 1.1 is almost a complete rewrite of WCS 1.0, with many editorial changes and detailed streamlining of schema files resulting from OWS Common conformance. However, the two versions of the specification share the same overall structure. WCS 1.1 introduces (only) the following functional changes:

- a) Use of GridCRS in coverage descriptions and requests
(See 9.3.1.2, 10.2.1.1, 10.2.2, 10.3.4, 10.3.6, and Annexes G and H)
- b) Hierarchical coverage descriptions (See 8.3.4.4)
- c) Multiple fields per coverage
(See 9.3.1.3, 9.3.2.1, 10.2.1.2, 10.2.2.1, 10.2.2.2, and Tables 18, 26-29)
- d) XML and (optionally) asynchronous GetCoverage responses (See 10.3.11)
- e) Different approach to coverage formats; reliance on encoding profiles (See 9.3.2.2)

Many of the changes were the direct or indirect result of addressing written change proposals, as summarized in OGC Doc. #06-116.

vi. Changes to the OGC Abstract Specification

OGC® Abstract Specification Topic 6: The Coverage Type and its Subtypes (which includes ISO 19123) requires the following change to accommodate the technical contents of this document:

Spatial interpolation needs to be a (repeatable) property of a single field in the range of a coverage, not a property of the coverage or of its range as a whole. Currently, the `interpolationType` is an attribute of the `CV_ContinuousQuadrilateralGridCoverage` class, which is associated to one `CV_GridValuesMatrix`. This `CV_GridValuesMatrix` contains a sequence of values of `Record` type, where each `Record` can contain multiple “fields”. However, different fields may require the use of different interpolation types; and more than one interpolation method may be associated with a single field.

vii. Future work

Future versions of this WCS specification are expected to consider various expansions of the abilities specified herein, some adding abilities that were deliberately not included in this Version 1.1.0. Some of the possible expansions thus include:

- a) Expand supported coverage types beyond grid coverages.
- b) Expand ability to retrieve elevation subset of a coverage, beyond current regularly spaced (grid) elevations.
- c) Expand ability to retrieve spatial subset of a coverage, beyond current regularly spaced (grid) positions.
- d) Expand ability to store coverage outputs, for later retrieval.

- e) Add WSDL description of SOAP encodings
- f) Add ability to Retrieve multiple coverages in one GetCoverage operation request
- g) Add optional interface operation(s) to input new coverages and update existing coverages stored by a WCS server
- h) Expand Annex A to specify abstract test suites.

Foreword

This version 1.1 of the Web Coverage Service (WCS) Specification supersedes previous version 1.0 [OGC 03-065r5 / 05-076]. Technical changes from the version 1.0 include building on the OGC Web Services Common Specification [OGC 05-008] and a substantially revised Capabilities schema; new schemas and syntax for operation requests (GetCoverage, DescribeCoverage); and integration with GML 3.1.

This document includes nine annexes; Annexes A, E, F, G, and I are normative, and Annexes B, C, D, and H are informative.

Some 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.

Introduction

The Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space-varying phenomena.

A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi-valued coverages, and input into scientific models and other clients. The WCS may be compared to the OGC Web Map Service (WMS) and the Web Feature Service (WFS); like them it allows clients to choose portions of a server's information holdings based on spatial constraints and other criteria.

Unlike the WMS [OGC 04-024], which portrays spatial data to return static maps (rendered as pictures by the server), the Web Coverage Service provides available data together with their detailed descriptions; defines a rich syntax for requests against these data; and returns data with its original semantics (instead of pictures) which may be interpreted, extrapolated, etc. – and not just portrayed.

Unlike WFS [OGC 02-058], which returns discrete geospatial features, the Web Coverage Service returns coverages representing space-varying phenomena that relate a spatio-temporal domain to a (possibly multidimensional) range of properties.

The Web Coverage Service provides three operations: *GetCapabilities*, *DescribeCoverage*, and *GetCoverage*. The *GetCapabilities* operation returns an XML document describing the service and brief descriptions of the coverages that clients may request. Clients would generally run the *GetCapabilities* operation and cache its result for use throughout a session, or reuse it for multiple sessions. When the *GetCapabilities* operation does not return such descriptions, then equivalent information must be available from a separate source, such as an image catalog.

The *DescribeCoverage* operation lets clients request a full description of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.

The *GetCoverage* operation is normally run after *GetCapabilities* and *DescribeCoverage* operation responses have shown what requests are allowed and what data are available. The *GetCoverage* operation returns a coverage (that is, values or properties of a set of geographic locations), encoded in a well-known coverage format. Its syntax and semantics bear some resemblance to the WMS *GetMap* and WFS *GetFeature* requests, but several extensions support the retrieval of coverages rather than static maps or discrete features.

Web Coverage Service (WCS) Implementation Specification

1 Scope

This document specifies how a Web Coverage Service (WCS) describes and delivers multi-dimensional coverage data over the World Wide Web. This version of the Web Coverage Service is limited to describing and requesting grid (or "simple") coverages.

Grid coverages have a domain comprised of regularly spaced locations along 0, 1, 2, or 3 axes of a spatial coordinate reference system. Their domain may also have a time dimension, which may be regularly or irregularly spaced. A coverage defines, at each location in the domain, a set of fields that may be scalar-valued (such as elevation), or vector-valued (such as brightness values in different parts of the electromagnetic spectrum). These fields (and their values) are known as the range of the coverage.

The WCS interface, while limited in this version to regular grid coverages, is designed to extend in future versions to other coverage types defined in OGC Abstract Specification Topic 6, "The Coverage Type" [OGC 00-106].

2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in Annex A (normative).

3 Normative references

The following normative documents contain provisions that, through reference in this text, constitute provisions of this specification. 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.

IETF RFC 2387, E. Levinson, *The MIME Multipart/Related Content-type*

IETF RFC 2396, T. Berners-Lee, R. Fielding, U.C. Irvine, L. Masinter: Uniform Resource Identifiers (URI): Generic Syntax. IETF, RFC 2396, August 1998

ISO 19123:2005, *Geographic Information — Coverage Geometry and Functions*

OGC 00-106, *The OpenGIS[™] Abstract Specification Topic 6: The Coverage Type and its Subtypes*

OGC 04-021r3, OGC[™] Catalogue Services Specification

OGC 04-024, OGC™ Web Map Service

OGC 05-094r1, *GML 3.1.1 CRS support profile*

OGC 05-095r1, *GML 3.1.1 common CRSs profile*

OGC 05-096r1, *GML 3.1.1 grid CRSs profile*

OGC 05-103, *Geographic information — Spatial referencing by coordinates (ISO/DIS 19111)*

OGC 05-008, *OGC Web Services Common Specification*

W3C Note, J. Barton, S. Thatte, H. Nielsen, *SOAP Messages With Attachments*, December 11, 2000

NOTE The OGC Web Services Common Specification (OGC 05-008) contains a list of normative references that are also applicable to this Implementation Specification.

In addition to this document, this specification includes normative XML Schema files. These are posted online at the URL <http://schemas.opengis.net/wcs/1.1.0/>. These XML Schema files are also bundled with the present document. In the event of a discrepancy between the bundled and online versions of the XML Schema files, the online files shall be considered authoritative.

4 Terms and definitions

For the purposes of this specification, the definitions given in OWS Common Implementation Specification [OGC 05-008] and in OGC Abstract Specification Topic 2: Spatial referencing by coordinates [OGC 05-103] and shall apply. In addition, the following terms and definitions apply.

4.1

coverage

feature that acts as a function to return values from its range for any direct position within its spatiotemporal domain

4.2

georectified grid

grid having regular spacing in a projected or geographic coordinate reference system (CRS)

NOTE A grid for which there is a linear relationship between the grid coordinates and those of a projected or geographic coordinate reference system.

4.3

georeferenced grid

grid that is not georectified, but is associated with (one or more) coordinate transformations which relate the image or engineering CRS to a projected or geographic CRS

NOTE These coordinate transformations are usually not affine or simple, and are usually empirically determined. (Synonym: *georeferenceable*).

5 Conventions

5.1 Symbols (and abbreviated terms)

Most of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 05-008] also apply to this document.

5.2 UML notation

All the diagrams that appear in this specification are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of the OGC Web Services Common Implementation Specification [OGC 05-008].

5.3 Used parts of other documents

This document uses significant parts of document [OGC 05-008]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a gray background (15%).

5.4 Platform-neutral and platform-specific specifications

As specified in Clause 10 of OGC Abstract Specification Topic 12 “OpenGIS Service Architecture” (which contains ISO 19119), this document includes both Distributed Computing Platform-neutral and platform-specific specifications. This document first specifies each operation request and response in platform-neutral fashion. This is done using a table for each data structure, which lists and defines the parameters and other data structures contained. These tables serve as data dictionaries for the UML model in Annex D, and thus specify the UML model data type and multiplicity of each listed item.

EXAMPLES 1 Platform-neutral specifications are contained in Subclauses 8.2.1, 8.3.1, 8.3.2, 8.3.4.1 to 8.3.4.5, 9.2.1, 9.3.1, 10.2.1, 10.3.9, and 10.3.11.

The specified platform-neutral data could be encoded in many alternative ways, each appropriate to one or more specific DCPs. This document now specifies encoding appropriate for use of HTTP GET transfer of operations requests (using KVP encoding), and for use of HTTP POST transfer of operations requests (using XML or KVP or SOAP encoding). However, the same operation requests and responses (and other data) could be encoded for other specific computing platforms.

EXAMPLES 2 Platform-specific specifications for KVP encoding are contained in Subclauses 8.2.2, 9.2.2, and 10.2.2.

EXAMPLES 3 Platform-specific specifications for XML encoding are contained in Subclauses 8.2.3, 8.3.4.7, 8.3.5, 9.2.3, 9.3.3, 10.2.3, and 10.3.13.

5.5 Data dictionary tables

The UML model data dictionary is specified herein in a series of tables. The contents of the columns in these tables are described in Table 1. The contents of these data dictionary tables are normative, including any table footnotes.

Table 1 — Contents of data dictionary tables

Column title	Column contents
Names (left column)	Two names for each included parameter or association (or data structure). The first name is the UML model attribute or association role name. The second name uses the XML encoding capitalization specified in Subclause 11.6.2 of [OGC 05-008]. Some names in the tables may appear to contain spaces, but no names contain spaces.
Definition (second column)	Specifies the definition of this parameter (omitting un-necessary words such as “a”, “the”, and “is”). When the parameter value is the identifier of something, not a description or definition, the definition of this parameter should read something like “Identifier of TBD”.
Data type and value (third column) <i>or</i> Data type (when no second items are included in rows of table)	Normally contains two items: The mandatory first item is often the data type used for this parameter, using data types appropriate in a UML model, in which this parameter is a named attribute of a UML class. Alternately, the first item can identify the data structure (or class) referenced by this association, and reference a separate table used to specify the contents of that class (or data structure). The optional second item in the third column of each table should indicate the source of values for this parameter, the alternative values, or other value information, unless the values are quite clear from other listed information.
Multiplicity and use (right or fourth column) <i>or</i> Multiplicity (when no second items are included in rows of table)	Normally contains two items: The mandatory first item specifies the multiplicity and optionality of this parameter in this data structure, either “One (mandatory)”, “One or more (mandatory)”, “Zero or one (optional)”, or “Zero or more (optional)”. (Yes, these are redundant, but we think ISO wants this information.) The second item in the right column of each table should specify how any multiplicity other than “One (mandatory)” shall be used. When that parameter is optional, under what condition(s) shall that parameter be included or not included? When that parameter can be repeated, for what is that parameter repeated?

When the data type used for this parameter, in the third column of such a table, is an enumeration or code list, all the values specified by a specific OWS shall be listed, together with the meaning of each value. When this information is extensive, these values and meanings should be specified in a separate table that is referenced in the third column of this table row.

6 WCS overview

The specified Web Coverage Service (WCS) interface supports electronic retrieval of geospatial data as “coverages” – that is, digital geospatial information representing space-varying phenomena. A WCS provides access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering and input into scientific models and other clients.

The WCS interface specifies three operations that may be requested by a WCS client and performed by a WCS server:

- a) *GetCapabilities* (required implementation by servers) – This operation allows a client to request the service metadata (or *Capabilities*) document. This XML document describes the abilities of the specific server implementation, usually including brief descriptions of the coverages available on the server. This operation also supports negotiation of the specification version being used for client-server interactions. Clients would generally request the *GetCapabilities* operation and cache its result for use throughout a session, or reuse it for multiple sessions. When the *GetCapabilities* operation does not return descriptions of its available coverages, that information must be available from a separate source, such as an image catalog.
- b) *DescribeCoverage* (required implementation by servers) – This operation allows a client to request full descriptions of one or more coverages served by a particular WCS server. The server responds with an XML document that fully describes the identified coverages.
- c) *GetCoverage* (required implementation by servers) – This operation allows a client to request a coverage comprised of selected range properties at a selected set of geographic locations. The server extracts the response data from the selected coverage, and encodes it in a known coverage format. The *GetCoverage* operation is normally run after *GetCapabilities* and *DescribeCoverage* operation responses have shown what requests are allowed and what data are available.

Many of these interface aspects that are common with other OGC Web Services (OWSs) are specified in the OGC Web Services Common Specification [OGC 05-008]. These common aspects are normatively referenced herein, instead of being repeated in this specification.

Figure 1 is a simple UML class diagram summarizing the WCS interface. This class diagram shows that the *WebCoverageServer* interface class inherits the *getCapabilities* operation from the *OGCWebService* interface class, and adds the *getCoverage* and *describeCoverage* operations. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the WCS interface is provided in Annex D (informative).

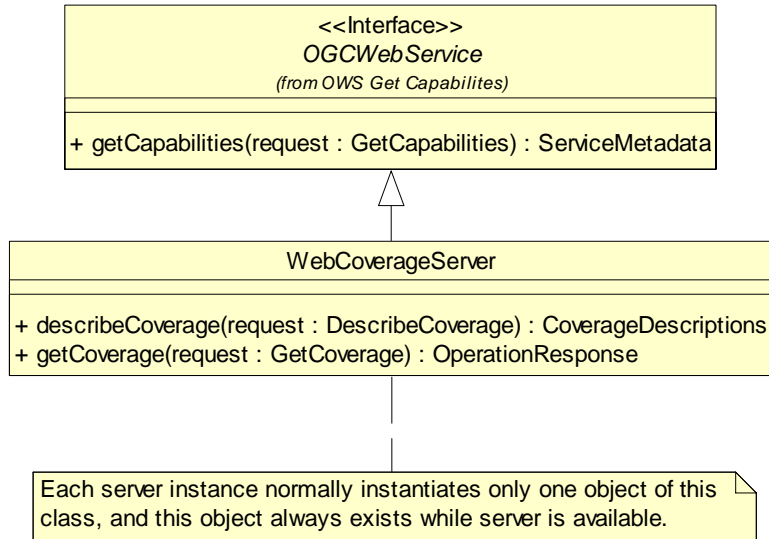


Figure 1 — WCS interface UML class diagram

NOTE In this UML diagram, the request and response for each operation is shown as a single parameter that is a data structure containing multiple lower-level parameters, which are discussed in subsequent clauses. The UML classes modeling these data structures are included in the complete UML model in Annex D.

Each of the three operations is described in more detail in subsequent clauses.

7 Shared aspects

7.1 Introduction

This clause specifies aspects of the WCS behavior that are independent of particular operations or are shared by several operations.

7.2 DomainType

The WCS uses a separate XML Schema for defining allowed values, named `owsDomainType.xsd`, also attached to this document. Since this `owsDomainType.xsd` is expected to be used by more than one OWS, it has been defined in the “ows” namespace. This WCS version uses the `owsDomainType.xsd` included in WPS [OGC 05-007r4], copied in Annex E of this WCS Implementation specification.

7.3 Operation request encoding

All WCS operation requests may use HTTP GET with KVP encoding or use HTTP POST with either XML or SOAP encoding, as specified in Clause 11 of [OGC 05-008].

7.4 XML Schema notes

The data type of many parameters is specified as “Character String type, not empty”. In the XML Schemas specified, these parameters are encoded with the `xsd:string` type, which does NOT require that these strings not be empty.

7.5 Operation response encoding and reserved characters

All operation responses shall comply with the XML syntax. In addition, several parameter values provided in *CoverageSummary* and *CoverageDescription* operation responses shall be usable literally (“as-is”) in subsequent *DescribeCoverage* and *GetCoverage* operation requests to the specific server, encoded in KVP or XML. Therefore, in stating values for these parameters, servers shall avoid, encode, or escape any “special” characters that could be interpreted as part of either the XML markup (`<` `>` `--` `&` *etc.*) or the KVP syntax (`?` `=` `&` *etc.*) of subsequent requests.

In particular, to support XML requests, operation responses shall avoid special characters, or encode or escape them using character or numeric entities (e.g., `>` or `>` in lieu of `>`) as defined in XML [W3C 2004]. To support KVP requests, operation responses shall avoid special characters, or URL-encode them (e.g., `%3F` for `?`) as defined in [RFC 2396].

7.6 CRS definitions and references

7.6.1 WCS use of CRSs

WCS operation requests and responses frequently include references to a defined Coordinate Reference System (CRS), and sometimes include a CRS definition. A defined CRS is required for spatial referencing by coordinates, and is a coordinate system which is related to an object by a datum. This specification makes use of material in several other OGC documents that specify how to define and reference CRSs.

In many cases, WCS operation requests and responses transfer only references to defined CRSs, not the actual CRS definitions. Each referenced CRS shall be defined somewhere, but that definition may not be electronically accessible or encoded in XML. Nonetheless, the referenced definitions often need to be used by WCS server software.

7.6.2 CRS definitions

OGC Abstract Specification Topic 2 “*Geographic information — Spatial referencing by coordinates*” [OGC 05-103], which contains ISO/DIS 19111, specifies how to define many types of CRSs, including the Datums and Coordinate Systems that define a CRS. Clause 7 of that document specifies the following types of CRSs, all of which might be used with a WCS grid coverage:

- f) GeodeticCRS (formerly GeographicCRS and GeocentricCRS)
- g) ProjectedCRS
- h) VerticalCRS

- i) CompoundCRS
- j) ImageCRS
- k) DerivedCRS
- l) EngineeringCRS

Clause 12 of GML 3.1.1 [OGC 03-105r1] specifies how to XML encode the definitions of all these types of CRSs, based on an older version of OGC Abstract Specification Topic 2 [OGC 04-046r3]. Three GML 3.1.1 profiles extract the parts of GML 3.1.1 that appear useful to a WCS, namely:

- a) OGC 05-094r1, *GML 3.1.1 CRS support profile*, which is used by the following two profiles
- b) OGC 05-095r1, *GML 3.1.1 common CRSs profile*, which covers GeographicCRS, ProjectedCRS, VerticalCRS, and CompoundCRS definitions
- c) OGC 05-096r1, *GML 3.1.1 grid CRSs profile*, which covers ImageCRS and DerivedCRS definitions

Annex G in this document specifies how to XML encode a GridCRS definition, in a simplified manner that is loosely based on a DerivedCRS.

7.6.3 CRS references

Clause 10.3 of OWS Common [OGC 05-008] specifies how CRSs shall be referenced. A URL can be used to reference a definition that is available using this URL. When not in the same XML document, those definitions shall be electronically available using this URL. Alternately, a URN in the “ogc” URN namespace can be used, and often should be used, to reference a CRS definition identified in that “ogc” URN namespace.

The format of URNs in the “ogc” namespace is recommended in Clause 7 of OGC Best Practices Paper [OGC 06-023r1], “*Definition identifier URNs in OGC namespace*”. Some recommended URN formats include parameter values that allow that URN to be used for an entire family of CRSs (such as different origins).

Numerous specific URN values for CRSs and components are currently specified in:

- a) Clause 8 and Annex A of [OGC 06-023r1], “*Definition identifier URNs in OGC namespace*”, for numerous CRSs and components
- b) Clause 10 of OGC 05-096r1, *GML 3.1.1 grid CRSs profile*, for ImageCRSs, DerivedCRSs, and components (including URNs for most possible 2D ImageCRSs)
- c) Subclause G.3 in this document, for three specific OperationMethods (expected to be referenced by most 2D GridCRS definitions)

7.7 BoundingBox use in WCS

7.7.1 General terms

This specification uses the BoundingBox and WGS84BoundingBox data structures, specified in Subclause 10.2 of OWS Common [OGC 05-008], to express the minimum rectangular bounding region around (available or requested) coverage data. In a BoundingBox data structure, the LowerCorner defines the edges of the (available or requested) region in the directions of decreasing coordinate values in a CRS (normally the algebraic minimum coordinates), and the UpperCorner defines the edges in the directions of increasing coordinate values (usually the algebraic maximum coordinates). For continuous (interpolatable) grid coverages, these values shall be computed from the areas of all the (available or requested) coverage grid cells with all corners at recorded grid points.

NOTE 1 Inclusion of the areas of all grid cells is not necessarily the same as inclusion of all the grid points, when the grid coverage spans the antimeridian as discussed below.

In keeping with ISO 19123, WCS considers only grids whose points lie at the corners (not the centres) of the grid cells. This document thus primarily considers “continuous” grid coverages, not “discrete” grid coverages, as those terms are defined in ISO 19123. As illustrated in Figure 2 below, grid points (a, b, c, d) define a grid cell (U) and are surrounded by “sample spaces” (A, B, C, D) as those terms are used in Subclause 8.2.2 of ISO 19123.

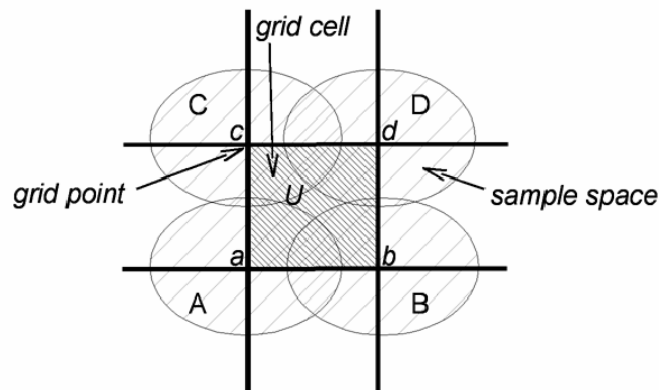


Figure 2 — Grid points, grid cells, and sample spaces

NOTE 2 For these sample spaces, this document avoids using terms such as “sample grid” or “pixels”, although such terms are often used elsewhere.

The following subclauses clarify the use of such a bounding box near discontinuities in a geographic CRS, and the handling of grid points at the edges of the coverage.

7.7.2 Spanning the antimeridian of a geographic CRS

In a geographic CRS, longitude values are often defined with a finite extent (e.g., from -180 degrees to +180 degrees). The minimum and maximum longitude limits define a single line (on the ellipsoid, sphere, or cylinder), known as the antimeridian, across which longitude values are discontinuous: as this line is crossed, longitude changes abruptly (e.g., going West

from a little more than -180 degrees to a little less than +180 degrees). This necessitates an “extended” interpretation of the Bounding Box definition:

A Bounding Box defined in a geographic CRS (or a WGS84BoundingBox) whose LowerCorner longitude is greater than that of its UpperCorner shall describe a region that crosses the longitude discontinuity.

EXAMPLE A bounding region encompassing the Pacific Ocean could be described by a BoundingBox with a positive LowerCorner longitude and a negative UpperCorner longitude.

NOTE Users and implementers may avoid using the above “extended” interpretation of the Bounding Box by EITHER (a) using a CRS whose value discontinuity lies outside the minimum bounding region (e.g., a geographic CRS with a different prime meridian); OR (b) treating longitude as unbounded but periodic (using the XML Schema strings “-INF” and “INF” to denote negative and positive infinity). (See also Subclause 10.2.5 of OWS Common [OGC 05-008].)

If the data represented by the coverage is continuous across the antimeridian, then a server should perform spatial interpolation near and across the antimeridian just as it does elsewhere.

7.7.3 Treatment of edge grid points

The spatial extent of a grid coverage extends only as far as the outermost grid points contained in the bounding box. It does NOT include any area (partial or whole grid cells or sample spaces) beyond those grid points.

NOTE This bounding box is NOT the extent sometimes considered, which also includes rectangular sample spaces (pixels) centered on the outermost grid points – as indicated in Subclause 7.3.3 6 of WMS 1.3 [OGC 04-042]. Such pixel extents are often poor approximations of the sensor physics / grid data collection process.

EXAMPLE The extent of a global data set in WGS84 geographic decimal degrees with a grid spacing of 1.0 degree might be expressed as a WGS84BoundingBox with LowerCorner and UpperCorner longitudes of (respectively) -179.5 and +179.5, or -179.0 and +180.0, when that data set is not considered continuous (interpolatable) across the antimeridian at 180 degrees. When that data set is interpolatable across the antimeridian, the LowerCorner and UpperCorner longitudes (respectively) might be -180 and +180 (with redundant first and last columns of data), or -INF and INF.

8 GetCapabilities operation (mandatory)

8.1 Introduction

The mandatory GetCapabilities operation allows WCS clients to retrieve service metadata from a WCS server. The response to a GetCapabilities request shall be an XML document containing service metadata about the server, usually including summary information about the data collections from which coverages may be requested. This clause specifies KVP and XML encoding of a GetCapabilities request and the XML document that a WCS server shall return to describe its capabilities.

8.2 GetCapabilities operation request

8.2.1 Request contents

The GetCapabilities operation request shall be as specified in Subclauses 7.2.2 through 7.2.4 of [OGC 05-008]. The value of the “service” parameter shall be “WCS”. The allowed set of service metadata (or Capabilities) XML document section names and meanings shall be as specified in Tables 3 and 7 in Subclauses 7.3.3 and 7.4.2 of [OGC 05-008].

The “Multiplicity and use” column in Table 1 of [OGC 05-008] specifies the optionality of each listed parameter in the GetCapabilities operation request. Table 2 specifies the implementation of those parameters by WCS clients and servers.

Table 2 — Implementation of parameters in GetCapabilities operation request

Names	Multiplicity	Server implementation	Client implementation
service	One (mandatory)	Each parameter shall be implemented by all servers ^a	Each parameter shall be implemented by all clients, using specified value
request	One (mandatory)		
Accept Versions	Zero or one (optional)	Shall be implemented by all servers	Should be implemented by all software clients, using allowed values
Sections	Zero or one (optional)	Each parameter should be implemented by each server ^{a, b}	Each parameter may be implemented by each client ^c
update Sequence	Zero or one (optional)		
Accept Formats	Zero or one (optional)		
<p>a Server shall check that each value received is an allowed value.</p> <p>b When this parameter is not implemented or not received, server shall provide the specified default response. When implemented and received, server shall provide specified response.</p> <p>c When this parameter is not provided, client shall expect the specified default response. When provided, client shall allow default or specified response.</p>			

All WCS servers shall implement HTTP GET transfer of the GetCapabilities operation request, using KVP encoding. Servers may also implement HTTP POST transfer of the GetCapabilities operation request, using XML encoding only.

8.2.2 KVP encoding

The KVP encoding of a WCS GetCapabilities operation request shall be as specified in Table 2 in Subclause 7.2.2 of [OGC 05-008].

8.2.3 XML encoding

The XML Schema fragment for encoding a WCS GetCapabilities operation request extends ows:GetCapabilitiesType in owsGetCapabilities.xsd from [OGC 05-008]:

```
<element name="GetCapabilities">
  <annotation>
    <documentation>Request to a WCS to perform the GetCapabilities
    operation. This operation allows a client to retrieve a Capabilities XML
    document providing metadata for the specific WCS server. In this XML
```

```

encoding, no "request" parameter is included, since the element name
specifies the specific operation. </documentation>
</annotation>
<complexType>
  <complexContent>
    <extension base="ows:GetCapabilitiesType">
      <sequence/>
      <attribute name="service" type="ows:ServiceType"
use="required" fixed="WCS"/>
    </extension>
  </complexContent>
</complexType>
</element>

```

8.2.4 SOAP encoding

WCS servers may implement SOAP version 1.2 transfer of the GetCapabilities operation request as specified in Annex F, using the XML encoding specified above.

8.3 GetCapabilities operation response

8.3.1 Normal response

The normal response to a GetCapabilities operation request shall be a service metadata (or Capabilities) document as specified in Subclause 7.4.2 of OWS common [OGC 05-008], and shown in the UML class diagram in Figure 3. The WCS service metadata document shall contain the four optional sections specified in Table 3. Depending on the values in the Sections parameter of the GetCapabilities operation request, any combination of these sections may be requested and returned.

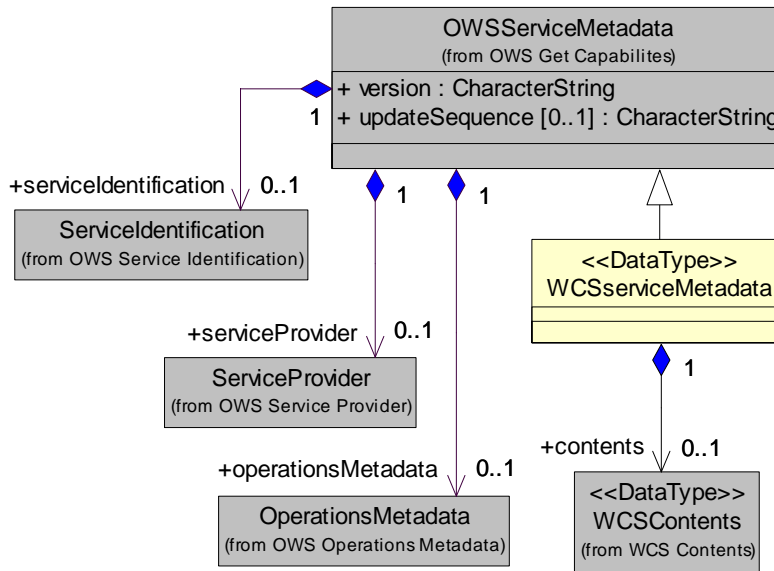


Figure 3 — WCS Service Metadata UML class diagram

NOTE To reduce the need for readers to refer to another document, the first three sections listed below are copied from Table 7 in Subclause 7.4.2 of [OGC 05-008].

Table 3 — Section name values and contents

Section name	Contents
ServiceIdentification	Metadata about this specific server. The schema of this section shall be slightly expanded from OWS Common, as specified in Subclause 8.3.2 below.
ServiceProvider	Metadata about the organization operating this server. The schema of this section shall be the same for all OWSs, as specified in Subclauses 7.4.4 and owsServiceProvider.xsd of [OGC 05-008].
OperationsMetadata	Metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The contents and organization of this section shall slightly modified from OWS Common, as specified in Subclause 8.3.3 below.
Contents	Metadata about the data served by this server. For the WCS, this section shall contain or reference brief descriptions of the coverages available from this server, as specified in Subclauses 8.3.5 and 8.3.6 below.

In addition to these sections, each service metadata document shall include the mandatory “version” parameter. Each service metadata document may include the optional “updateSequence” parameter specified in Table 6 in Subclause 7.4.1 of [OGC 05-008].

8.3.2 ServiceIdentification section contents

The ServiceIdentification section of a service metadata document contains basic metadata about this specific server. For WCS 1.1, the contents of this section shall be slightly expanded from the contents specified in Subclause 7.4.3 of OWS Common [OGC 05-008]. Specifically, the following row shall be added to Table 8 in Subclause 7.4.3.

Table 4 — Additional row in ServiceIdentification data structure

Name ^a	Definition	Data type and value	Multiplicity and use
Profile	Identifier of OWS Application Profile	URI Value specified by each Application Profile	Zero or more (optional) Include for each specified application profile implemented by server

The XML encoding of this parameter shall be added to the definition of the ServiceIdentification element, in owsServiceIdentification.xsd, using the following XML Schema fragment:

```
<element name="Profile" type="anyURI" minOccurs="0"
maxOccurs="unbounded">
  <annotation>
    <documentation>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. When this element is omitted, no meaning is implied.
</documentation>
  </annotation>
</element>
```

8.3.3 OperationsMetadata section contents

8.3.3.1 Overview

The OperationsMetadata section provides metadata about the operations specified by this service and implemented by this server, including the URLs for operation requests. The contents and organization of this section shall be slightly modified from OWS Common, as specified in Subclauses 7.4.5, 7.4.8, and owsOperationsMetadata.xsd of [OGC 05-008]. The change is replacing the ows:DomainType definition in owsOperationsMetadata.xsd with using the ows:DomainType definition in owsDomainType.xsd.

The mandatory values of various (XML) attributes shall be as specified in Table 5. In Table 5, the “Attribute name” column uses dot-separator notation to identify parts of a parent item. The “Attribute value” column references an operation parameter, in this case an operation name, and the meaning of including that value is listed in the right column.

Table 5 — Required values of OperationsMetadata section attributes

Attribute name	Attribute value	Meaning of attribute value
Operation.name	GetCapabilities	The GetCapabilities operation is implemented by this server.
	DescribeCoverage	The DescribeCoverage operation is implemented by this server.
	GetCoverage	The GetCoverage operation is implemented by this server.

In addition, there are many optional values of “name” attribute and “value” elements within optional Parameter and Constraint elements in the OperationsMetadata section, which may be included when considered useful. These Parameter and Constraint elements are for recording the domains of various parameters and quantities.

EXAMPLE 1 The domain of the “exceptionCode” parameter could record all the codes implemented for each operation by that specific server. Similarly, each of the GetCapabilities operation optional request parameters might have its domain recorded.

EXAMPLE 2 The domain of the “Sections” parameter in the GetCapabilities operation request could record all the sections implemented by that specific server.

8.3.3.2 Indicating allowed encodings

All WCS servers shall specify the encodings that may be sent using HTTP POST transfer of operation requests. Specifically, an ows:Constraint element shall be included, with “PostEncoding” as the value of the “name” attribute and specifying different allowed values for each allowed encoding:

- a) The value “SOAP” shall indicate that SOAP encoding is allowed, as specified in Annex F.
- b) The value “XML” shall indicate that XML encoding is allowed (without SOAP message encapsulation).

When the HTTP POST connect point URL is different for different encodings of the operation requests, this ows:Constraint element shall be included in each Post element. When

the connect point URL is the same for all encodings of all operation requests, this `ows:Constraint` element shall be included in the `OperationsMetadata` element.

8.3.3.3 Indicating support for the “store” parameter

All WCS servers shall specify whether or not that server implements the “store” parameter in `GetCoverage` operation requests, doing this in an `ows:Operation` element for the `GetCoverage` operation. When this WCS server implements the “store” parameter, this shall be specified by including an `ows:Parameter` element with “store” as the “name” attribute value, and both “True” and “False” as allowed Values. When this WCS server does not implement the “store” parameter, this shall be specified by including an `ows:Parameter` element with “store” as the “name” attribute value, and only “False” as an allowed Value.

8.3.4 Contents section

8.3.4.1 Overview

The Contents section shall contain and/or reference summary metadata about the coverages available. It shall be structured as shown in the UML class diagram in Figure 4. The attributes and associations of the two new classes shall include the data structure and parameters specified in Table 6. Table 7 lists the contents of the `CoverageSummary` data structure, allowing these summaries to be arranged in a hierarchy.

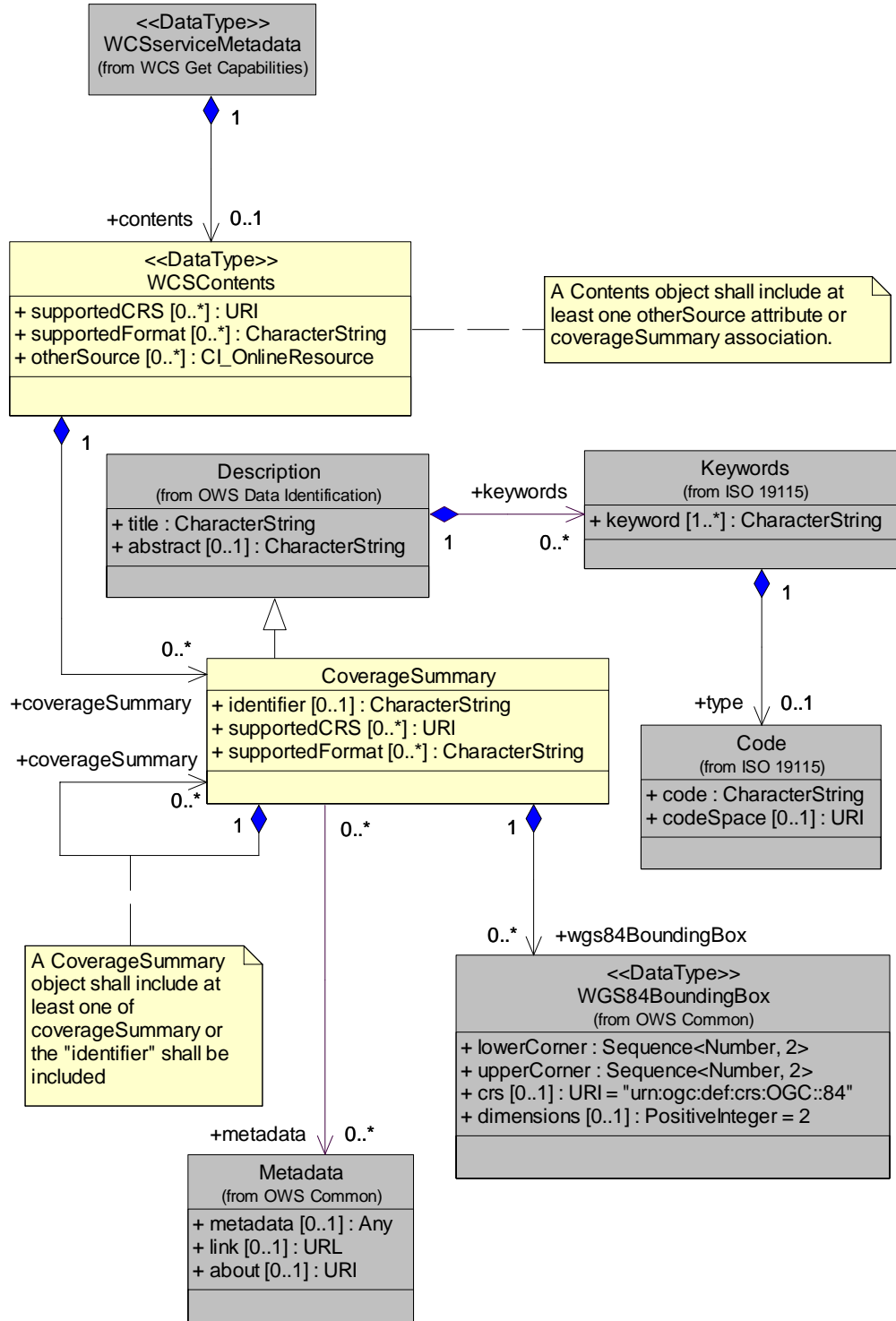


Figure 4 — Contents UML class diagram

Table 6 — Parts of Contents section

Name ^a	Definition	Data type	Multiplicity and use
coverageSummary CoverageSummary	Brief metadata describing a top-level coverage available from this WCS server	CoverageSummary data structure, see Table 7	Zero or more (optional) ^b One for each coverage available, unless OtherSource is used
supportedCRS SupportedCRS	Reference to CRS in which GetCoverage operation response may be expressed.	URI	Zero or more (optional) Include one for each CRS supported ^{c, d, f}
supportedFormat SupportedFormat	Identifier of format in which GetCoverage operation response may be encoded	CharacterString type	Zero or more (optional) Include one for each format supported ^{c, e, f}
otherSource OtherSource	Reference to another source of contents metadata	CI_OnlineResource class in ISO 19115	Zero or more (optional) Include when useful or no CoverageSummaries included
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b Shall be included unless the OtherSource is included and all this metadata is available from that source.</p> <p>c There may be a dependency of SupportedCRS on SupportedFormat, as described in Subclause 10.3.5.</p> <p>d This list of SupportedCRSs shall be the union of all of the supported CRSs in all of the nested CoverageSummaries.</p> <p>e This list of SupportedFormats shall be the union of all of the supported formats in all of the nested CoverageSummaries.</p> <p>f Servers should implement this list since it reduces the work clients need to do to determine that they can interoperate with the server.</p>			

NOTE The CoverageSummary and OtherSource parts of the Contents section are not mutually exclusive, although we expect that only one will often be included. This document specifies no meaning to inclusion of both parts, with or without overlapping information.

8.3.4.2 CoverageSummary

The CoverageSummary provides a summary-level description of coverage data available from a WCS server. Clients may be able to formulate simple GetCoverage requests based only on this information. However, in order to make more finely tuned GetCoverage requests, clients will usually need to obtain further details about a particular coverage, using the DescribeCoverage operation (see Clause 9). Table 7 lists the contents of the CoverageSummary data structure, allowing these summaries to be arranged in a hierarchy.

NOTE To reduce the need for readers to refer to another document, all except the CoverageSummary parameters listed below are largely copied from Table 23 in Subclause 10.6.1 of [OGC 05-008].

Table 7 — CoverageSummary data structure

Names ^a	Definition	Data type	Multiplicity and use
(Description) ^b	Description of a coverage	Description data structure, see [OGC 05-008]	One (mandatory) ^b
metadata Metadata	More metadata about this coverage	Metadata data structure, see [OGC 05-008]	Zero or more (optional) Include when useful
wgs84BoundingBox WGS84BoundingBox	Minimum bounding rectangle surrounding coverage, using WGS 84 CRS with decimal degrees and longitude before latitude	WGS84BoundingBox, see Subclause 10.2 in OGC 05-008	Zero or more (optional) Include when useful or needed ^d
supportedCRS SupportedCRS	Reference to CRS in which GetCoverage operation response may be expressed	URI	Zero or more (optional) ^{e, f} Include one for each CRS supported
supportedFormat SupportedFormat	Identifier of format in which GetCoverage operation response may be encoded	Character String type, not empty (ows:MimeType)	Zero or more (optional) ^{e, f} Include one for each format supported
identifier Identifier ^c	Unambiguous identifier of this coverage, unique for this WCS server.	Character String type, not empty	Zero or one (optional) Include when this coverage is available for DescribeCoverage and GetCoverage requests ^g
coverageSummary CoverageSummary	Brief metadata describing one subsidiary coverage available from this server	CoverageSummary data structure, see this table	Zero or more (optional) One for each subsidiary coverage, unordered ^f

a See Table 1 for UML and XML naming conventions.

b Contents inherited, not included as a named data structure.

c Values are identifiers in Contents section of GetCapabilities operation response. When the Capabilities XML document does not have any CoverageSummary sections, clients must obtain a valid coverage identifier from another source, such as a catalog service.

d This WGS84BoundingBox may be approximate, but should be as precise as practical. When multiple WGS84 bounding boxes are included, this shall be interpreted as the union of the areas of these bounding boxes. Table 8 below specifies when a WGS84BoundingBox is needed. Currently, this WCS specification provides no way for a server to specify the precision of a bounding box.

e When included, this list, including all values inherited from parent coverages, shall be an exact copy of the equivalent parameters in the CoverageDescription. Each CoverageSummary shall list or inherit at least one supportedCRS and one supportedFormat.

f There may be a dependency of SupportedCRS on SupportedFormat as described in Subclause 10.3.5.

g This Identifier shall be included unless one or more CoverageSummaries are included. When one or more CoverageSummaries are included, inclusion of the Identifier is optional.

8.3.4.3 Reserved characters in CoverageSummary parameter values

The following CoverageSummary parameter values shall be usable literally (“as-is”) in DescribeCoverage or GetCoverage operation requests to the originating server, encoded in KVP or XML:

- a) Identifier

b) SupportedFormat

In stating values for these parameters, servers shall avoid, encode, or escape certain special characters, as detailed in Subclause 7.5, to avoid misinterpreting them as part of the XML or KVP syntax of subsequent requests.

8.3.4.4 CoverageSummary hierarchy

The CoverageSummary data structures may form a hierarchical tree. When the CoverageSummaries form a tree, a subsidiary (child) CoverageSummary inherits some of the (optional) parameters from its higher level (parent) CoverageSummaries in the hierarchy. In this subclause, a “leaf” CoverageSummary is defined as one that has no subsidiary (child) CoverageSummary. Table 8 specifies the inheritance rules for the inherited parameters.

Table 8 — Inheritance of parts of CoverageSummary data structure

Name	Definition	Inheritance by subsidiary coverages
Description	Description of a coverage	Not inherited ^{a, b}
Identifier	Unambiguous identifier of this coverage, unique for this WCS server	Not inherited ^{a, c}
Metadata	Reference to more metadata about this coverage	Not inherited ^{a, c}
WGS84-BoundingBox	Minimum bounding rectangle surrounding coverage, using WGS 84 CRS with decimal degrees and longitude before latitude	Inherited when not provided ^{d, e}
SupportedCRS	CRS in which GetCoverage operation response may be expressed	Inherited and possibly added to ^{f, h}
Supported Format	Format in which GetCoverage operation response may be encoded	Inherited and possibly added to ^{g, h}
Coverage Summary	Metadata describing one subsidiary coverage available from this server	Not inherited

a Although these parameters are not inherited by a subsidiary coverage, the values of these parameters for all higher levels in a hierarchy of coverages may be relevant and are available to clients.

b A value for this mandatory parameter shall be provided at all levels in the hierarchy.

c Values for these optional parameters may be provided at all levels in the hierarchy.

d When no value is provided for a coverage, any value recorded for a higher level in a hierarchy of coverages shall apply to this coverage. When a value is provided for a coverage, it overrides any value recorded for a higher level in a hierarchy of coverages.

e For each “leaf” CoverageSummary, at least one WGS84BoundingBox shall be either recorded or inherited, for each georectified or georeferenced coverage. (This WGS84BoundingBox is included to simplify searching for coverages that might overlap a specified region.)

f The full list of CRSs supported by a coverage shall be the union of the CoverageSummary’s own supportedCRSs and those supported by its parent CoverageSummaries. These SupportedCRSs can be referenced as the baseCRS of a GridCRS specified in the Output part of a GetCoverage request. In addition, the GetCoverage operation response can be expressed in the ImageCRS of each stored coverage, which shall not be referenced here. These SupportedCRSs can also be referenced in the BoundingBox in the DomainSubset part of a GetCoverage request.

g The full list of formats supported by a coverage shall be the union of the CoverageSummary’s own supportedFormats and those supported by its parent CoverageSummary.

h There can be a dependency of SupportedCRS values on SupportedFormat values, as described in Subclause 10.3.5.

NOTE 1 This specification supports a CoverageSummary hierarchy only as a means to simplify and shorten the Capabilities document. No other possible meanings and uses of a CoverageSummary hierarchy are specified here.

A “leaf” CoverageSummary shall have an Identifier parameter; a non-leaf CoverageSummary may have an Identifier. A WCS server that supplies a CoverageSummary containing an Identifier element shall accept that Identifier value in DescribeCoverage and GetCoverage operation requests. A client shall not issue a DescribeCoverage or GetCoverage request for a coverage whose CoverageSummary has no Identifier parameter.

NOTE 2 This specification does not organize CoverageDescription data structures into a hierarchy, and the CoverageSummary hierarchy is not encoded in the CoverageDescription. A CoverageDescription provides all of the information necessary to request a coverage using the GetCoverage operation, and to correctly interpret the results from that GetCoverage operation. A coverage with a non-leaf CoverageSummary, when it has an Identifier, is described via DescribeCoverage, and requested via GetCoverage, in the same way as one with a leaf CoverageSummary.

NOTE 3 This subclause implies that only coverages which satisfy all the requirements to be correctly called a grid coverage, as specified in ISO 19123, can have an Identifier. Such a coverage may be a “virtual” coverage, which is not stored but is assembled (in an unspecified way) by the server in response to a GetCoverage operation.

8.3.4.5 OtherSource

The optional OtherSource parameter references a catalogue server from which coverage metadata is available. This ability is expected to be used by servers with thousands or millions of coverages, for which searching a catalogue is more feasible than retrieving and then searching a very large Capabilities XML document. When no CoverageSummary items are included, and one or more catalogue servers are referenced, this set of catalogues shall contain current metadata summaries for all the coverages currently available from this WCS server, with the metadata for each such coverage referencing this WCS server.

NOTE There is no requirement that all the coverages available from a specific WCS be listed in the Capabilities document (so that a catalogue may harvest them). For example, a server may need to serve on the order of (~) 10,000,000 distinct coverages with ~10,000 new coverages per day! In that case, this OtherSource may point to one or more catalogues that may be searched. For each catalogued coverage, that catalogue shall list the WCS server(s) that offer(s) that coverage, and the coverage identifier(s) used.

Any catalogue servers that are referenced should implement one of the interfaces specified in the OGC Catalogue Services Specification [OGC 04-021r3], preferably using the CSW protocol binding specified in Clause 10 of [OGC 04-021r3]. For any OGC catalogue protocol binding, an Application Profile for cataloging WCS coverages should be formally specified. Similarly, when no CoverageSummary items are included, a WCS Application Profile should be formally specified.

8.3.4.6 Implementation requirements

The “Multiplicity and use” columns in Table 6 through Table 16 in [OGC 05-008], and in Table 6 and Table 7 of this document, specify the optionality of each listed parameter and data structure in the GetCapabilities operation response. All the “mandatory” parameters and data structures shall be implemented by all WCS servers, using a specified value(s).

Implementation by WCS servers of the “updateSequence” parameter defined in Table 6 of [OGC 05-008] is optional. As indicated in Table 2 of this document, the “updateSequence” parameter may be implemented by each WCS server, and then shall also be required in the operation response.

All other “optional” parameters and data structures, in the GetCapabilities operation response, should be implemented by all WCS servers using specified values, whenever and wherever each is considered useful metadata for that server.

8.3.4.7 XML encoding

The Contents section of the Capabilities document shall be XML encoded. The XML Schema Document that specifies the Contents section for the WCS is built upon the owsContents.xsd schema, is named wcsContents.xsd, and is attached.

8.3.5 Capabilities document XML encoding

The XML Schema fragment for a WCS service metadata document extends the ows:CapabilitiesBaseType in owsGetCapabilities.xsd from [OGC 05-008]:

```
<element name="Capabilities">
  <complexType>
    <complexContent>
      <extension base="ows:CapabilitiesBaseType">
        <sequence>
          <element ref="wcs:Contents" minOccurs="0"/>
        </sequence>
      </extension>
    </complexContent>
  </complexType>
</element>
```

8.3.6 GetCapabilities document SOAP encoding

WCS servers that implement SOAP version 1.2 transfer of GetCoverage operation requests shall also implement SOAP transfer of the corresponding GetCoverage operation responses. These operation responses shall be encoded as specified in Annex F, using the XML encoding referenced above.

8.3.7 Exceptions

When a WCS server encounters an error while performing a GetCapabilities operation, it shall return an exception report message as specified in Clause 8 of [OGC 05-008]. The allowed exception codes shall include those listed in Table 5 of Subclause 7.4.1 of [OGC 05-008] when the updateSequence parameter is implemented by the server.

8.4 GetCapabilities operation examples

To request a Capabilities document, a client could issue the following GetCapabilities minimal operation request encoded as KVP:

```
http://hostname:port/path?
service=WCS&request=GetCapabilities
```

The corresponding minimum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
```

```

xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1/wcsGetCapabilities.xsd"
service="WCS"/>

```

In response to such a GetCapabilities request, the WCS server might generate a document that looks like the examples given in Subclause C.2: a top-level Capabilities element with all four child elements present – ServiceIdentification, ServiceProvider, OperationsMetadata, and Contents.

Next is a more detailed GetCapabilities operation request, requesting just the Contents section – first in KVP encoding:

```

http://hostname:port/path?
Service=WCS&request=GetCapabilities&
updateSequence=XYZ123&
AcceptVersions=1.1.0,1.0.0&
Sections=Contents&
AcceptFormats=text/xml

```

And in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:owcs="http://www.opengis.net/wcs/1.1/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1/wcsGetCapabilities.xsd"
service="WCS" updateSequence="XYZ123">
  <owcs:AcceptVersions>
    <owcs:Version>1.1.0</owcs:Version>
    <owcs:Version>1.0.0</owcs:Version>
  </owcs:AcceptVersions>
  <owcs:Sections>
    <owcs:Section>Contents</owcs:Section>
  </owcs:Sections>
  <owcs:AcceptFormats>
    <owcs:OutputFormat>text/xml</owcs:OutputFormat>
  </owcs:AcceptFormats>
</GetCapabilities>

```

The response to such a request would be an XML file with a top-level Capabilities element – as illustrated in Subclause C.2, but with ONLY the Contents section (omitting ServiceIdentification, ServiceProvider, and OperationsMetadata).

9 DescribeCoverage operation (mandatory)

9.1 Introduction

Once a client has obtained summary descriptions of a WCS server's available coverages, it may be able to make simple GetCoverage requests. However, in most cases the client will need to issue a DescribeCoverage request to obtain a full description of one or more coverages available. The server responds to such a request with an XML document describing one or more coverages served by that WCS. Each WCS server shall implement the DescribeCoverage operation.

9.2 DescribeCoverage operation request

9.2.1 DescribeCoverage request parameters

A DescribeCoverage operation request lists one or more coverages to be described, each identified by an identifier parameter. A request to perform the DescribeCoverage operation shall use the data structure shown in the UML diagram in Figure 5. The attributes of the new class shall include the parameters listed and defined in Table 9.

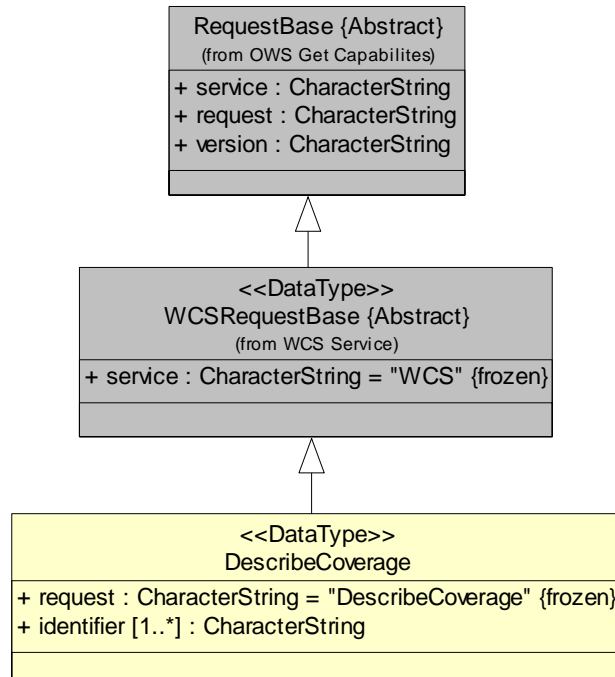


Figure 5 — DescribeCoverage operation request UML class diagram

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008]. The Identifier parameter listed below is largely copied from Table 7 above.

Table 9 — Parameters in DescribeCoverage operation request

Names ^a	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCS”	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely “DescribeCoverage”	One (mandatory)
version version	Specification version for operation	Character String type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
identifier Identifier	Unambiguous identifier of a coverage, unique for this WCS	Character String type, not empty ^b	One or more (mandatory) Include for each desired coverage
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b Values are identifiers in the Contents section of the GetCapabilities operation response. When the Capabilities XML document does not have a Contents section, clients must obtain a valid coverage identifier from another source, such as from a catalog service. See Subclause 8.3.4.1.</p>			

The “Multiplicity and use” column in Table 9 specifies the optionality of each listed parameter in the DescribeCoverage operation request. All parameters in the operation request are mandatory. That is, WCS clients shall implement all parameters, using the specified values, and WCS servers shall accept all parameters of the DescribeCoverage operation request, checking that each parameter contains the specified value.

9.2.2 DescribeCoverage request KVP encoding

WCS servers may implement KVP encoding for HTTP GET transfer of DescribeCoverage operation requests. Table 10 below specifies the DescribeCoverage request in KVP encoded form.

Table 10 — DescribeCoverage request URL encoding

URL component ^a	Description	Optionality
http://server_address/path/script?	URL of WCS server.	Required
service=WCS	Service name. Must be “WCS”.	Required
request=DescribeCoverage	Request name. Must be “DescribeCoverage”	Required
version=1.1.0	Request protocol version.	Required
identifiers=identifier1, identifier2, ...	A comma-separated list of coverage identifiers to describe (identified by their identifier values in the Capabilities document ^b)	Required
<p>a All parameter names are here listed using mostly lower case letters. However, see Table 1 for UML and XML naming conventions..</p> <p>b See footnote b from Table 9.</p>		

9.2.3 DescribeCoverage request XML encoding

WCS servers may implement XML encoding for HTTP POST transfer of DescribeCoverage operation requests. The schema fragment that specifies the contents and structure of a

DescribeCoverage operation request encoded in XML shall be as specified in the early part of the attached wcsDescribeCoverage.xsd file

9.2.4 DescribeCoverage request SOAP encoding

WCS servers may implement SOAP version 1.2 transfer of the DescribeCoverage operation request as specified in Annex F, using the XML encoding referenced above.

9.3 DescribeCoverage operation response

9.3.1 Normal response parameters

9.3.1.1 Coverage descriptions

The normal response to a valid DescribeCoverage operation request shall describe each coverage requested. More precisely, the response from a DescribeCoverage operation shall use the CoverageDescriptions data structure shown in the UML diagram in Figure 6, which uses the Domain and Range data structures shown in the UML diagrams in Figure 7 through Figure 9. The attributes and associations of the new classes shall include the parameters and data structures listed and defined in Table 11 through Table 21.

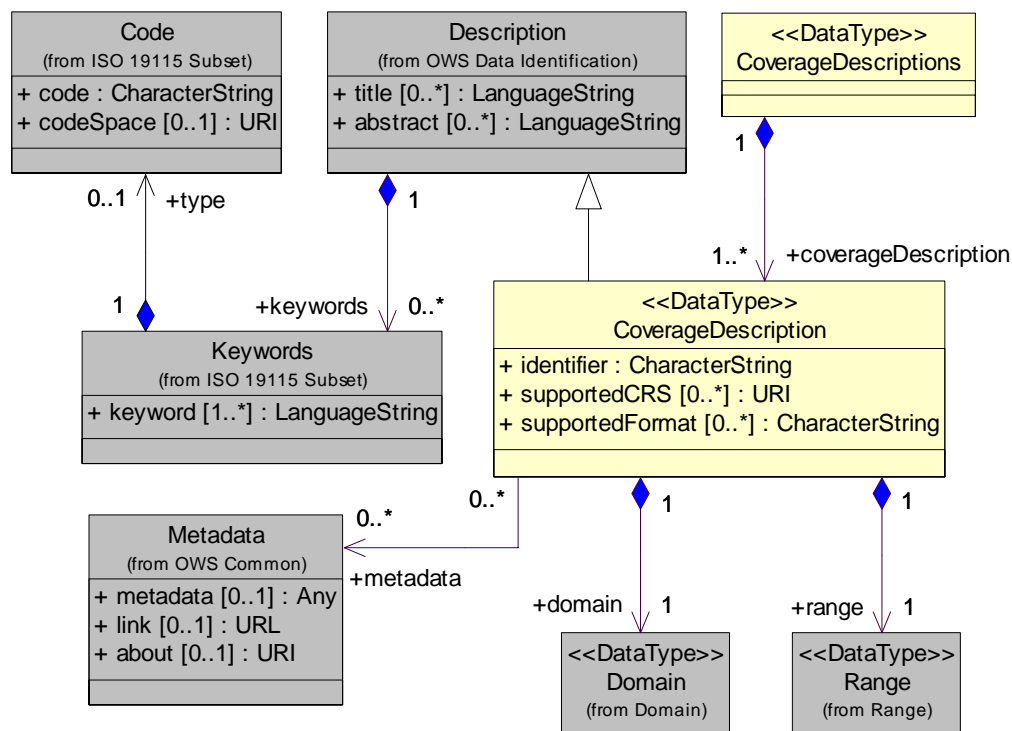


Figure 6 — DescribeCoverage operation response UML class diagram

Table 11 — Parts of CoverageDescriptions data structure

Names ^a	Definition	Data type	Multiplicity
coverageDescription CoverageDescription	Full description of one coverage available from WCS server	CoverageDescription data structure, see Table 12	One or more (mandatory)
a See Table 1 for UML and XML naming conventions.			

NOTE 1 To reduce the need for readers to refer to parts of this document, the second parameter listed below is largely copied from Table 7 of this document.

Table 12 — Parts of CoverageDescription data structure

Names ^a	Definition	Data type	Multiplicity and use
(Description) ^b	Description of a coverage	Description data structure, see [OGC 05-008]	One (mandatory) ^b
identifier Identifier	Unambiguous identifier of this coverage, unique for this WCS server	Character String type, not empty	One (mandatory)
metadata Metadata	More metadata about this coverage	Metadata data structure, see [OGC 05-008]	Zero or more (optional) Include when useful
domain Domain	Available coverage locations in space and/or time	Domain data structure, see Table 13	One (mandatory)
range Range	Available coverage values	Range data structure, see Table 18	One (mandatory)
supportedCRS SupportedCRS	Reference to CRS in which GetCoverage operation response can be expressed ^c	URI	One or more (mandatory) Include one for each CRS supported ^d
supportedFormat SupportedFormat	Identifier of format in which GetCoverage operation response can be encoded	CharacterString type, not empty	One or more (mandatory) Include one for each format supported ^d
a See Table 1 for UML and XML naming conventions.			
b Contents inherited, not included as a named data structure. Normally includes only one Metadata item, but may include more than one when used here, especially for an unrectified but georeferenced image coverage.			
c These SupportedCRSs can be referenced as the baseCRS of a GridCRS specified in the Output part of a GetCoverage request. Alternately, the GetCoverage operation response can be expressed in the ImageCRS of the stored coverage. These SupportedCRSs can also be referenced in the BoundingBox in the DomainSubset part of a GetCoverage request.			
d There may be a dependency of a SupportedCRS upon a SupportedFormat: see Subclause 10.3.5.			

9.3.1.2 Domain of coverage

The contents of the Domain data structure shall be as specified in Figure 7 and Table 13 through Table 17.

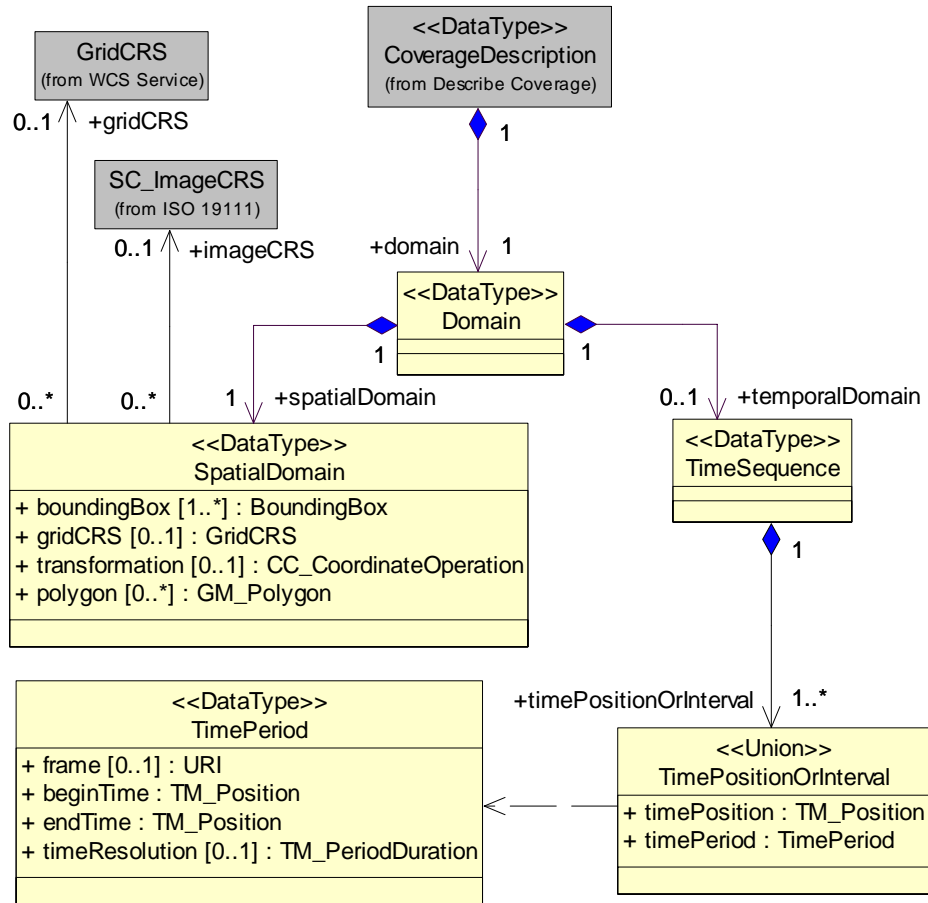


Figure 7 — Domain UML class diagram

Table 13 — Parts of Domain data structure

Names ^a	Definition	Data type	Multiplicity and use
spatialDomain SpatialDomain	Definition of spatial domain of this coverage	SpatialDomain data structure, see Table 14	One (mandatory)
temporalDomain TemporalDomain	Definition of temporal domain of this coverage	TimeSequence data structure, see Table 15	Zero or one (optional) Include when available ^b

a See Table 1 for UML and XML naming conventions.
b Although optional, the TemporalDomain should be included whenever a value is known or a useful estimate is available.

Table 14 — Parts of SpatialDomain data structure

Names ^a	Definition	Data type	Multiplicity and use
boundingBox BoundingBox	Unordered list of bounding boxes whose union covers spatial domain of this coverage ^b	BoundingBox data structure, see Subclause 10.2 in [OGC 05-008]	One or more (mandatory) Include for each bounding box useful for recording spatial domain
gridCRS GridCRS	Definition of GridCRS of this coverage	GridCRS data structure, see Annex G	Zero or one (optional) Include when coverage is georectified and thus has GridCRS
transformation Transformation	Georeferencing coordinate transformation for unrectified coverage	CC_CoordinateOperation, see ISO 19111 ^c	Zero or one (optional) Include when available ^d
imageCRS imageCRS	Association to ImageCRS of this coverage	Association to ImageCRS	Zero or one (optional) Include when coverage is an image
polygon Polygon	Unordered list of polygons whose union covers spatial domain of this coverage ^e	GM_Polygon, see ISO 19107	Zero or more (optional) Include one for each polygon needed
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b One bounding box could simply duplicate the information in the ows:WGS84BoundingBox; but the intent is to describe the spatial domain in more detail (e.g., in several different CRSs, or several rectangular areas instead of one overall bounding box). This bounding box shall use the ImageCRS of the stored coverage when that coverage is not georeferenced and not georectified. WCS use of this BoundingBox is further specified in specification Subclause 7.7.</p> <p>c This CoordinateOperation can be a Transformation or a ConcatenatedOperation that includes at least one Transformation.</p> <p>d To support use cases 4, 5, 9, or 10 specified in Annex H, a WCS server needs to use a georeferencing coordinate transformation for a coverage that is georeferenced but not georectified. However, a WCS server may opt not to support those use cases, or to use other means (such as a linked metadata file) to convey the georeferencing transformation, or not to make that transformation available to clients at all. Conversely, a client may opt to ignore any Transformation and to treat such a coverage as a non-georeferenced coverage (or to obtain the transformation by some other means).</p> <p>e Polygons are particularly useful for areas that are poorly approximated by an ows:BoundingBox (such as satellite image swaths, island groups, other non-convex areas).</p>			

Table 15 — Parts of TimeSequence data structure

Names ^a	Definition	Data type	Multiplicity
timePositionOrInterval TimePositionOrInterval	Ordered sequence of time positions or intervals ^b	TimePositionOrInterval type, see Table 16	One or more (mandatory)
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b The time positions and periods should normally be ordered from the oldest to the newest.</p>			

Table 16 — Parts of TimePositionOrInterval data structure

Names ^a	Definition	Data type	Multiplicity
timePosition TimePosition	A specific time	TM_Position, see ISO 19108 ^{b,c}	Zero or one (optional) ^c
timePeriod TimePeriod	A period of time	TimePeriod type, see Table 17	Zero or one (optional) ^c

- a See Table 1 for UML and XML naming conventions.
- b Values shall be expressed in the ISO 8601:2000 syntax, summarized in Subclause 9.3.2.4.
- c Inclusion of the timePosition or TimePeriod is mutually exclusive.

Table 17 — Parts of TimePeriod data structure

Names ^a	Definition	Data type	Multiplicity
beginTime BeginTime	Start of this time period	TM_Position, see ISO 19108 ^b	One (mandatory)
endTime EndTime	End of this time period	TM_Position, see ISO 19108 ^b	One (mandatory)
timeResolution TimeResolution	Spacing between valid times within this time period	TM_PeriodDuration, see ISO 19108 ^b	Zero or one (optional) Include when describing periodic samples in time.
frame frame	Temporal reference system	URI	Zero or one (optional) Default: “#ISO-8601”

a See Table 1 for UML and XML naming conventions.

b Values shall be expressed in the ISO 8601:2000 syntax, summarized in Subclause 9.3.2.4.

NOTE 5 The TimePeriod data structure is a variation of the GML TimePeriod, which allows the beginning and end of a time-period to be expressed in short-form inline using the Begin and End Position values.

9.3.1.3 Range of coverage

The contents of the Range data structure shall be as specified in Figure 8, Figure 9, and Table 18 through Table 21.

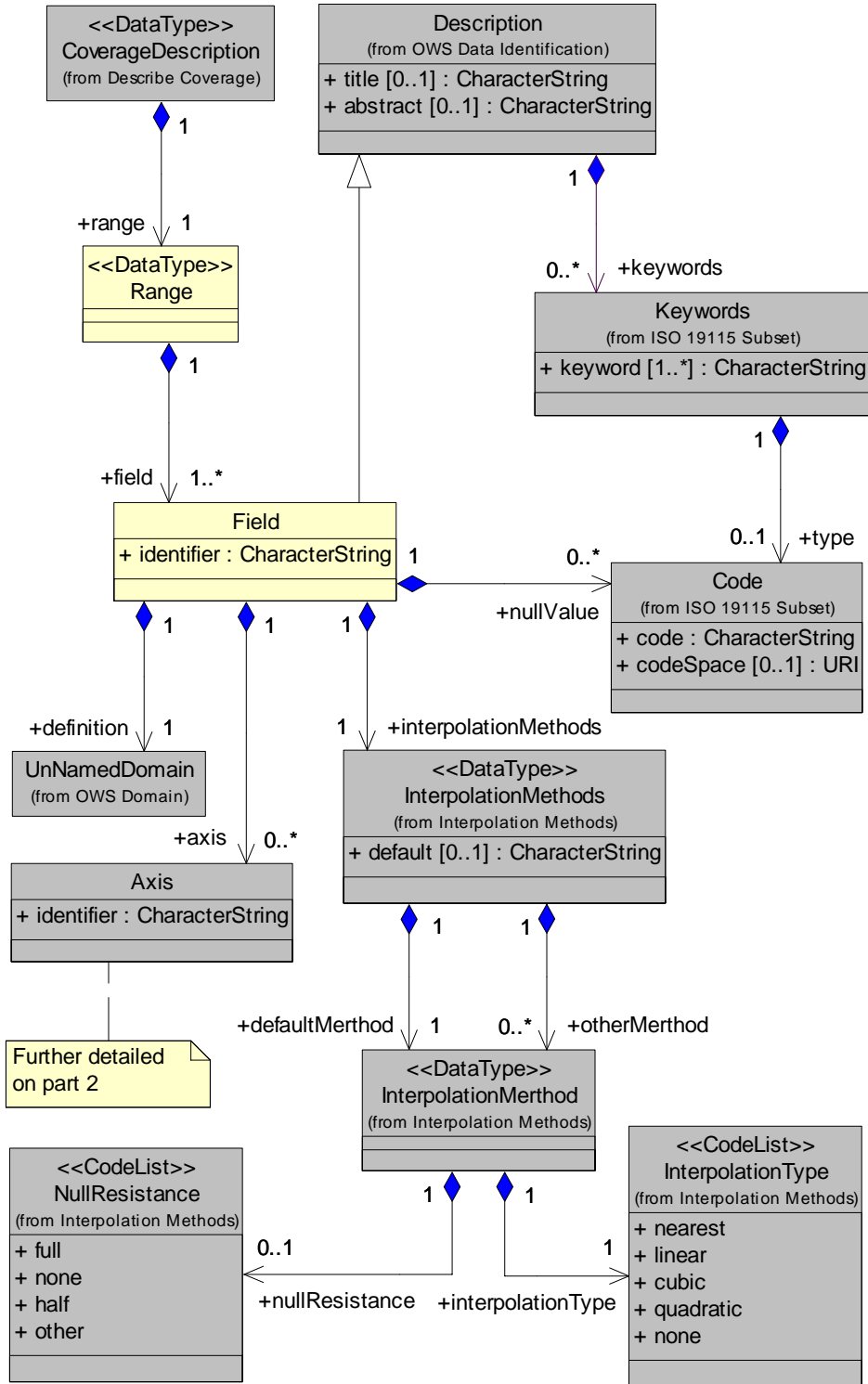


Figure 8 — Range UML class diagram, part 1

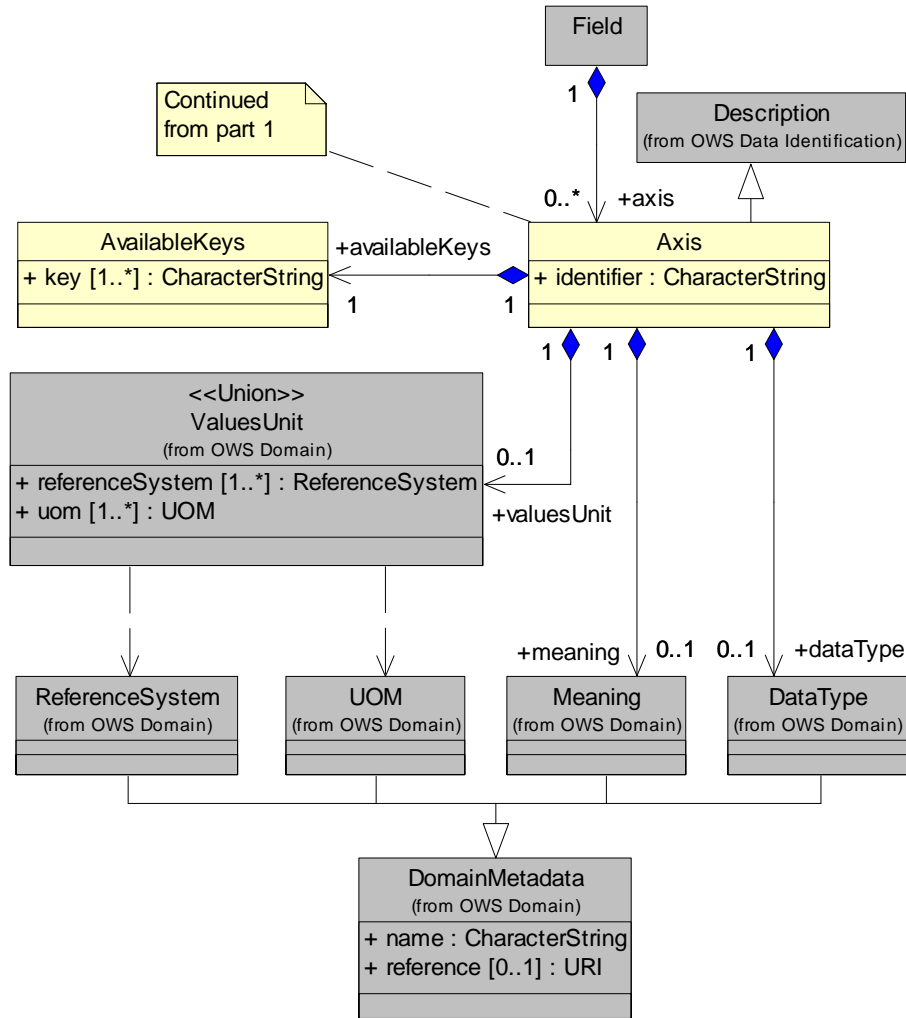


Figure 9 — Range UML class diagram, part 2

Table 18 — Parts of Range data structure

Names ^a	Definition	Data type	Multiplicity and use
field Field	Description of a field in range records, either scalar-valued or vector-valued	Field data structure, see Table 19	One or more (mandatory) One for each field in this Range record
<p>^a See Table 1 for UML and XML naming conventions.</p>			

Table 19 — Parts of Field data structure

Names ^a	Definition	Data type	Multiplicity and use
(Description) ^b	Description of this field	Description data structure, see [OGC 05-008]	One (mandatory) ^b
identifier Identifier	Identifier of this field, unique for this coverage	Character string type, not empty	One (mandatory)
definition Definition	Further definition of this field, including meaning, units, etc.	UnNamedDomain data structure, see Annex E ^e	One (mandatory)
nullValue NullValue	Value used when valid range values are not available ^c	Code data structure, see [OGC 05-008]	Zero or more (optional) Include when needed ^c
interpolationMethods InterpolationMethods	Spatial interpolation method(s) that server can apply to this field ^d	InterpolationMethods data structure, see Subclause I.4	One (mandatory)
axis Axis	Axis (“control variable”) of vector field for which there are range values	Axis data structure, see Table 20	Zero or one (optional) Include for each axis of a vector field (i.e., one that has axes)
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b Contents inherited, not included as a named data structure.</p> <p>c The coverage encoding itself may specify a fixed value for null (e.g. “-99999” or “N/A”), but often the choice is up to the provider and must be communicated to the client outside of the coverage itself. The optional codeSpace XML attribute can reference a definition of the reason why this value is null.</p> <p>d One of these interpolation methods shall be used when a GetCoverage operation request requires resampling. When the only interpolation method listed is ‘none’, clients may only retrieve coverages from this coverage in its native CRS at its native resolution.</p> <p>e In the Definition, the AllowedValues should be used to encode the extent of possible values for this field or for all subfields, excluding the Null Value(s). If the extent of possible values is not known, AnyValue should be used.</p>			

Table 20 — Parts of Axis data structure

Names ^a	Definition	Data type	Multiplicity and use
(Description) ^b	Description of this field	Description data structure, see [OGC 05-008]	One (mandatory) ^b
Identifier identifier	Identifier of this axis, unique for this field	Character string type, not empty	One (mandatory)
availableKeys AvailableKeys	List of values of keys for this axis	AvailableKeys data structure, see Table 21	One (mandatory)
meaning Meaning	Reference to meaning or semantics of this value or set of values	DomainMetadata data structure, see Table E.7	Zero or one (optional) Include when useful
dataType DataType	Reference to the data type of this set of values	DomainMetadata data structure, see Table E.7	Zero or one (optional) Include when useful
valuesUnit ValuesUnit	Indicates that this quantity has units or reference system, and provides the value used ^c	ValuesUnit data structure, see Table E.4	Zero or one (optional) Include when values have units or reference system
metadata Metadata	Additional metadata about domain of this quantity	ows:Metadata, see Table 23 of OGC 05-008	Zero or more (optional) One for each such metadata object ^d

a See Table 1 for UML and XML naming conventions.

b Contents inherited, not included as a named data structure.

c Provides the identifier of the units or reference system used by the AllowedValues or ValuesListReference.

d These metadata objects can be included in any order. A list of the required and/or optional metadata objects for each quantity can be specified in a WCS Application profile.

Table 21 — Parts of AvailableKeys data structure

Names ^a	Definition	Data type	Multiplicity and use
key Key	Value for a key for this axis	Character string type, not empty ^b	One or more (mandatory) ^c One for each key value ^d

a See Table 1 for UML and XML naming conventions.

b Value shall be string encoding of any value of another data type.

c There will normally be more than one key value for an axis.

d For numeric keys, signed values shall be ordered from negative infinity to positive infinity.

9.3.1.4 Implementation requirements

The “Multiplicity and use” columns in Table 11 through Table 21 specify the optionality of each listed parameter and data structure in the DescribeCoverage operation response. All the “mandatory” parameters and data structures shall be implemented by all WCS servers, using a specified value(s).

All the “optional” parameters and data structures in the DescribeCoverage operation response should be implemented by all WCS servers using specified values, whenever and wherever each is considered useful metadata for that coverage.

9.3.2 Supporting information

9.3.2.1 Scalar and vector fields

The fields in a coverage range record may be scalar-valued or vector-valued. A scalar-valued field associates a single number (or other “atomic” value) to each location in the domain.

EXAMPLE 1 A scalar-valued field would be useful for terrain elevation or panchromatic (full-spectrum) brightness of a landscape.

A vector-valued field in a coverage range conveys multiple identically defined observations (measuring the same phenomenon and expressing it in same reference system) at each location, recorded for different values (“keys”) of one or more control variables (“axes”). Clients may request subsets of such a field by selecting a subset of these keys.

EXAMPLE 2 The range of a hyperspectral image coverage would usually have a single vector-valued field, representing brightness values at different points along the electromagnetic spectrum. The axis of such a field would be the set of spectral bands, or the wavelengths, for which a sensor has recorded brightness values. A client might retrieve the near-infrared portions of a hyperspectral image by requesting only certain keys on an axis named “wavelength.”

For a vector field, the **Axis** data structure describes the nature of each control variable, and indicates (as **Keys**) the constraint values available for selecting field subsets. The Key values shall be textual names, which can be numbers encoded as character strings. The axes of a field span a multi-dimensional array of values, as depicted in Figure 10 below.

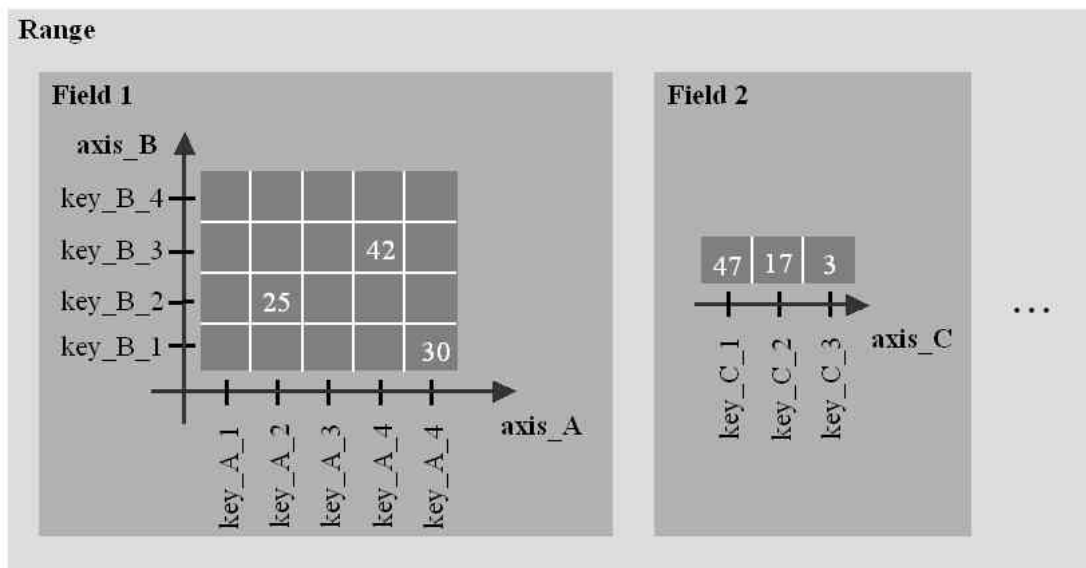


Figure 10 — Example vector fields

Figure 10 shows two example vector fields, one with two axes, and the other with one axis. These define rectangular arrays: that is, for all Keys along each Axis, the corresponding "slices" through the array have the same size.

NOTE 1 If a set of recorded observations has any semantic variation (including any differences in the measurement process or the reference system in which quantities are expressed), then the different observations belong in different fields or possibly different coverages. For instance, atmospheric pressure, water temperature, and land ownership would require distinct fields in a coverage range. Distinct fields would also be needed for a multispectral image in which some bands represent emissive radiance, and others reflective radiance.

NOTE 2 When a WCS service offers two sets of data for one area but with different spatial resolutions, then they need to be described as two separate coverages because they cannot share a common domain description.

9.3.2.2 SupportedFormat

The *required SupportedFormat* element(s) advertises an output format in which coverage may be requested from a WCS server. Formats are identified by a MIME type string. Any coverage format that can adequately convey the domain and range of the requested coverage is acceptable. The following formats have been used to convey certain kinds of coverages:

- GeoTIFF <<http://www.remotesensing.org/geotiff/geotiff.html>>
- HDF-EOS <<http://www.hdfeos.org>>
- NITF <<http://www.ismc.nga.mil/ntb/baseline/1999.html>>
- CF-NetCDF <<http://www.cgd.ucar.edu/cms/eaton/cf-metadata>>

NOTE This specification deprecates the privileged status given to some of the above formats in WCS 1.0. The above formats are cited merely as examples, with no normative intent.

However, for improved interoperability, clients and servers should use formats defined in WCS encoding profiles adopted by the OGC. These encoding profiles shall contain the following information:

- a) MIME type(s) and brief description: a concise overview of the encoding format, including the MIME type string(s) used to refer to it, the files required (e.g. header, dictionary, georeferencing, etc.), and the “role” values in the Xlink references in the GetCoverage response (see Subclauses 10.3.11.2 and I.3.2).
- b) Pointers to documentation for the encoding format. This documentation shall clarify how the encoding convention represents locations, times, and physical quantities represented in the dataset.
- c) Data model mapping to the Coverage Abstract Specification. This should include conventions for representing the spatio-temporal domain, and for representing the dimensions in the range. It should also describe limitations of the format for encoding complex coverages, and limitations of the coverage model for representing complex data structures encoded in the format.
- d) Examples: A set of examples of the encoding format, and of corresponding Coverages XML response documents (see Subclauses 10.3.11.2 and I.3).
- e) Pointers to implementing software for the encoding format. Providers of this software may license it in source code or executable form.
- f) Compliance Testing: Pointers to mechanisms for testing whether resulting WCS coverages conform to the encoding format.

The titles of these encoding profiles shall follow the following pattern:

WCS 1.1 Application Profile for [Format] [formatVersion] encoding,
[profileVersion]

EXAMPLE WCS 1.1 Application Profile for GeoTIFF 6.0 encoding, 0.9

9.3.2.3 Reserved characters in CoverageDescription parameter values

The following CoverageDescription parameter values shall be usable literally (“as-is”) in subsequent GetCoverage operation requests to the originating server, encoded in KVP or XML:

- a) “Identifier” parameter in CoverageDescription
- b) “SupportedFormat” parameter in CoverageDescription
- c) “Identifier” parameter in Field
- d) “InterpolationMethod” parameters in InterpolationMethods in Field
- e) “Identifier” parameter in Axis
- f) “Key” parameters in AvailableKeys in Axis

In stating values for these parameters, servers shall avoid, encode, or escape certain special characters, as detailed in Subclause 7.5, to avoid misinterpreting them as part of the XML or KVP syntax of subsequent requests.

In addition, when the WCS server supports KVP GetCoverage requests, the identifiers for Field, Axis, Key, and InterpolationMethod shall be usable literally within a RangeSubset parameter (see 10.2.2.2 below). This requires avoiding, encoding, or escaping any characters (‘[’ ‘]’ ‘:’ ‘;’ ‘,’) that are meaningful to the RangeSubset syntax. Servers may use any scheme for encoding or escaping these characters.

NOTE The method for encoding or escaping special characters for RangeSubset is not specified. This is because clients will present the resulting identifier strings only to the originating server.

9.3.2.4 Summary of ISO 8601 syntax for time positions and time periods

The [ISO 8601:2000] syntax for dates and times may be summarized by the following template (see Annex D of the OGC Web Map Service [OGC 04-024]):

ccyy-mm-ddThh:mm:ss.sssZ

Where

“ccyy-mm-dd” is the date (a four-digit year, and a two-digit month and day);

“T” is a separator between the data and time strings;

“hh:mm:ss.sss” is the time (a two-digit hour and minute, and fractional seconds);

“Z” represents the Coordinated Universal Time (UTC or “zulu”) time zone.

ISO 8601:2000 allows (i) up to a 14-digit year, with a negative sign, to denote the distant past; (ii) omitting less-significant numbers in the template to reduce the precision of the date

and time; and (iii) use of other time zones, denoted as in Subclause 5.3.4.1 of ISO 8601:2000. Here are a few examples:

EXAMPLE 1	2006	The year 2006
EXAMPLE 2	2006-09	September 2006
EXAMPLE 3	2006-09-27T10:00Z	10 o'clock (Universal Time) on 27 September, 2006

ISO 8601:2000 also provides a syntax for expressing time periods: the designator P, followed by a number of years Y, months M, days D, a time designator T, number of hours H, minutes M, and seconds S. Unneeded elements may be omitted. Here are a few examples:

EXAMPLE 4	P1Y	1 year
EXAMPLE 5	P1M10D	1 month plus 10 days
EXAMPLE 6	PT2H	2 hours

The WCS GetCoverage KVP syntax (defined in 10.2.2) extends ISO 8601:2000 with a syntax for time lists and ranges, as specified in Annex D of [OGC 04-024]. Some examples follow:

EXAMPLE 7	2000-06-23T20:07:48.11Z	(A single moment)
EXAMPLE 8	1999-01-01,1999-04-01,1999-07-01,1999-10-01	(A list of four dates)
EXAMPLE 9	1995-04-22T12:00Z/2000-06-21T12:00Z/P1D	(Daily ("P1D") between 12 noon on April 22 1995 and 12 noon on June 21, 2000)

9.3.3 Normal response XML encoding

The XML Schema fragment that specifies the contents and structure of a DescribeCoverage operation response shall be as specified in the latter part of the attached wcsDescribeCoverage.xsd file.

9.3.4 Normal response SOAP encoding

WCS servers that implement SOAP version 1.2 transfer of DescribeCoverage operation requests shall also implement SOAP transfer of the corresponding GetCoverage operation responses. These operation responses shall be encoded as specified in Annex F, using the XML encoding referenced above.

9.4 DescribeCoverage exceptions

When a WCS server encounters an error while performing a DescribeCoverage operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The allowed standard exception codes shall include those listed in Table 10. For each listed exceptionCode, the contents of the "locator" parameter value shall be as specified in the right column of Table 22.

NOTE To reduce the need for readers to refer to other documents, all four values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 22 — Exception codes for DescribeCoverage operation

exceptionCode value	Meaning of code	“locator” value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value ^a
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter

a When the Coverage parameter is invalid, operation request identifies a Coverage not offered by this server.

9.5 DescribeCoverage operation examples

The following example requests a description of a single coverage – first in KVP:

```
http://server_address/path/script?service=WCS
&request=DescribeCoverage&version=1.1.0
&identifiers=SPVIEW_530_274
```

And in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<DescribeCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1.0/DescribeCoverage.xsd"
service="WCS" version="1.1.0">
  <Identifier>SPVIEW_530_274</Identifier>
</DescribeCoverage>
```

In response to such a request, the WCS server might generate a document like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<CoverageDescriptions xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows"
xmlns:owcs="http://www.opengis.net/wcs/1.1/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
../wcsDescribeCoverage.xsd http://www.opengis.net/wcs/1.1/ows
../owsDataIdentification.xsd">
  <CoverageDescription>
    <ows:Title>Coverage File:
SPVIEW_530_274_0_020809_5_1_J_3.TIF</ows:Title>
    <ows:Abstract>Image 530_274_0_020809_5_1_J_3 Image, SAN FRANCISCO
UNITED STATES, Cnes 1986-2002, Distribution Spot Image</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>SPOT5</ows:Keyword>
      <ows:Keyword>HRG1</ows:Keyword>
      <ows:Keyword>SPOT</ows:Keyword>
    </ows:Keywords>
    <Identifier>SPVIEW_530_274</Identifier>
    <Domain>
      <SpatialDomain>
```

```

<ows:BoundingBox crs="urn:ogc:def:crs:EPSG::26910">
  <ows:LowerCorner>516430.13 4151612.83</ows:LowerCorner>
  <ows:UpperCorner>589060.13 4224332.83</ows:UpperCorner>
</ows:BoundingBox>
<GridCRS>
  <GridBaseCRS>urn:ogc:def:crs:EPSG::26910</GridBaseCRS>
  <GridType>urn:ogc:def:method:WCS:1.1:grid2dIn2dMethod</GridType>
  <GridOrigin>516430.13 4224332.83</GridOrigin>
  <GridOffsets>10.0 0 0 -10.0</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
</SpatialDomain>
<TemporalDomain>
  <gml:timePosition>2002-08-09T19:08:46Z</gml:timePosition>
</TemporalDomain>
</Domain>
<Range>
  <Field>
    <ows:Title>Sensor Wavelength</ows:Title>
    <ows:Abstract>Wavelength Selection, 4 Channels</ows:Abstract>
    <Identifier>Wavelength</Identifier>
    <Definition>
      <owcs:AnyValue/>
    </Definition>
    <NullValue>0.0</NullValue>
    <owcs:InterpolationMethods>
      <owcs:DefaultMethod>nearest</owcs:DefaultMethod>
      <owcs:OtherMethod>linear</owcs:OtherMethod>
    </owcs:InterpolationMethods>
    <Axis identifier="Band">
      <AvailableKeys>
        <Key>band1</Key>
        <Key>band2</Key>
        <Key>band3</Key>
        <Key>band4</Key>
      </AvailableKeys>
    </Axis>
  </Field>
</Range>
<SupportedCRS>urn:ogc:def:crs:EPSG:6.6:63266405</SupportedCRS>
<SupportedCRS>urn:ogc:def:crs:EPSG::26910</SupportedCRS>
<SupportedFormat>image/GeoTIFF</SupportedFormat>
<SupportedFormat>image/ECW</SupportedFormat>
<SupportedFormat>image/DTED</SupportedFormat>
<SupportedFormat>image/JPEG2000</SupportedFormat>
<SupportedFormat>image/NITF</SupportedFormat>
</CoverageDescription>
</CoverageDescriptions>

```

Clients may request descriptions of several coverages at once, in KVP:

```

http://server_address/path/script?service=WCS
&request=DescribeCoverage&version=1.1.0
&identifiers=Cov1,Cov2,Cov3

```

Or equivalently, in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<DescribeCoverage xmlns="http://www.opengis.net/wcs/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/1.1
  http://schemas.opengis.net/wcs/1.1.0/DescribeCoverage.xsd"

```

```
service="WCS" version="1.1.0">  
  <Identifier>Cov1</Identifier>  
  <Identifier>Cov2</Identifier>  
  <Identifier>Cov3</Identifier>  
</DescribeCoverage>
```

The corresponding response would have multiple CoverageDescription elements under the top-level CoverageDescriptions.

10 GetCoverage operation (mandatory)

10.1 Introduction

The mandatory GetCoverage operation allows retrieval of subsets of coverages. A WCS server processes a GetCoverage operation request and returns a response to the client that either contains or references the requested coverage(s). The GetCoverage operation shall be implemented by all WCS servers and clients.

10.2 GetCoverage operation request

10.2.1 GetCoverage request parameters

10.2.1.1 Overview

A request to perform the GetCoverage operation shall use the data structure shown in the UML diagram in Figure 11. The RangeSubset data structure is shown in the UML diagram in Figure 12. The attributes and associations of the new classes shall include the parameters and data structures listed and defined in Table 23 through Table 28.

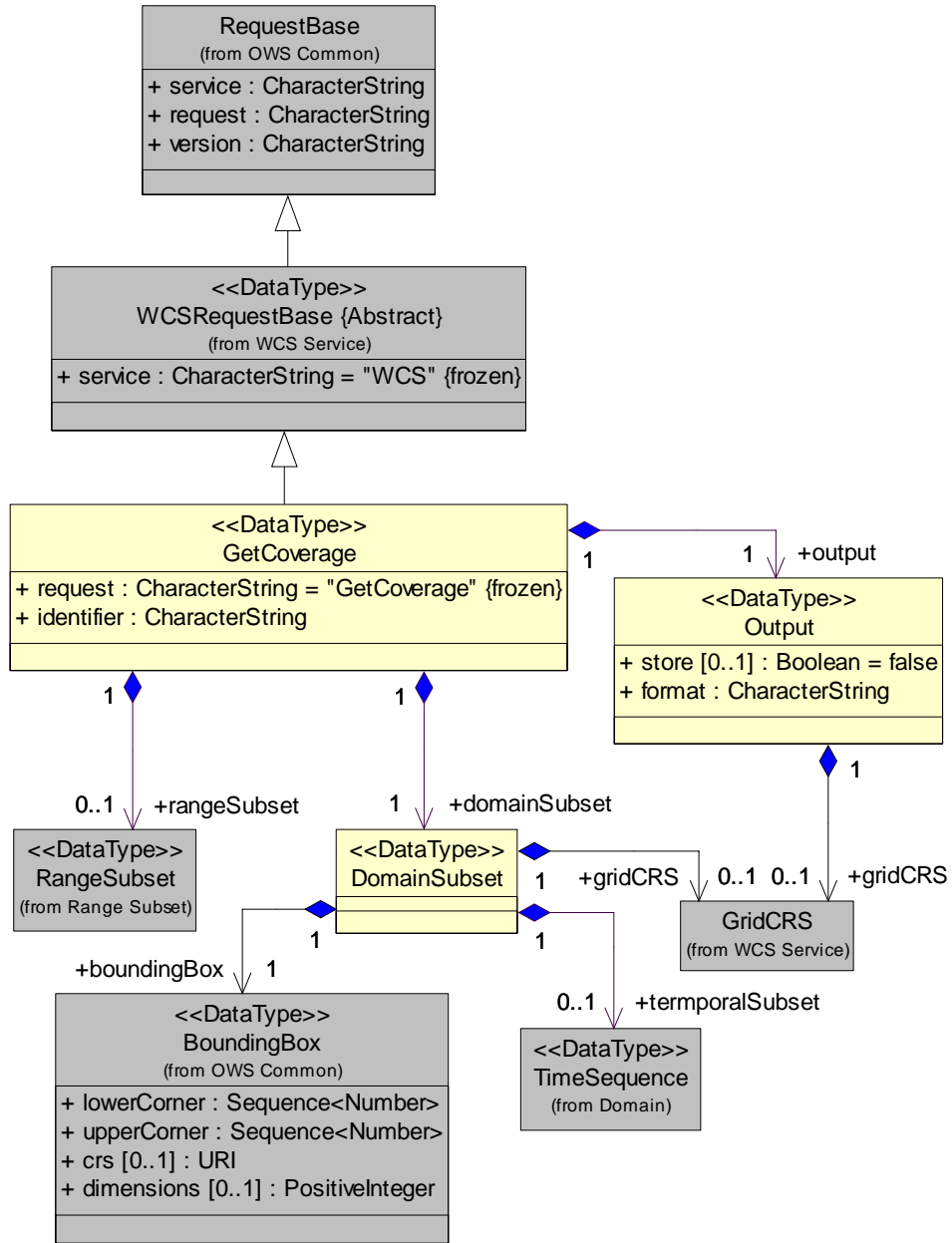


Figure 11 — GetCoverage request UML class diagram

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 21 in Subclause 9.2.1 of [OGC 05-008]. The fourth parameter listed below is largely copied from Table 7 above.

Table 23 — Parts of GetCoverage operation request

Names ^a	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	CharacterString type, not empty Value is OWS type abbreviation, namely “WCS”	One (mandatory)
request request	Operation name	CharacterString type, not empty Value is operation name, namely GetCoverage	One (mandatory)
version version	Specification version for operation	CharacterString type, not empty Value is specified by each Implementation Specification and Schemas version	One (mandatory)
identifier Identifier	Unambiguous identifier of coverage to draw from	CharacterString type, not empty ^b	One (mandatory)
domainSubset DomainSubset	Defines desired subset of domain set of identified coverage	DomainSubset data structure, see Table 24	One (mandatory)
rangeSubset RangeSubset	Defines desired subset of range(s) of identified coverage	RangeSubset data structure, see Table 27	Zero or one (optional) ^c
output Output	Specifies desired output options	Output data structure, see Table 25	One (mandatory)
<p>^a See Table 1 for UML and XML naming conventions.</p> <p>^b Values are identifiers in Contents section of GetCapabilities operation response. When this information is not in the Contents section of the Capabilities XML document, clients shall obtain a valid coverage identifier from another source, such as a catalog service.</p> <p>^c Providing a RangeSubset specifies a selection of range components and/or selects spatial interpolation methods for selected range components. Omitting a RangeSubset shall result in all range fields being delivered in their entirety, using the default spatial interpolation methods specified for each field.</p>			

Table 24 — Parts of DomainSubset data structure

Names ^a	Definition	Data type	Multiplicity and use
boundingBox BoundingBox	Desired spatial subset of coverage, in any CRS	BoundingBox data structure, see Subclause 10.2 of [OGC 05-008]	One (mandatory) Include when requesting spatial subset. ^b
gridCRS GridCRS	Definition of GridCRS referenced by bounding box above	GridCRS data structure, see Annex G	Zero or one (optional) Include when bounding box references a not-well-known GridCRS ^c
temporalSubset TemporalSubset	Desired temporal subset of coverage	TimeSequence data structure, see Table 15	Zero or one (optional) Include when requesting temporal subset. ^d
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b To request coverage values at all available spatial locations, use a BoundingBox supplied in the CoverageDescription for the chosen coverage. WCS use of this BoundingBox is further specified in Subclause 7.7.</p> <p>c This GridCRS definition may be omitted when it is the GridCRS of the stored georectified coverage. The baseCRS of this GridCRS can be any SupportedCRS, the stored coverage ImageCRS, or the stored coverage GridCRS listed in the CoverageDescription for this coverage.</p> <p>d Omit the TemporalSubset to request coverage values for all available times.</p>			

Table 25 — Parts of Output data structure

Names ^a	Definition	Data type	Multiplicity and use
gridCRS GridCRS	Definition of Coordinate Reference System in which GetCoverage operation response shall be expressed	GridCRS data structure, see Annex G	Zero or one (optional) Include unless desire ImageCRS ^d of stored coverage
format Format	Identifier of format in which GetCoverage operation response shall be encoded	CharacterString type ^b	One (mandatory)
store store	<p>”True” value asks server to store all components of the response coverage in a Web-accessible location, referenced by URL(s) in the GetCoverage response.</p> <p>”False” value asks server to return all components of the response coverage synchronously, bundled with the GetCoverage response.</p>	Boolean type	Zero or one (optional) Default is false ^c
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b The format identifier value shall be among those listed under supportedFormats in the DescribeCoverage XML response.</p> <p>c This parameter should be included only when this parameter is supported by this server, as encoded in the OperationsMetadata section of the Capabilities document.</p> <p>d When this GridCRS is not included, the output coverage shall use the gml:ImageCRS or wcs:GridCRS of the stored coverage, normally with the origin at a different grid point. When included, this wcs:GridCRS specifies the desired grid for the output coverage, including the desired grid spacings, one of many possible origin locations, optional rotation, and optional skewing (angle between the axes) in a base CRS. This wcs:GridCRS is NOT required or expected to specify the actual position of the origin in the GetCoverage operation output, so it shall not be interpreted as being different than the origin directly or indirectly specified by the BoundingBox in the DomainSubset.</p>			

10.2.1.2 RangeSubset

The contents of the RangeSubset data structure shall be as specified in Figure 7 and Table 26 through Table 28.

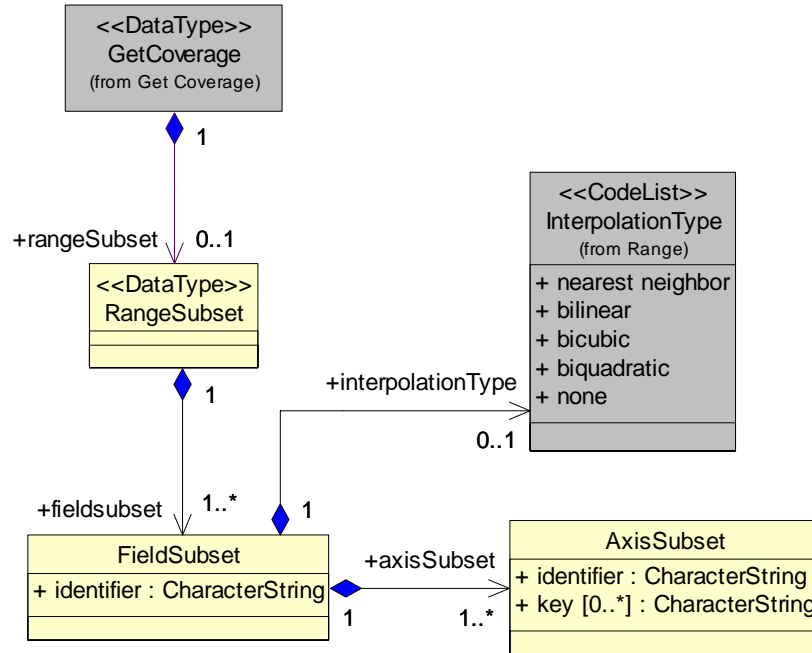


Figure 12 — RangeSubset UML class diagram

Table 26 — Parts of RangeSubset data structure

Names ^a	Definition	Data type	Multiplicity and use
fieldSubset FieldSubset	Requested field defined in range of requested coverage	FieldSubset data structure, see Table 27 below	One or more (mandatory) Include one for each requested field ^b

a See Table 1 for UML and XML naming conventions.

b To request all values in all fields, with their default spatial interpolations, omit the RangeSubset.

Table 27 — Parts of FieldSubset data structure

Names ^a	Definition	Data type and value	Multiplicity and use
identifier Identifier	Name of requested field	Character string, not empty ^b	One (mandatory)
interpolationType InterpolationType	Spatial interpolation method to be used to resample values in this field	InterpolationMethodType ^c	Zero or one (optional) Include to request interpolation method other than default for this field
axisSubset AxisSubset	Desired keys in one Axis for one (vector) field	AxisSubset data structure, see Table 28	Zero or more (optional) Include to request coverage values for only some of keys in an Axis defined for a (vector) field

a See Table 1 for UML and XML naming conventions.

b This identifier shall be for a Field specified in the CoverageDescription, and shall be unique for this Coverage.

c This interpolation type shall be one that is identified for this Field in the CoverageDescription. When this parameter is omitted, the interpolation type used shall be the default interpolation method specified for this Field, if any.

Table 28 — Parts of AxisSubset data structure

Names ^a	Definition	Data type and value	Multiplicity and use
identifier Identifier	Name of requested Axis	Character string, not empty ^b	One (mandatory)
key Key	Desired key in this Axis	Character string, not empty ^c	One or more (mandatory) ^d Include to request coverage values for only some of the keys in this Axis
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b Axis identifier shall be an Axis listed for the requested coverage field in the DescribeCoverage operation response.</p> <p>c Keys shall be ones listed for this Axis of this Field in the CoverageDescription.</p> <p>d To request field values for all keys on an Axis, omit the Axis from the FieldSubset.</p>			

The range subset of the output coverage shall be a set of fields taken from the range fields of the requested coverage. Each range field shall occur at most once in the RangeSubset list of Fields. Each Field data structure selects a field from the original coverage, optionally selects components of this field, and optionally chooses an interpolation method to be used for this field (and all of its components, if any).

The InterpolationType parameter selects, for each range field, the interpolation method to be applied (if needed) in resampling operations. The InterpolationType value shall be one of those listed for the requested coverage Field in the DescribeCoverage operation response. One and the same interpolation method shall be used by a server for all values in a given (scalar or vector) Field. However, different Fields may employ different interpolation methods. When the GetCoverage request specifies no interpolation method for a Field, then the server shall apply the default interpolation method listed in the coverage's CoverageDescription.

For a vector field, each AxisSubset data structure shall consist of a nonempty subset of the Keys defined for this Axis. Each AxisSubset selects subsets of this field constrained by the given set of Keys. When an Axis Subset is absent, the Field Subset shall be unconstrained along that Axis, with its values delivered in the Axis and Key sequence given in the CoverageDescription. Thus, when no AxisSubset is present for a vector field, the entire field shall be delivered.

NOTE 2 Subclause 10.3.6 details the effect of the RangeSubset parameter on the result coverage.

10.2.1.3 Implementation requirements

The “Multiplicity and use” columns in Table 23 through Table 28 specify the optionality of each listed parameter and data structure in the GetCoverage operation request. All the “mandatory” parameters and data structures shall be implemented by all WCS clients, using a specified value(s). Similarly, all the “mandatory” parameters and data structures shall be implemented by all WCS servers, checking that each request parameter or data structure is received with any specified value(s).

Implementation of the “store” parameter in the GetCoverage operation request is optional for both clients and servers. When a WCS server that does not implement the “store” parameter receives the “store” parameter with the “False” value, it shall not send an exception report message and shall not store the result coverage. When a WCS server that does not implement

the “store” parameter receives the “store” parameter with the “True” value, it shall return an exception report message with the InvalidParameterValue value of the exceptionCode and “store” as the value of the “locator”. Whether or not a WCS server implements the “store” parameter shall be specified in the OperationsMetadata section of the Capabilities document, as specified in Subclause 8.3.3 above.

All WCS servers shall, and all WCS clients should, implement the other “optional” parameters and data structures in the GetCoverage operation request, where appropriate.

10.2.2 GetCoverage request KVP encoding

10.2.2.1 Overview

WCS servers may use KVP encoding for HTTP GET transfer of GetCoverage operation requests. The KVP encoding of the GetCoverage operation request shall be as specified in Table 29. The parameters listed in Table 29 shall be as specified in Table 23 through Table 28 above.

Table 29 — GetCoverage request expressed as Key Value Pairs

Name and example ^a	Optionality & use	Definition and format
http://server_address/path/script?	Mandatory	URL of WCS server GetCoverage operation
service=WCS	Mandatory	Service name, shall be “WCS”
version=1.1.0	Mandatory	Request protocol version
request=GetCoverage	Mandatory	Name of the operation, shall be “GetCoverage”
identifier=identifier	Mandatory	Unique identifier of an available coverage
BoundingBox=47,-71, - 51,66, urn:ogc:def:crs:EPSG:6.6:6326 6405	Optional, include when spatial subset desired ^b	Request a coverage subset defined by the specified bounding box in the referenced coordinate reference system
TimeSequence= 20060801, 20060811, ... OR TimeSequence = 20060801/ 2006-0901 / PID, ...	Optional, include when temporal subset desired other than default ^b	Request a subset corresponding to the specified time instants or intervals, expressed in the extended ISO 8601 syntax defined in Annex D of [OGC 04-024]. (See 9.3.2.4.)
RangeSubset=temp:nearest; radiance[band[1,2,5]] (see syntax below)	Optional; include when requesting values for only some of the available fields or field keys; or non-default interpolation types on fields.	Request only some fields OR subsets of some fields (OR list fields to request non- default interpolation on them). (see syntax below)
format=image/netcdf	Mandatory	Requested output format of Coverage. Shall be one of those listed in the description of the selected coverage
store=true	Optional, include when supported and not false	Specifies whether response coverage should be stored, remotely from client at network URL, a boolean value
GridBaseCRS=urn:ogc:def:crs:EP SG:6.6:32618	Optional, include all of five GridCRS parameters, except for defaulted parameters, to request output in a CRS other than the ImageCRS of stored coverage	Reference to baseCRS of desired output GridCRS, a URN ^c
GridType=urn:ogc:def:method:W CS:1.1:2dGridIn2dCrS		Reference to grid type of desired output GridCRS, a URN ^{c,d}
GridCS=urn:ogc:def:cs:OGC:0.0: Grid2dSquareCS		Reference to coordinate system of desired output GridCRS, a URN ^{c,e}
GridOrigin=0,0		Position coordinates of one possible grid origin, in baseCRS of desired output GridCRS ^{c,f}
GridOffsets=0.0707, -0.0707,0.1414,0.1414&		Offsets between adjacent grid points, in baseCRS of desired output GridCRS ^{c,g}
<p>^a All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 05-008].</p> <p>^b At least one of BoundingBox and TimeSequence shall be included. WCS use of this BoundingBox is further specified in Subclause 7.7.</p> <p>^c The GridType parameter shall reference the known definition of a grid type for a GridCRS, as specified in Annex G. That GridType definition shall specify the meanings of the other “Grid...” parameters listed, including the number of dimensions in each other “Grid...” parameter.</p> <p>^d This GridType may be omitted when the GridBaseCRS is 2D, in which case the default value “urn:ogc:def:method:WCS:1.1:2dSimpleGrid” shall be used.</p> <p>^e This GridCS may be omitted when the GridBaseCRS is 2D, in which case the default value shall be “urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS”.</p> <p>^f This GridOrigin may be omitted when the GridBaseCRS is 2D, in which case the default value shall be “0.0 0.0”.</p> <p>^g When the GridType is “urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS”, there shall be four values in this GridOffsets. The center two of these offsets will be zero when the GridCRS is not rotated or skewed in the GridBaseCRS.</p>		

10.2.2.2 RangeSubset KVP syntax

The syntax of the KVP RangeSubset parameter shall conform to the following grammar (using EBNF (Extended Backus-Naur Form) notation [IETF RFC 2396]):

```

RangeSubset = FieldSubset *( ";" FieldSubset )
FieldSubset = FieldId [ ":" Interpolation ] [ "[" AxisSubsets "]" ]
AxisSubsets = AxisSubset *( "," AxisSubset )
AxisSubset = AxisId "[" Keys "]"
Keys = Key *( "," Key )
FieldId = <field identifier, as defined in Subclause 9.3.1.3 and Table 19>
AxisId = <axis identifier, as defined in Subclause 9.3.1.3 and Table 20>
Key = <key, as defined in Subclause 9.3.1.3 and Table 21>
Interpolation = <interpolation type identifier, as defined in Table I.7>

```

NOTE No extra whitespace is allowed in a RangeSubset string – any whitespace is always considered part of an adjacent identifier.

EXAMPLES

```

rangeSubset=radiance;temperature
rangeSubset=radiance:linear;temperature:nearest
rangeSubset=radiance:linear[wavelengths[L1,L2],forecastTimes[T1];
    temperature:nearest[forecastTimes[T1]]

```

When the WCS server supports KVP GetCoverage requests, the identifiers for FieldID, AxisID, Key, and InterpolationMethod given in the CoverageDescription shall be used literally (“as-is”) in the RangeSubset parameter, with no additional character encoding or escaping. (See 9.3.2.3 above.)

10.2.3 GetCoverage request XML encoding

WCS servers may implement XML encoding for HTTP Post transfer of GetCoverage operation requests. The XML Schema fragment for the GetCoverage operation request shall be as specified in the attached wcsGetCoverage.xsd file.

10.2.4 GetCoverage request SOAP encoding

WCS servers may implement SOAP version 1.2 transfer of the GetCoverage operation request as specified in Annex F, using the XML encoding referenced above.

10.3 GetCoverage operation response

10.3.1 Introduction

The normal response to a valid GetCoverage operation request shall be a single coverage extracted from the coverage requested, with the specified spatial reference system, bounding box, size, format, and range subset.

10.3.2 GetCoverage exceptions

When a WCS server encounters an error while performing a GetCoverage operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 05-008]. The

allowed standard exception codes shall include those listed in Table 30. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 30.

NOTE To reduce the need for readers to refer to other documents, the first three values listed below are copied from Table 20 in Subclause 8.3 of [OGC 05-008].

Table 30 — Exception codes for GetCoverage operation

exceptionCode value	Meaning of code	“locator” value
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value ^a
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter
UnsupportedCombination	Operation request contains an output CRS that can not be used within the output format.	Name of Format or name of CRS
NotEnoughStorage	Operation request specifies “store” the result, but not enough storage is available to do this.	None, omit “locator” parameter
^a The Coverage parameter is not valid when the operation request identifies a Coverage that is not offered by this server. The Format parameter is not valid when the operation request identifies a Format not offered for this coverage by this server.		

10.3.3 Coverage processing

The WCS GetCoverage request invokes some or all of a set of six basic operations:

- a) domain subsetting
- b) range subsetting
- c) resampling
- d) data format encoding
- e) result file delivery

Processing a GetCoverage request can conceptually be understood as executing the operation sequence shown in Figure 13. Depending on the particular request, some steps may be skipped. While a concrete implementation may choose a different evaluation strategy it shall ensure that the result is identical to the one achieved by the sequence depicted in Figure 13.

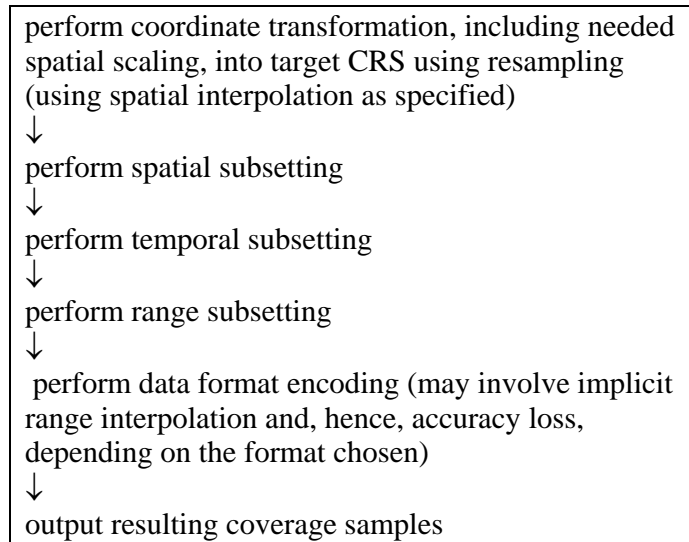


Figure 13 — Conceptual GetCoverage request evaluation sequence

10.3.4 Output coverage CRS

Each output coverage from a GetCoverage operation shall be in the GridCRS that is specified in the Output data structure within the GetCoverage operation request, when such a GridCRS is specified there. When no GridCRS is included, the output coverage shall use the ImageCRS or GridCRS of the stored coverage, normally with the origin at a different grid point. The stored coverage will use a GridCRS when it is georectified, or an ImageCRS when it is an image that has not been georectified. An image that has not been georectified may be georeferenced or not.

When included, this GridCRS specifies the desired grid for the output coverage, including the desired grid spacings, one of many possible origin locations, optional rotation, and optional skewing (angle between the axes) in a base CRS. However, this GridCRS is NOT required or expected to specify the actual position of the origin in the GetCoverage operation output. The position of the origin in the GetCoverage operation output shall be as directly or indirectly specified by the BoundingBox in the DomainSubset in the GetCoverage operation request.

10.3.5 SupportedFormat and SupportedCRS dependencies

Some formats require that the data be stored in a specific CRS, or the CRS must be chosen from a small list of CRSes. The dependency between CRSes and formats can not be expressed in this specification. When a WCS server receives a request where the requested output format and the requested output CRS are incompatible then the server shall return an UnsupportedCombination exception, see Table 30.

10.3.6 Output coverage spatial extent

The spatial extent of each output coverage shall be a rectangle in the CRS of that output coverage, using null values for any grid points whose values cannot be interpolated (when allowed) from grid point values included in the stored coverage. The BoundingBox of this spatial extent shall be computed, in the CRS of that output coverage, from the DomainSubset

BoundingBox in the GetCoverage operation request. The specified extent may be skewed with curved edges in the desired output coverage CRS, although it is rectangular in the CRS of the DomainSubset BoundingBox. This will usually occur when a GridCRS is defined as the output coverage CRS, and this GridCRS (not the baseCRS of this GridCRS) is different from the CRS used in the DomainSubset BoundingBox. Such a skewed and curved extent in the desired resampled output coverage shall be interpreted as the minimum extent desired.

NOTE 1 WCS use of BoundingBoxes is further specified in specification Subclause 7.7.

When the specified extent is skewed/curved in the desired output coverage CRS, the server shall extend this extent to the minimum rectangle surrounding that skewed/curved extent. And the server shall extend this minimum rectangle to the next grid point rows and columns in the output coverage GridCRS or ImageCRS.

NOTE 2 Computing the minimum surrounding rectangle may be done by considering a polygon with vertices at the four corner points of the specified bounding box and at least the center points in each side of that bounding box. The positions of these eight or more polygon vertices may be transformed from the CRS of the DomainSubset BoundingBox to the desired output coverage GridCRS (not the baseCRS of this GridCRS). The minimum rectangle surrounding all these polygon vertex positions can then be computed, followed by extending each side to the next row of column of grid point positions.

The transformed and squared-off extent will often be skewed with curved edges in the stored georectified coverage, although it is rectangular in the output coverage CRS. When the transformed skewed/curved extent does not lie wholly within the stored coverage, padding with null values shall be applied to all output grid points falling outside the stored coverage.

10.3.7 Output coverage temporal extent

The temporal extent of the output coverage shall be the time instance(s) or period(s) requested in the GetCoverage request. For a particular time instance or interval, when the server cannot supply coverage values, it shall substitute null values. When the GetCoverage request specifies no temporal subset, then the returned coverage shall return values for all times in the domain of the requested Coverage.

NOTE The WCS currently does not allow interpolation of coverage values along the time dimension.

10.3.8 Output coverage range

The range of a result coverage is controlled by the RangeSubset data structure. The output coverage range shall consist of all and only the fields specified in the RangeSubset of the GetCoverage request, ordered as they appear there. When the GetCoverage request has no RangeSubset data structure, the result coverage range shall contain all fields of the original coverage, in the sequence defined in the CoverageDescription, and with the default interpolation method(s) applied whenever interpolation is required.

In the case of a vector field, the output coverage field shall be limited to the values corresponding to keys in the AxisSubset structure in the GetCoverage FieldSubset. The absence of an AxisSubset on a FieldSubset shall be equivalent to requesting field values for all keys of that Axis. (Thus, the absence of any AxisSubset at all for a field shall be equivalent to requesting the field in its entirety.)

The `InterpolationType` parameter selects, for each range field, the interpolation method to be applied whenever spatial interpolation is used during resampling. A server shall use the same interpolation method for all values in a field. However, it may use different interpolation methods for different fields. When the `getCoverage` request does not specify an interpolation method for a field, then the server shall apply the default interpolation method listed for that field in the `CoverageDescription`.

EXAMPLE 1 Assume a coverage with two fields, *radiance* and *temperature*. *radiance* has two Axes, “`waveLength`” with three keys (“0.6”, “0.7”, and “0.9”) and “`forecastTime`” with four keys (“08:00”, “14:00”, “20:00”, “02:00”). The field *temperature* has the Axis “`forecastTimes`”, with keys “06:00”, “12:00”, “18:00”, “00:00”. Both fields are assumed to support interpolation methods “linear” and “cubic”. Then, the following `RangeSubset` clause selects radiance at `waveLength` keys 0.6 and 0.9 and `forecastTime` keys 08:00 and 14:00, and temperature at all forecast times. When spatial interpolation is needed, it requests linear interpolation for radiance and the (server-defined) default interpolation method for temperature.

```
RangeSubset=radiance:linear[waveLength[0.6,0.9],forecastTime[08:00,14:00]];
temperature
```

10.3.9 Output coverage metadata

The metadata listed in Table 31, or the equivalent information, shall be included with each WCS output coverage. Such equivalent metadata may be encoded with the coverage in some of the possible coverage formats.

Table 31 — Required output coverage metadata

Names ^a	Definition	Data type	Multiplicity and use
boundingBox BoundingBox	Bounding box that specifies extent of output coverage	ows:BoundingBoxType in ^b CRS of output coverage	One (mandatory)
gridCRS GridCRS	Definition of GridCRS used by output coverage	wcs:GridCRSType	Zero or one (conditional) Include when GridCRS used by BoundingBox
transformation Transformation	Definition of georeferencing coordinate transformation	gml:Transformation or gml:ConcatenatedOperation	Zero or one (conditional) Include when output coverage is not georectified but is georeferenced
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b The Coordinate Reference System (CRS) of an output coverage may be either a GridCRS or an ImageCRS.</p>			

10.3.10 Output coverage formats

A WCS server shall serve coverages in any of the formats listed as a `SupportedFormat` for the requested coverage (see 9.3.2.2 above). For most possible formats, more information is required to specify how a WCS output coverage shall (or can) be encoded using that format. That specific information should be specified in a WCS Application Profile document (see 9.3.2.2 above).

NOTE Encoding may reduce the coverage’s range accuracy when the format chosen does not support the coverage’s accuracy in terms of range types. For example, GIF can only represent 256 color values, hence a GIF converter will approximate colors while determining a best-match color map. Encoding a result coverage in

JPEG (with quality less than 1.0) will incur range value approximation. In all of these cases the pixels' radiometry will be inaccurate.

Regardless of the coverage format, the GetCoverage response shall be encoded as a Coverages XML document that references one or more files, which may be bundled with it or accessible separately via URL(s), as detailed in 10.3.11 below.

10.3.11 Response encodings

10.3.11.1 Overview

The encoding of the GetCoverage response consists of a Coverages XML document (see 10.3.11.2 below), packaged and bundled according to the request encoding and the value of the “store” parameter (see Table 23). WCS servers shall implement response encodings according to Table 32.

Table 32 — GetCoverage response encodings

Request encoding	store=true ^a	store=false (<i>default</i>)
KVP or XML	Coverages alone (see 10.3.11.2)	Coverages in a MIME Multipart message (see 10.3.11.3)
SOAP	Coverages in a SOAP envelope (see 10.3.11.4)	Coverages in a SOAP envelope with Attachments (see 10.3.11.5)
a Applies only when the server supports the “store” parameter (see Subclause 8.3.3.3).		

If the “store” parameter has the value “true”, the server shall store the result file(s) at URL-addressable location(s) of its choosing, and return only the Coverages data structure (within a SOAP envelope in the case of a SOAP request) with references to the other files as indicated in 10.3.11.2 and 10.3.11.4 below.

When the “store” parameter is absent or has the value “false”, the server shall transfer the complete GetCoverage response to the client, either as a MIME multipart message (for KVP or XML requests) or as a SOAP message with attachments (for SOAP requests). The Coverages shall reference the other parts of the MIME multi-part message (or SOAP attachments) as indicated in 10.3.11.3 below.

10.3.11.2 Coverages data structure

The GetCoverage operation response shall be XML encoded using the Coverages data structure specified in Annex I.3. This Coverages data structure allows the response to reference multiple files, and shall be output even when the GetCoverage operation response would otherwise contain only one file.

As shown in the UML diagram in Figure I.2, this Coverages object contains one or more Coverage reference group objects, each of which includes one or more Reference elements. A Coverages class in a GetCoverage operation response shall reference all the data files for one output coverage, including the coverage values and any associated metadata.

EXAMPLE An example XML document using the Coverages element is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Coverages xmlns="http://www.opengis.net/wcs/1.1/ows"
  xmlns:ows="http://www.opengis.net/ows"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:schemaLocation="http://www.opengis.net/wcs/1.1/ows ../owsCoverages.xsd
http://www.opengis.net/ows ../../../../ows/1.0.0/ows19115subset.xsd">
  <Coverage>
    <ows:Title>TBD</ows:Title>
    <ows:Abstract>Coverage created from GetCoverage operation request to a
WCS</ows:Abstract>
    <Identifier>TBD</Identifier>
    <Reference xlink:href="http://my.server.com/coverage/image.tiff"
xlink:role="urn:ogc:def:role:WCS:1.1:coverage"/>
    <Reference xlink:href="http://my.server.com/coverage/metadata.xml"
xlink:role="urn:ogc:def:role:WCS:1.1:metadata"/>
  </Coverage>
</Coverages>
```

NOTE The URN values used for Reference “role” attributes are to be specified in encoding profiles.

10.3.11.3 MIME multipart response

GetCoverage responses transferred using MIME multipart shall be encoded according to [IETF RFC 2387], with the first part containing an XML encoded Coverages element. Every part of the multipart message shall have a content-id. The part containing the Coverages element shall have a content-id equal to **“urn:ogc:wcs:1.1:coverages”**; other parts may have any content-id values. The Coverages element shall reference other parts of the multipart message by their content-id values, prefixed by **“cid:”**.

EXAMPLE 1 If a multipart message contains a part with “content-id:ottawa_temp.tiff”, then the Coverage element would reference that part with the following XML fragment:

```
<Reference xlink:href="cid:ottawa_temp.tiff"/>
```

NOTE Using prefixes other than **“cid:”**, the Coverages element can (but normally should not) reference resources external to the MIME multipart message.

10.3.11.4 SOAP response

WCS servers that implement SOAP transfer of GetCoverage operation requests shall also implement SOAP version 1.2 transfer of the corresponding GetCoverage operation responses. These operation responses shall be encoded as specified in Annex F. When the “store” parameter has the value “true”, the body of the SOAP envelope shall contain one XML encoded Coverages element with one or more references to URLs of files containing coverage content and metadata.

10.3.11.5 SOAP with Attachments response

When the “store” parameter is absent or has the value “false”, responses to GetCoverage SOAP requests shall be encoded as SOAP with Attachments as defined in [W3C Note] (but using SOAP 1.2 rather than SOAP 1.1). These responses shall consist of MIME multipart messages, as specified in 10.3.11.3, with the Coverages element contained within a SOAP 1.2 envelope, encoded as specified in Annex F.

10.3.12 Coverages XML encoding

Application Profiles should be written for all formats, as specified in Subclause 9.3.2.2. These Profiles shall specify how any of the coverage metadata specified in Subclause 10.3.9 shall be included in the same file as the coverage grid values. XML encoding is not expected to be used for the coverage grid values part of that file.

For each coverage, XML encoding should be used for the required coverage metadata specified in Subclause 10.3.9, when not included in the same file as the coverage grid values. It is also possible for an Application Profile to specify use of XML encoding for some of the metadata that is included in the same file as the coverage grid values. When XML encoding is used, the XML encoding referenced in Subclause 10.3.9 shall be used.

EXAMPLE The GML in JPEG 2000 encoding specification [OGC 05-047r3] could be adapted to encode both coverage grid values and associated metadata.

10.4 GetCoverage operation examples

10.4.1 A minimal GetCoverage request

A client might issue the following GetCoverage operation request with minimum contents – encoded in KVP:

```
http://my.service.org/path/script?
service=WCS
&version=1.1.0
&request=GetCoverage
&identifier=Cov123
&BoundingBox=-71,47,-66,51,urn:ogc:def:crs:OGC:2:84
&format=image/netcdf
```

This corresponds to the following minimum request encoded in XML:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1.0/wcsGetCoverage.xsd"
service="WCS" version="1.1.0">
  <Identifier>Cov123</Identifier>
  <DomainSubset>
    <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
      <ows:LowerCorner>-71 47</ows:LowerCorner>
      <ows:UpperCorner>-66 51</ows:UpperCorner>
    </ows:BoundingBox>
  </DomainSubset>
  <Output format="image/netcdf"/>
</GetCoverage>
```

This would request the portion of coverage “Cov123” falling within the given longitude and latitude bounds, with output encoded according to the “image/netcdf” convention for WCS (as specified in a separate profile). This request omits the following parameters:

- a) TimeSequence (thus requesting coverage values for all available times)

- b) RangeSubset (thus requesting all fields of “Cov123” in their entirety);
- c) Output CRS parameters: GridBaseCRS, GridType, GridCS, GridOrigin, GridOffsets.
This requests output expressed in the native CRS of the coverage “Cov123”.
- d) Store (thus requesting the coverage in-line rather than posted to a URL)

10.4.2 A longer GetCoverage request

The following adds a simple range subset and a simple 2D output GridCRS to the previous example. It also asks the server to store the result rather than deliver it inline:

```
http://my.service.org/path/script?
service=WCS&version=1.1.0&request=GetCoverage
&identifier=Cov123&format=image/netcdf
&BoundingBox=-71,47,-66,51,urn:ogc:def:crs:OGC:2:84
&RangeSubset=radiance[waveLength[0.6,0.9]];temperature:cubic
&GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618
&GridOffsets=10,10
&store=true
```

The above KVP request is identical to the previous one except that it requests

- a) a subset of the coverage range (only radiance for wavelength values 0.6 and 0.9, with the default spatial interpolation for this field; and temperature with bicubic interpolation);
- b) an output grid with 10x10 meter pixels aligned with the northing/easting axes of WGS 84 UTM meters, Zone 18 north (EPSG 32618).

This corresponds to the following XML request:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows"
xmlns:owcs="http://www.opengis.net/wcs/1.1/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1.0/wcsGetCoverage.xsd"
service="WCS" version="1.1.0">
  <Identifier>Cov123</Identifier>
  <DomainSubset>
    <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
      <ows:LowerCorner>-71 47</ows:LowerCorner>
      <ows:UpperCorner>-66 51</ows:UpperCorner>
    </ows:BoundingBox>
  </DomainSubset>
  <RangeSubset>
    <FieldSubset>
      <owcs:Identifier>Radiance</owcs:Identifier>
      <AxisSubset>
        <Identifier>waveLength</Identifier>
        <Key>0.6</Key>
        <Key>0.9</Key>
      </AxisSubset>
    </FieldSubset>
    <FieldSubset>
      <owcs:Identifier>Temperature</owcs:Identifier>
      <InterpolationType>cubic</InterpolationType>
    </FieldSubset>
  </RangeSubset>
```

```

<Output format="image/netcdf" store="true">
  <GridCRS>
    <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS>
    <GridOffsets>10 10</GridOffsets>
  </GridCRS>
</Output>
</GetCoverage>

```

The request (whether in KVP or XML) implicitly uses default values for the following GridCRS parameters:

- a) GridType: default is “urn:ogc:def:method:WCS:1.1:2dSimpleGrid”
(This GridType disallows rotation or skew relative to the GridBaseCRS – therefore GridOffsets has only two numbers.)
- b) GridCS: default is “urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS”
- c) GridOrigin: default is “0,0” (KVP) or “0 0” (XML).

10.4.3 Another GetCoverage request

The following request adds a time constraint and non-default gridCRS parameters for the output coverage.

```

http://my.service.org/path/script?
service=WCS&version=1.1.0&request=GetCoverage
&identifier=Cov123&format=image/netcdf
&TimeSequence=2006-08-01/2006-09-01/P1D
&GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618
&GridType=urn:ogc:def:method:WCS:1.1:2dGridin2dCrs
&GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS
&GridOrigin=3000,4000
&GridOffsets=8.0,6.0,-6.0,8.0

```

The TimeSequence KVP parameter provides the time constraint: daily (“P1D”) between Aug. 1 and Sept. 1, 2006, inclusive. As before, the requested output grid is defined relative to WGS 84 UTM meters, Zone 18 North (EPSG 32618) – but rotated relative to the northing/easting lines: the two pairs of offsets in GridOffsets form 3-4-5 right triangles, for a 10x10 meter grid spacing and a counter-clockwise rotation of about 37 degrees. The output grid has its origin at (3000,4000) rather than (0,0).

This corresponds to the following XML request:

```

<?xml version="1.0" encoding="UTF-8"?>
<GetCoverage xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1.0/wcsGetCoverage.xsd"
service="WCS" version="1.1.0">
  <Identifier>Cov123</Identifier>
  <DomainSubset>
    <ows:BoundingBox crs="urn:ogc:def:crs:OGC:2:84">
      <ows:LowerCorner>-71 47</ows:LowerCorner>
      <ows:UpperCorner>-66 51</ows:UpperCorner>
    </ows:BoundingBox>
    <TemporalSubset>
      <TimePeriod>

```

```

        <BeginPosition>2006-08-01</BeginPosition>
        <EndPosition>2006-09-01</EndPosition>
        <TimeResolution>P1D</TimeResolution>
    </TimePeriod>
</TemporalSubset>
</DomainSubset>
<Output format="image/netcdf">
    <GridCRS>
        <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS>
        <GridType>urn:ogc:def:method:WCS:1.1:2dGridin2dCr</GridType>
        <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
        <GridOrigin>3000 4000</GridOrigin>
        <GridOffsets>6.0 8.0 -8.0 6.0</GridOffsets>
    </GridCRS>
</Output>
</GetCoverage>

```

10.4.4 Example responses

In response to such a request, the WCS server might generate a document that looks like:

```

<?xml version="1.0" encoding="UTF-8"?>
<Coverages xmlns="http://www.opengis.net/ows/1.1"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/ows/1.1 ../owsCoverages.xsd">
    <Coverage>
        <Abstract>Coverage created from GetCoverage operation request to a
WCS</Abstract>
        <Reference xlink:href="coverage/image.tiff"
xlink:role="urn:ogc:def:role:WCS:1.1:coverage"/>
        <Reference xlink:href="coverage/metadata.xml"
xlink:role="urn:ogc:def:role:WCS:1.1:metadata"/>
    </Coverage>
</Coverages>

```


Annex A (normative)

Conformance

Specific conformance tests for a Web Coverage Service will be added in a future revision of this specification. At the moment, a WCS implementation must satisfy the following system characteristics to be minimally conformant with this specification:

- a) *WCS Clients and servers* must support the *GetCapabilities*, *DescribeCoverage*, and *GetCoverage* operations.
- b) *WCS clients* must issue *GetCapabilities* requests in Key-Value Pair (KVP), XML, or SOAP form. *GetCapabilities* KVP requests must conform to Subclause **8.2.2**. *GetCapabilities* XML requests must conform to Subclause **8.2.3**, and must be valid against the XML Schema definition in *wcsGetCapabilities.xsd*. *GetCapabilities* SOAP requests must conform to Subclause **8.2.4**.
- c) *WCS servers* must respond to a *GetCapabilities* KVP or XML request with an XML document that conforms to Subclause **8.3**, and is valid against the XML Schema definition in *wcsGetCapabilities.xsd* and *wcsContents.xsd*. The response to a *GetCapabilities* SOAP request shall encapsulate this XML document in the body of a SOAP envelope, as specified in Annex F.
- d) *WCS clients* must issue *DescribeCoverage* requests in Key-Value Pair (KVP) or XML form. *DescribeCoverage* KVP requests must conform to Subclause **9.2.2**. *DescribeCoverage* XML requests must conform to Subclause **9.2.3**, and must be valid against the XML Schema definition in *wcsDescribeCoverage.xsd*. *DescribeCoverage* SOAP requests must conform to Subclause **9.2.4**.
- e) *WCS servers* must respond to a *DescribeCoverage* KVP or XML request with an XML document that conforms to Subclause **9.3**, and is valid against the XML Schema definition in *wcsDescribeCoverage.xsd*. The response to a *DescribeCoverage* SOAP request shall encapsulate this XML document in the body of a SOAP envelope, as specified in Annex F.
- f) *WCS clients* must issue *GetCoverage* requests in Key-Value Pair (KVP) or XML form. *GetCoverage* KVP requests must conform to Subclause **10.2.2**. *GetCoverage* XML requests must conform to Subclause **10.2.3**, and must be valid against the XML Schema definition in *wcsGetCoverage.xsd*. *GetCoverage* SOAP requests must conform to Subclause **10.2.4**.
- g) *WCS servers* must be able to respond to a *GetCoverage* KVP or XML operation with an XML document that conforms to Subclause **10.3.11**. The response to a *GetCoverage* SOAP request shall encapsulate this XML document in the body of a SOAP envelope, as specified in Subclause **10.3.11** and Annex F.
- h) All paragraphs in the normative clauses of this specification that use the keywords "required", "shall", and "shall not" must be satisfied.

Annex B (informative)

WCS XML Schemas

In addition to this document, this specification includes normative XML Schema Document files. The WCS-specific XML Schema Documents are posted online at the URL <http://schemas.opengis.net/wcs/1.1.0>. These XML Schema Documents are also bundled with the present document. In the event of a discrepancy between the bundled and online versions of the XML Schema files, the online files shall be considered authoritative.

The WCS abilities specified in this document use eleven specified XML Schema Documents included in the zip file with this document, which are named:

- a) wcsGetCapabilities.xsd
- b) wcsDescribeCoverage.xsd
- c) wcsGetCoverage.xsd
- d) wcsContents.xsd
- e) wcsCommon.xsd
- f) wcsAll.xsd
- g) wcs:GridCRS.xsd (specified in Annex G)
- h) owsDomainType.xsd (specified in Annex E)
- i) owsManifest.xsd (specified in Subclause I.2)
- j) owsCoverages.xsd (specified in Subclause I.3)
- k) owsInterpolationMethod.xsd (specified in Subclause I.4)

The last four XML Schema Documents listed above are likely to be useful in multiple OWS interface specifications, and are thus specified in the (temporary) namespace “<http://www.opengis.net/wcs/1.1/ows>”, identified by the the prefix “owcs”.

The owsInterpolationMethod.xsd XML Schema Document references an XML document that specifies some values and meanings of the InterpolationType specified in Subclause I.4. That XML document is named interpolationMethods.xml, and is referenced at <http://schemas.opengis.net/wcs/1.1.0/interpolationMethods.xml>.

These XML Schema Documents use and build on the OWS common XML Schema Documents listed in Annex A of [OGC 05-008], specified in the “ows” namespace and named:

- a) ows19115subset.xsd
- b) owsCommon.xsd
- c) owsExceptionReport.xsd
- d) owsServiceProvider.xsd

- e) owsGetCapabilities.xsd (slightly modified, in temporary owcs namespace)
- f) owsOperationsMetadata.xsd (slightly modified, in temporary owcs namespace)
- g) owsServiceIdentification.xsd (slightly modified, in owcs namespace)

All these XML Schemas contain documentation of the meaning of each element and attribute, and this documentation is considered normative as specified in Subclause 11.6.3 of [OGC 05-008].

Annex C (informative)

XML document examples

C.1 Introduction

As an aid to understanding and a guide for implementation, this annex contains more example XML documents which are valid according to the XML schemas referenced in Annex B.

C.2 GetCapabilities operation XML encoding examples

To request a Capabilities document, a client could issue the following GetCapabilities minimal operation request encoded as KVP:

```
http://hostname:port/path&service=WCS&request=GetCapabilities
```

The corresponding minimum request encoded in XML is:

```
<?xml version="1.0" encoding="UTF-8"?>
<GetCapabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1/wcsGetCapabilities.xsd"
service="WCS"/>
```

In response to such a GetCapabilities operation request, a WCS server might generate a document that looks like the following:

```
<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wcs/1.1"
xmlns:ows="http://www.opengis.net/ows"
xmlns:owcs="http://www.opengis.net/wcs/1.1/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/wcs/1.1
http://schemas.opengis.net/wcs/1.1/wcsGetCapabilities.xsd
http://www.opengis.net/wcs/1.1/ows
http://schemas.opengis.net/ows/1.0/owsDataIdentification.xsd"
version="1.1.0" updateSequence="ABC123">
  <owcs:ServiceIdentification>
    <ows:Title>Acme Corp. Coverage Server</ows:Title>
    <ows:Abstract>Web Coverage Server maintained by Acme Corporation.
  </ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>USA</ows:Keyword>
    </ows:Keywords>
    <owcs:ServiceType>OGC WCS</owcs:ServiceType>
    <owcs:ServiceTypeVersion>1.0.0</owcs:ServiceTypeVersion>
    <owcs:ServiceTypeVersion>0.8.3</owcs:ServiceTypeVersion>
    <owcs:Fees>NONE</owcs:Fees>
    <owcs:AccessConstraints>NONE</owcs:AccessConstraints>
  </owcs:ServiceIdentification>
  <ows:ServiceProvider>
```

```

<ows:ProviderName>Acme Corporation</ows:ProviderName>
<ows:ProviderSite xlink:href="http://hostname/" />
<ows:ServiceContact>
  <ows:IndividualName>Jeff Smith, Server
Administrator</ows:IndividualName>
  <ows:PositionName>Computer Scientist</ows:PositionName>
  <ows:ContactInfo>
    <ows:Phone>
      <ows:Voice>+1 301 555-1212</ows:Voice>
      <ows:Facsimile>+1 301 555-1212</ows:Facsimile>
    </ows:Phone>
    <ows:Address>
      <ows:DeliveryPoint>NASA Goddard Space Flight
Center</ows:DeliveryPoint>
      <ows:City>Greenbelt</ows:City>
      <ows:AdministrativeArea>MD</ows:AdministrativeArea>
      <ows:PostalCode>20771</ows:PostalCode>
      <ows:Country>USA</ows:Country>

    <ows:ElectronicMailAddress>user@host.com</ows:ElectronicMailAddress>
      </ows:Address>
    </ows:ContactInfo>
    <ows:Role>TBD</ows:Role>
  </ows:ServiceContact>
</ows:ServiceProvider>
<owcs:OperationsMetadata>
  <owcs:Operation name="GetCapabilities">
    <owcs:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://hostname:port/path?" />
        <ows:Post xlink:href="http://hostname:port/path?" />
      </ows:HTTP>
    </owcs:DCP>
  </owcs:Operation>
  <owcs:Operation name="GetCoverage">
    <owcs:DCP>
      <ows:HTTP>
        <ows:Post xlink:href="http://hostname:port/path?" />
      </ows:HTTP>
    </owcs:DCP>
    <owcs:Parameter name="Format">
      <owcs:AllowedValues>
        <ows:Value>image/gif</ows:Value>
        <ows:Value>image/png</ows:Value>
        <ows:Value>image/jpeg</ows:Value>
      </owcs:AllowedValues>
    </owcs:Parameter>
  </owcs:Operation>
  <owcs:Operation name="DescribeCoverage">
    <owcs:DCP>
      <ows:HTTP>
        <ows:Get xlink:href="http://hostname:port/path?" />
      </ows:HTTP>
    </owcs:DCP>
    <owcs:Parameter name="Format">
      <owcs:AllowedValues>
        <ows:Value>text/xml</ows:Value>
      </owcs:AllowedValues>
    </owcs:Parameter>
  </owcs:Operation>
</owcs:OperationsMetadata>
<Contents>
  <CoverageSummary>

```

```

    <ows:Title>TBD</ows:Title>
    <ows:Abstract>TBD</ows:Abstract>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-180 -90</ows:LowerCorner>
      <ows:UpperCorner>180 90</ows:UpperCorner>
    </ows:WGS84BoundingBox>
    <Identifier>TBD</Identifier>
  </CoverageSummary>
</Contents>
</Capabilities>

```

Here is another longer example response to a GetCapabilities operation request. It features fully detailed service metadata – Service Identification, Service Provider, OperationsMetadata, and Contents sections – the latter showing hierarchical descriptions afforded by CoverageSummary.

```

<?xml version="1.0" encoding="UTF-8"?>
<Capabilities xmlns="http://www.opengis.net/wcs/1.1"
  xmlns:ows="http://www.opengis.net/ows"
  xmlns:owcs="http://www.opengis.net/wcs/1.1/ows"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/1.1
    ../wcsGetCapabilities.xsd http://www.opengis.net/wcs/1.1/ows
    ../owsDataIdentification.xsd" version="1.1.0" updateSequence="1.0">
  <!-- ***** -->
  <!-- * SERVICE IDENTIFICATION SECTION * -->
  <!-- ***** -->
  <owcs:ServiceIdentification>
    <ows:Title>CubeWerx Demonstation WCS</ows:Title>
    <ows:Abstract>A demonstration server used to illustrate CubeWerx's
    compliance with the Web Coverage Service 1.1.0 implementation
    specification</ows:Abstract>
    <ows:Keywords>
      <ows:Keyword>Web Coverage Service</ows:Keyword>
      <ows:Keyword>06-083</ows:Keyword>
      <ows:Keyword>CubeWerx</ows:Keyword>
      <ows:Keyword>GeoTIFF</ows:Keyword>
      <ows:Keyword>Imagery</ows:Keyword>
    </ows:Keywords>
    <owcs:ServiceType>WCS</owcs:ServiceType>
    <owcs:ServiceTypeVersion>1.0.0</owcs:ServiceTypeVersion>
    <owcs:ServiceTypeVersion>1.1.0</owcs:ServiceTypeVersion>
    <owcs:Fees>NONE</owcs:Fees>
    <owcs:AccessConstraints>NONE</owcs:AccessConstraints>
  </owcs:ServiceIdentification>
  <!-- ***** -->
  <!-- * SERVICE PROVIDER SECTION * -->
  <!-- ***** -->
  <ows:ServiceProvider>
    <ows:ProviderName>CubeWerx Inc.</ows:ProviderName>
    <ows:ProviderSite xlink:href="http://www.cubewerx.com"/>
    <ows:ServiceContact>
      <ows:IndividualName>Panagiotis (Peter) A.
    Vretanos</ows:IndividualName>
      <ows:PositionName>Senior Developer</ows:PositionName>
      <ows:ContactInfo>
        <ows:Phone>
          <ows:Voice>123-456-7890</ows:Voice>
          <ows:Facsimile>234-567-8901</ows:Facsimile>
        </ows:Phone>
        <ows:Address>

```

```

    <ows:DeliveryPoint>15 rue Gamelin</ows:DeliveryPoint>
    <ows:DeliveryPoint>Suite 506</ows:DeliveryPoint>
    <ows:City>Gatineau</ows:City>
    <ows:AdministrativeArea>Quebec</ows:AdministrativeArea>
    <ows:PostalCode>J8Y 6N5</ows:PostalCode>
    <ows:Country>Canada</ows:Country>

    <ows:ElectronicMailAddress>pvretano[at]cubewerx[dot]com</ows:Electron
icMailAddress>
    </ows:Address>
    <ows:OnlineResource
xlink:href="http://www.cubewerx.com/~pvretano" />
    <ows:HoursOfService>24x7x365</ows:HoursOfService>
    <ows:ContactInstructions>email</ows:ContactInstructions>
    </ows:ContactInfo>
    <ows:Role>Developer</ows:Role>
  </ows:ServiceContact>
</ows:ServiceProvider>
<!-- ***** -->
<!-- * OPERATIONS METADATA * -->
<!-- ***** -->
<owcs:OperationsMetadata>
  <owcs:Operation name="GetCapabilities">
    <owcs:DCP>
      <owcs:HTTP>
        <owcs:Get
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cgi? " />
        </owcs:HTTP>
      </owcs:DCP>
      <owcs:Parameter name="service">
        <owcs:AllowedValues>
          <owcs:Value>WCS</owcs:Value>
        </owcs:AllowedValues>
      </owcs:Parameter>
      <owcs:Parameter name="version">
        <owcs:AllowedValues>
          <owcs:Value>1.0.0</owcs:Value>
          <owcs:Value>1.1.0</owcs:Value>
        </owcs:AllowedValues>
      </owcs:Parameter>
      <owcs:Parameter name="AcceptVersions">
        <owcs:AllowedValues>
          <owcs:Value>1.0.0</owcs:Value>
          <owcs:Value>1.1.0</owcs:Value>
        </owcs:AllowedValues>
      </owcs:Parameter>
    </owcs:Operation>
    <owcs:Operation name="DescribeCoverage">
      <owcs:DCP>
        <owcs:HTTP>
          <owcs:Get
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cgi? " />
          </owcs:HTTP>
        </owcs:DCP>
        <owcs:Parameter name="service">
          <owcs:AllowedValues>
            <owcs:Value>WCS</owcs:Value>
          </owcs:AllowedValues>
        </owcs:Parameter>
        <owcs:Parameter name="version">
          <owcs:AllowedValues>
            <owcs:Value>1.0.0</owcs:Value>
            <owcs:Value>1.1.0</owcs:Value>
          </owcs:AllowedValues>
        </owcs:Parameter>
      </owcs:Operation>
    </owcs:OperationsMetadata>
  </owcs:OperationsMetadata>

```

```

        </owcs:AllowedValues>
    </owcs:Parameter>
    <owcs:Parameter name="Identifier">
        <owcs:AllowedValues>
            <owcs:Value>ETOP02</owcs:Value>
            <owcs:Value>GTOPO30</owcs:Value>
            <owcs:Value>RELIEF</owcs:Value>
        </owcs:AllowedValues>
    </owcs:Parameter>
</owcs:Operation>
<owcs:Operation name="GetCoverage">
    <owcs:DCP>
        <owcs:HTTP>
            <owcs:Get
xlink:href="http://demo.cubewerx.com/demo/cubeserv/cubeserv.cgi? "/>
            </owcs:HTTP>
        </owcs:DCP>
        <owcs:Parameter name="service">
            <owcs:AllowedValues>
                <owcs:Value>WCS</owcs:Value>
            </owcs:AllowedValues>
        </owcs:Parameter>
        <owcs:Parameter name="version">
            <owcs:AllowedValues>
                <owcs:Value>1.0.0</owcs:Value>
                <owcs:Value>1.1.0</owcs:Value>
            </owcs:AllowedValues>
        </owcs:Parameter>
        <owcs:Parameter name="Identifier">
            <owcs:AllowedValues>
                <owcs:Value>ETOP02</owcs:Value>
                <owcs:Value>GTOPO30</owcs:Value>
                <owcs:Value>RELIEF</owcs:Value>
            </owcs:AllowedValues>
        </owcs:Parameter>
        <owcs:Parameter name="InterpolationType">
            <owcs:AllowedValues>
                <owcs:Value>nearest</owcs:Value>
                <owcs:Value>linear</owcs:Value>
            </owcs:AllowedValues>
        </owcs:Parameter>
        <owcs:Parameter name="format">
            <owcs:AllowedValues>
                <owcs:Value>image/tiff</owcs:Value>
                <owcs:Value>image/tiff;
PhotometricInterpretation=RGB</owcs:Value>
                <owcs:Value>image/png</owcs:Value>
                <owcs:Value>image/png;
PhotometricInterpretation=PaletteColor</owcs:Value>
                <owcs:Value>image/png;
PhotometricInterpretation=RGB</owcs:Value>
                <owcs:Value>image/gif</owcs:Value>
                <owcs:Value>image/jpeg</owcs:Value>
                <owcs:Value>image/ppm</owcs:Value>
            </owcs:AllowedValues>
        </owcs:Parameter>
    </owcs:Operation>
</owcs:OperationsMetadata>
<!-- ***** -->
<!-- * CONTENTS SECTION * -->
<!-- ***** -->
<Contents>
    <CoverageSummary>

```



```

<ows:Title>Coverage 123</ows:Title>
<ows:Abstract>Use at your own risk</ows:Abstract>
<ows:Keywords>
  <ows:Keyword>images</ows:Keyword>
  <ows:Keyword>scenes</ows:Keyword>
  <ows:Keyword>photo</ows:Keyword>
</ows:Keywords>
<ows:WGS84BoundingBox crs="urn:ogc:def:crs:OGC:2:84"
dimensions="2">
  <ows:LowerCorner>-60 -130</ows:LowerCorner>
  <ows:UpperCorner>45 90</ows:UpperCorner>
</ows:WGS84BoundingBox>
<SupportedCRS>urn:ogc:def:crs:EPSG:6.6:63266405</SupportedCRS>
<SupportedCRS>urn:ogc:def:crs:EPSG:6.6:32618</SupportedCRS>
<SupportedFormat>image/netcdf; cf=1.0</SupportedFormat>
<SupportedFormat>image/tiff; geo=true</SupportedFormat>
<SupportedFormat>image/hdfeos; type=grid</SupportedFormat>
<Identifier>Cov123</Identifier>
</CoverageSummary>
<CoverageSummary>
  <ows:Title>Landsat 7 collection</ows:Title>
  <ows:Abstract>Comprehensive global Landsat imagery: 2001-
present. Provided as a global mosaic; also as regional subsets and a few
individual scenes. Other scenes are described in an external CSW
catalog.</ows:Abstract>
  <ows:Keywords>
    <ows:Keyword>Land</ows:Keyword>
    <ows:Keyword>Multispectral</ows:Keyword>
  </ows:Keywords>
  <ows:WGS84BoundingBox>
    <ows:LowerCorner>-180 -90</ows:LowerCorner>
    <ows:UpperCorner>180 90</ows:UpperCorner>
  </ows:WGS84BoundingBox>
  <SupportedCRS>urn:ogc:def:crs:EPSG:6.6:63266405</SupportedCRS>
  <SupportedCRS>urn:ogc:def:crs:EPSG:6.6:32619</SupportedCRS>
  <SupportedCRS>urn:ogc:def:crs:EPSG:6.6:32618</SupportedCRS>
  <SupportedCRS>urn:ogc:def:crs:EPSG:6.6:32617</SupportedCRS>
  <SupportedCRS>urn:ogc:def:crs:EPSG:6.6:32616</SupportedCRS>
  <SupportedFormat>image/tiff; geo=true</SupportedFormat>
  <CoverageSummary>
    <ows:Title>N. America Landsat: 2001</ows:Title>
    <ows:WGS84BoundingBox>
      <ows:LowerCorner>-120 25</ows:LowerCorner>
      <ows:UpperCorner>-40 75</ows:UpperCorner>
    </ows:WGS84BoundingBox>
    <CoverageSummary>
      <ows:Title>Scene XYZ.1232456789.ABC</ows:Title>
      <ows:WGS84BoundingBox>
        <ows:LowerCorner>-76.15 47.9</ows:LowerCorner>
        <ows:UpperCorner>-75.9 48.15</ows:UpperCorner>
      </ows:WGS84BoundingBox>
      <Identifier>XYZ.1232456789.ABC</Identifier>
    </CoverageSummary>
    <Identifier>NorthAmerica_Landsat</Identifier>
  </CoverageSummary>
  <Identifier>TopLevelLandsatMosaic</Identifier>
</CoverageSummary>
<OtherSource xlink:href="http://www.myservice.com/csw?"/>
</Contents>
</Capabilities>

```

C.3 DescribeCoverage operation XML encoding examples

These examples are provided in Subclause 9.5.

C.4 GetCoverage XML operation XML encoding examples

These examples are provided in Subclause 10.4.

Annex D (informative)

UML model

D.1 Introduction

This annex provides a UML model of the WCS interface, using the OGC/ISO profile of UML summarized in Subclause 5.2 of the OWS Common [OGC 05-008].

Figure D.1 is a UML diagram summarizing the WCS interface. This class diagram shows that the `WebCoverageService` class inherits the `getCapabilities` operation from the abstract `OGCWebService` class, which is common to all OGC Web Services. The `WebCoverageService` class adds the `getCoverage` and `describeCoverage` operations. (The capitalization of class, operation, and data type names uses the OGC/ISO profile of UML.)

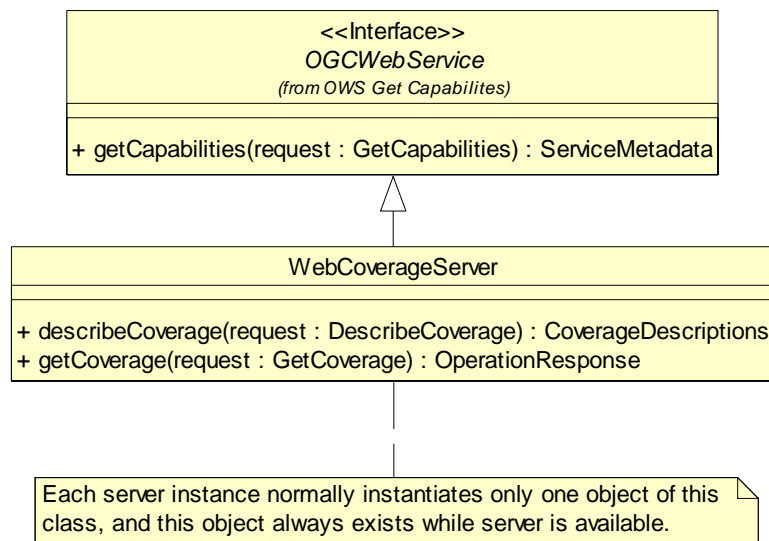


Figure D.1 — WCS interface UML class diagram

Each of the three operations uses a request and a response data type, each of which is defined by one or more additional UML classes. The following subclauses provide a more complete UML model of the WCS interface, adding UML classes defining the operation request and response data types.

D.2 UML packages

The WCS interface UML model is organized in eight packages, as shown in the package diagram in Figure D.2. These eight WCS-specific packages make use of most OGC Web Service Common (OWS Common) packages, some ISO 19111 packages and the ISO 19108 package. Only the packages used directly are shown, without listing the classes in the OWS Common packages. This package diagram shows the dependencies among the various packages

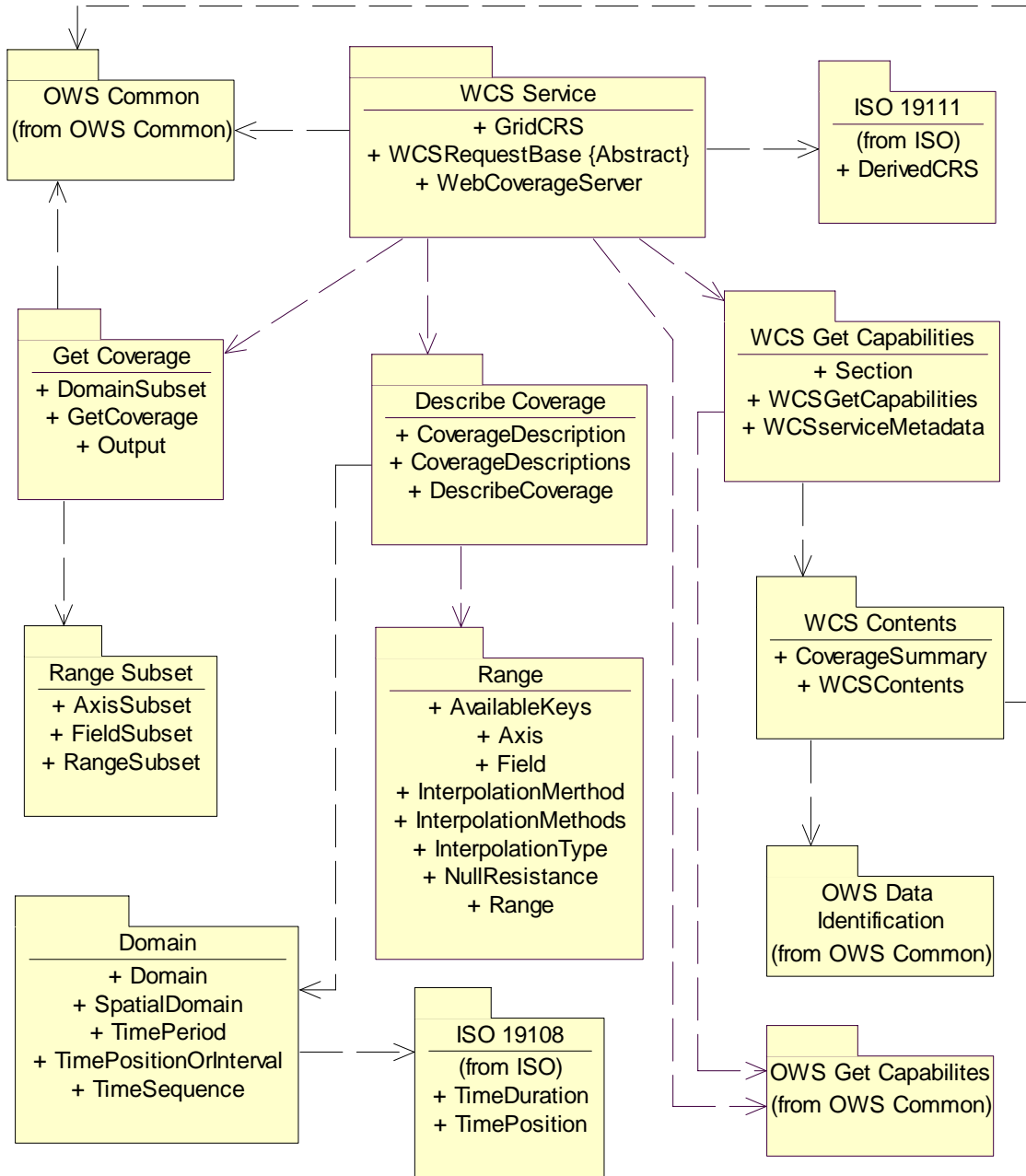


Figure D.2 — WCS interface package diagram

Each of the eight WCS-specific packages shown in Figure D.2 is described in the following subclauses.

D.3 WCS Service package

The WCS Service package is shown in the class diagram in Figure D.3. This diagram does not show the classes used by the three operation requests and responses, which are shown (with part of this package) in the Get Coverage, Describe Coverage, and WCS

GetCapabilities packages. This diagram also shows three used classes from the OGC GetCapabilities, OGC Common, and ISO 19111 packages.

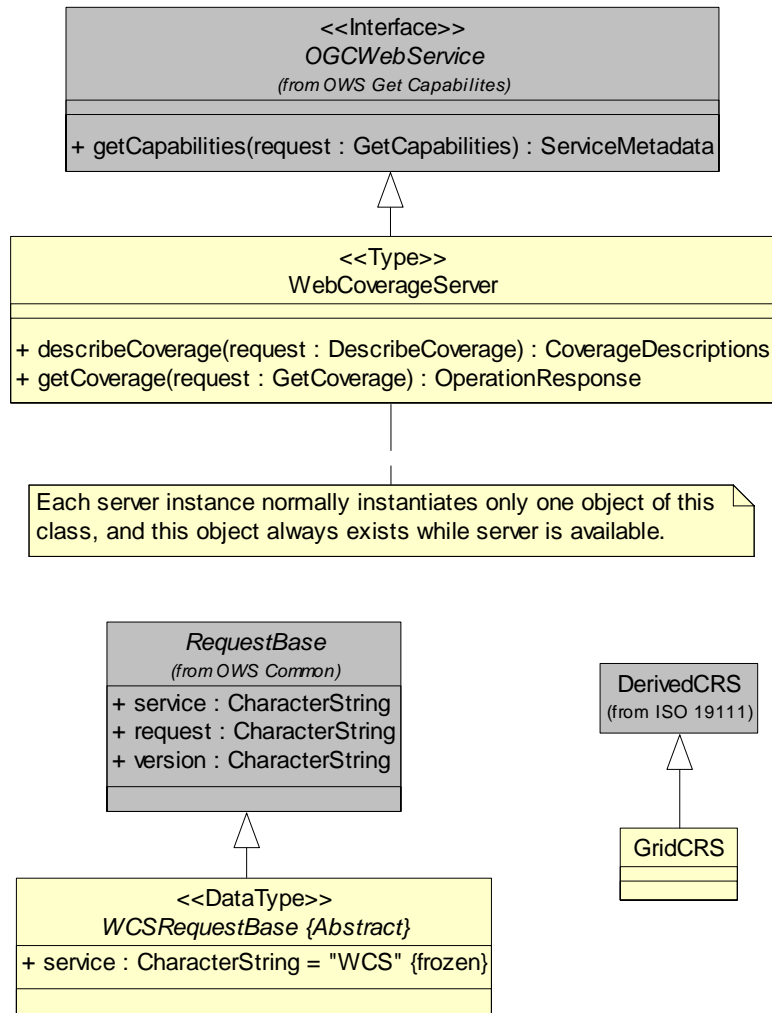


Figure D.3 — WCS Service package class diagram

D.4 Get Coverage package

The Get Coverage package is shown in the class diagram in Figure D.4. This diagram does not show details of the RangeSubset class, which is in the Range Subset package that is detailed in the following subclause. This diagram also shows several used classes from other packages. The three classes introduced by this package are further defined by Table 23 through Table 25 in this document.

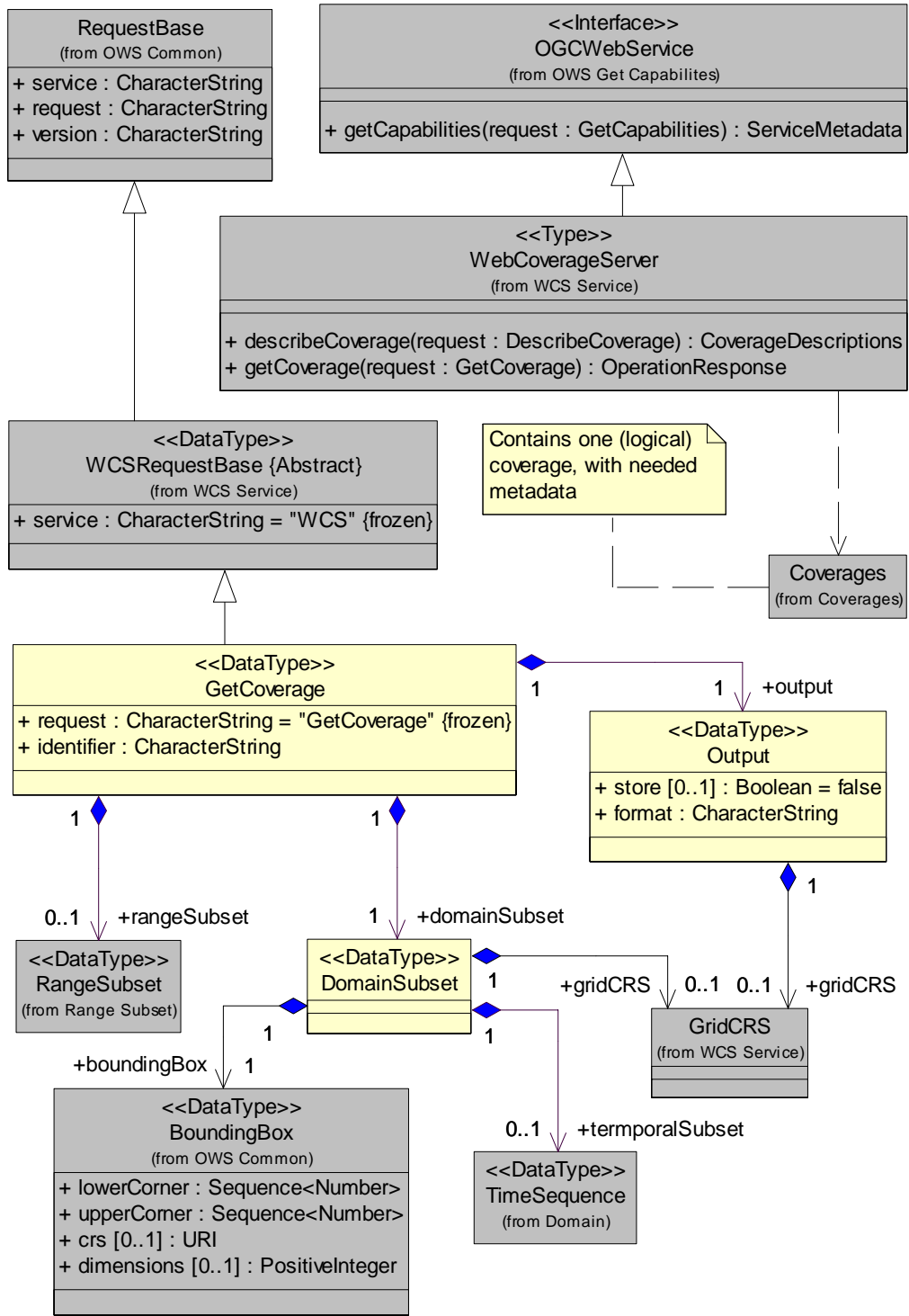


Figure D.4 — Get Coverage package class diagram

D.5 Range Subset package

The Range Subset package is shown in the class diagram in Figure D.6. This diagram also shows two used classes from other packages. The three classes introduced by this package are further defined by Table 26 through Table 28 in this document.

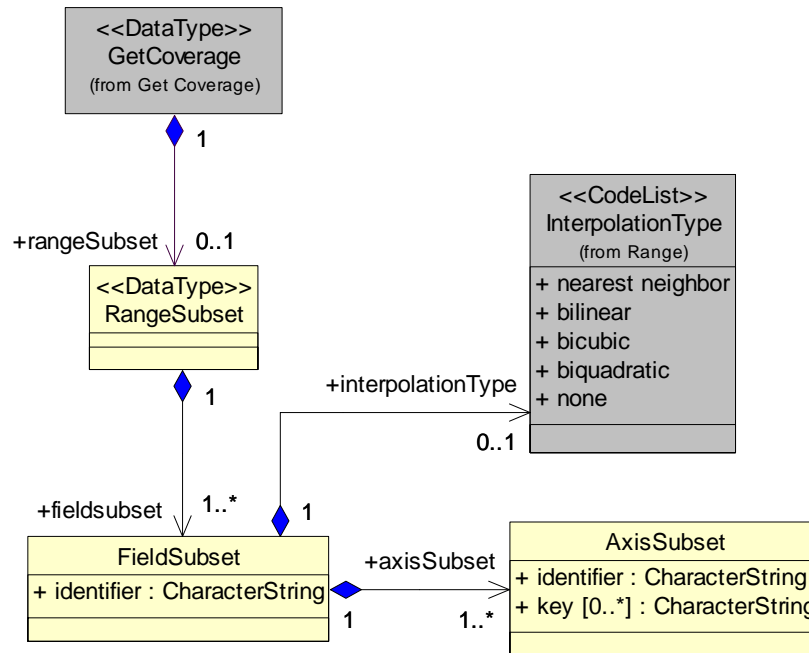


Figure D.5 — Range Subset package class diagram

D.6 Describe Coverage package

The Describe Coverage package is shown in the class diagram in Figure D.6. This diagram does not show details of the Domain and Range classes, which are in the Domain and Range packages that are detailed in following subclauses. This diagram also shows the some used classes from other packages. The three classes introduced by this package are further defined by Table 9 through Table 12 in this document.

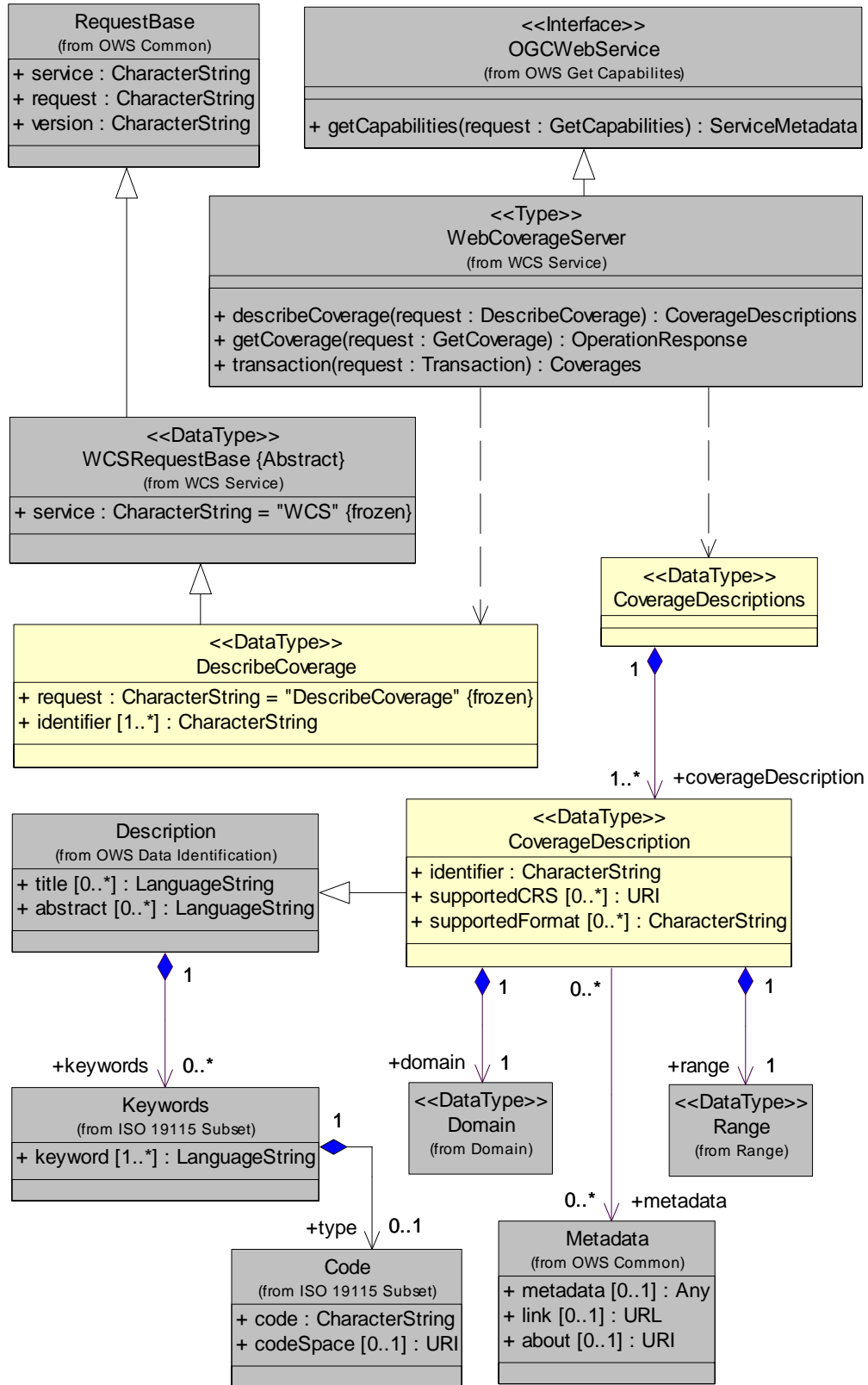


Figure D.6 — Describe Coverage package class diagram

D.7 Domain package

The Domain package is shown in the class diagram in Figure D.7. This diagram also shows one class from the Describe Coverage Package. The five classes introduced by this package are further defined by Table 13 through Table 17 in this document.

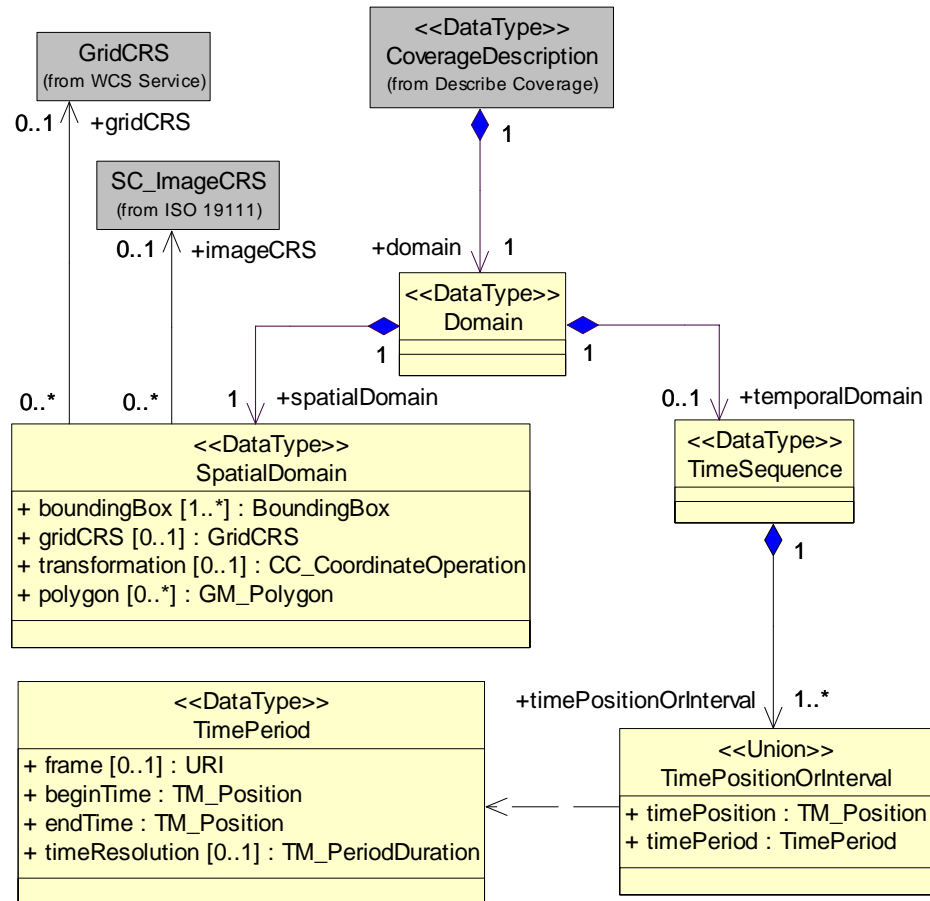


Figure D.7 — Domain package class diagram

D.8 Range package

The Range package is shown in the class diagram in Figures D.8 and D.9. This diagram also shows several used classes from other packages. The seven classes introduced by this package are further defined by Table 18 through Table 21 in this document.

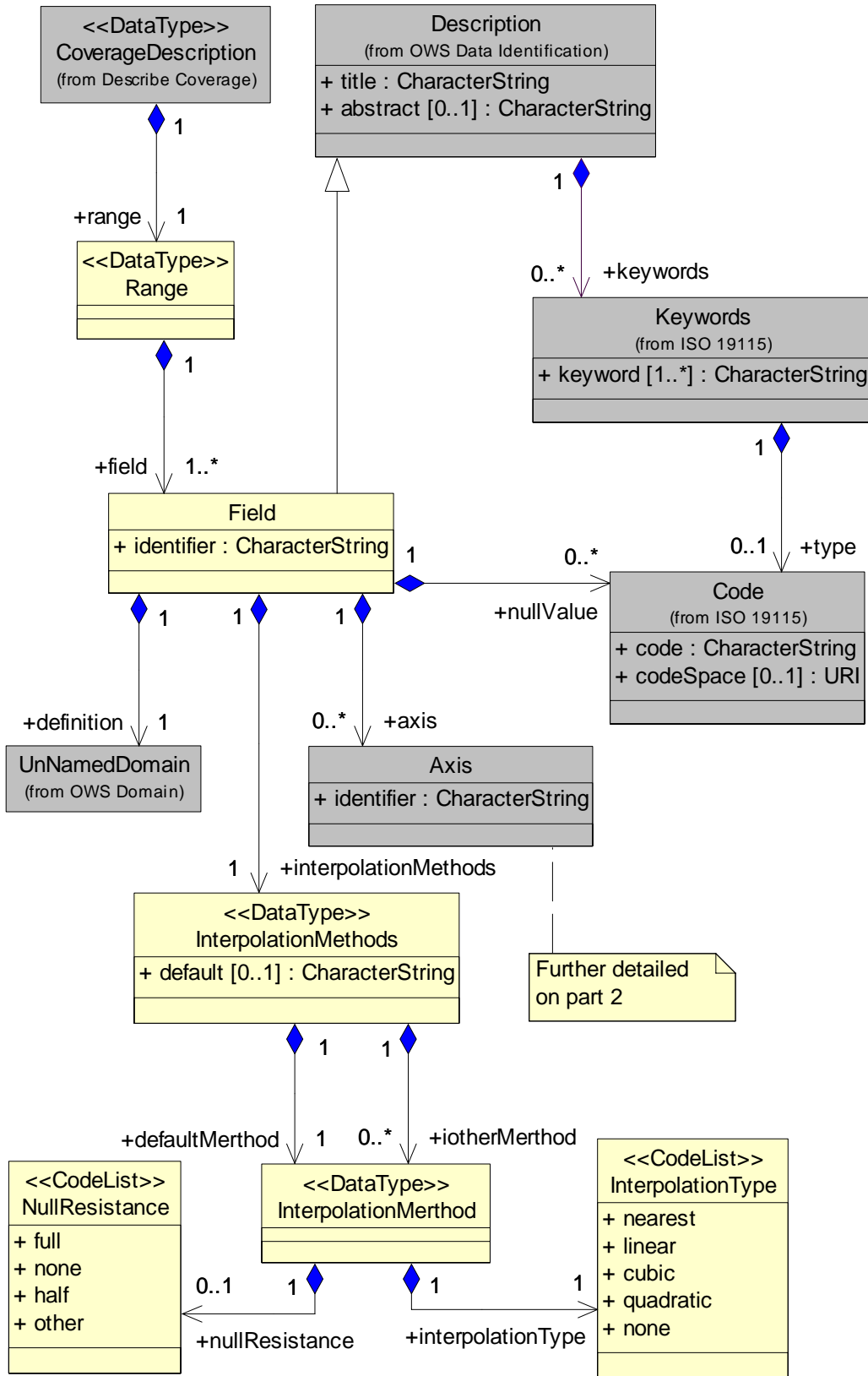


Figure D.8 — Range package class diagram, part 1

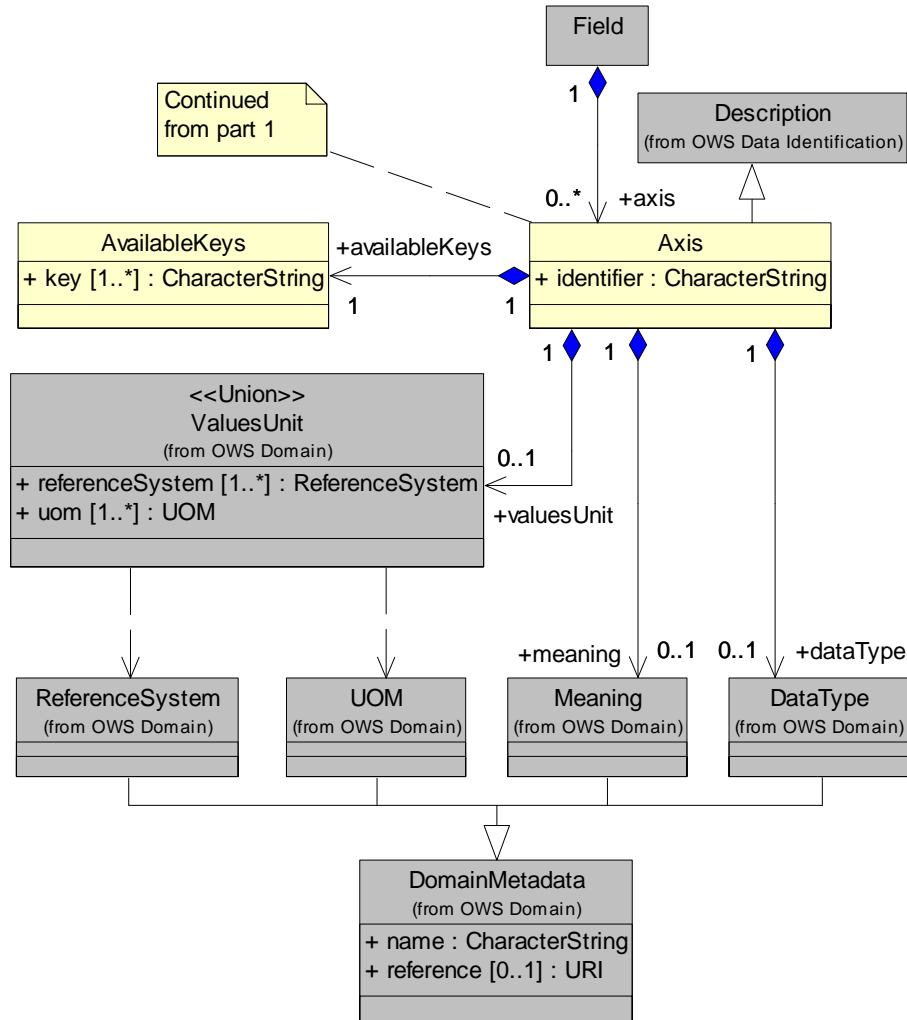


Figure D.9 — Range package class diagram, part 2

D.10 WCS Contents package

The WCS Contents package is shown in the class diagram in Figure D.11. This diagram also shows several used classes from other packages. The two classes introduced by this package are further defined by Table 6 and Table 7 in this document.

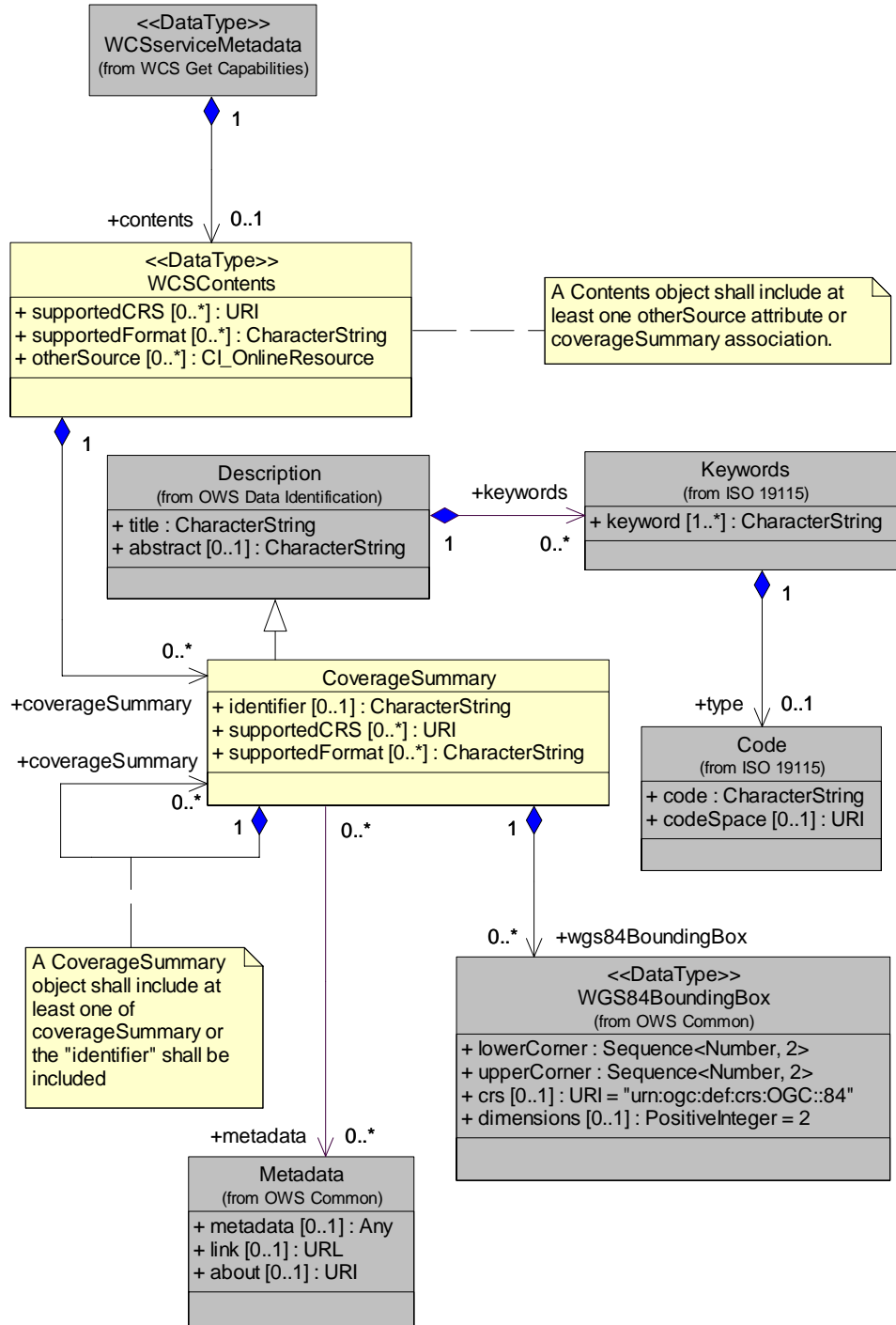


Figure D.11 — WCS Contents package class diagram

Annex E (normative)

DomainType data structure

E.1 Overview

This annex specifies the DomainType data structure that is used by WCS 1.1.

NOTE The owsDomainType.xsd is a proposed addition to OWS common [OGC 05-008], and is thus in the “ows” XML namespace. The draft expanded version of OWS Common OGC 05-008r1 included a draft owsDomainType.xsd. However that draft owsDomainType.xsd has been simplified for use here.

Some data structures can specify the allowed values and metadata for a parameter or other quantity. For all such quantities, the DomainType data structure specified in Figure E.1 and Table E.1 through Table E.8 is used.

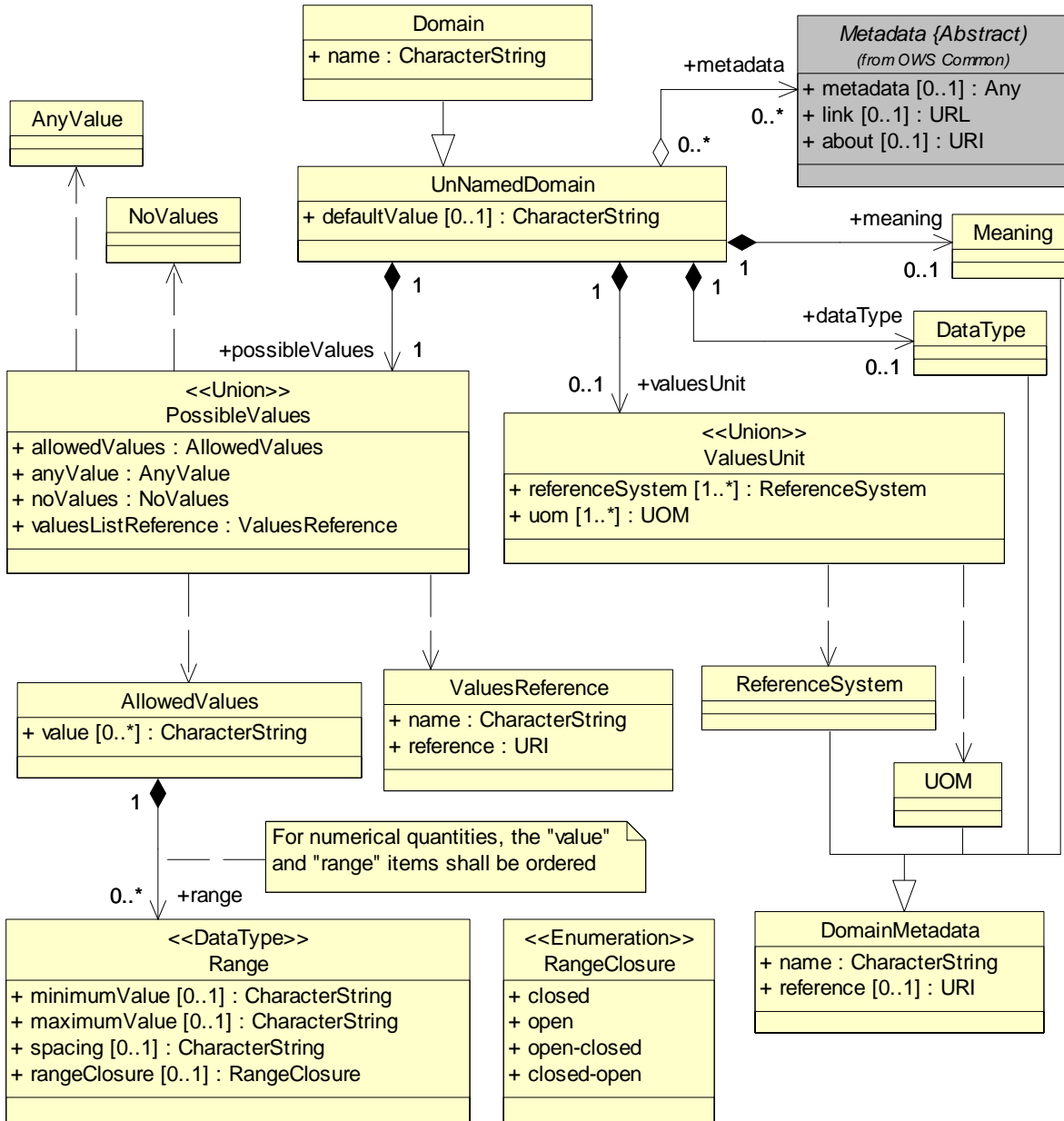


Figure E.1 — Domain package class diagram

NOTE 1 The first 6 parameters listed below (with partially grey background) are copied from Table E.2 below.

Table E.1 — Parts of DomainType data structure

Names^a	Definition	Data type	Multiplicity
possibleValues PossibleValues	Inherited from UnNamed-DomainType data structure, see Table E.2	PossibleValues data structure	One (mandatory)
defaultValue DefaultValue		Character string type, not empty	Zero or one (optional)
meaning Meaning		ows:DomainMetadata data structure	Zero or one (optional)
dataType DataType		ows:DomainMetadata data structure	Zero or one (optional)
valuesUnit ValuesUnit		ValuesUnit data structure	Zero or one (optional)
metadata Metadata		ows:DomainMetadata data structure	Zero or more (optional)
name name	Name or identifier of this quantity	Character string type, not empty	One (mandatory)
a See Table 1 for UML and XML naming conventions.			

Table E.2 — Parts of UnNamedDomainType data structure

Names^a	Definition	Data type	Multiplicity and use
possibleValues PossibleValues	Specifies the possible values of this quantity ^b	PossibleValues data structure, see Table E.3	One (mandatory)
defaultValue DefaultValue	Default value for this quantity	Character string type, not empty ^c	Zero or one (optional) Include when is a default
meaning Meaning	Reference to meaning or semantics of this value or set of values ^d	DomainMetadata data structure, see Table E.7	Zero or one (optional) Include when useful
dataType DataType	Reference to the data type of this set of values	DomainMetadata data structure, see Table E.7	Zero or one (optional) Include when useful ^e
valuesUnit ValuesUnit	Indicates that this quantity has units or reference system, and provides value used ^f	ValuesUnit data structure, see Table E.4	Zero or one (optional) Include when values have units or reference system
metadata Metadata	Additional metadata about domain of this quantity	Metadata data structure, see Table 23 of OGC 05-008	Zero or more (optional) One for each such metadata object ^g
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b For quantities that contain a list or sequence of values, these values shall be for individual values in the list.</p> <p>c Default value shall be string encoding of any value of another data type.</p> <p>d This Meaning can provide more specific, complete, precise, machine accessible, and machine understandable semantics about this quantity, relative to other available semantic information. For example, other semantic information is often provided in <documentation> elements in XML Schemas or <description> elements in GML objects.</p> <p>e This metadata should be referenced or included unless this information is clearly specified elsewhere.</p> <p>f Provides the identifier of the units or reference system used by the AllowedValues or ValuesListReference.</p> <p>g These metadata objects can be included in any order. A list of the required and/or optional metadata objects for each quantity should be specified in the Implementation Specification for a specific OWS service.</p>			

Table E.3 — Parts of PossibleValues data structure

Names^a	Definition	Data type	Multiplicity
allowedValues AllowedValues	List of all valid values and/or ranges of values for this quantity	AllowedValues data structure, see Table E.5	Zero or one (conditional) ^b
anyValue AnyValue	Specifies that any value is allowed for this quantity	Empty data structure	Zero or one (conditional) ^b
noValues NoValues	Specifies that no values are allowed for this quantity	Empty data structure	Zero or one (conditional) ^b
valuesReference ValuesReference	Reference to list of all valid values and/or ranges of values for this quantity	ValuesReference data structure, see Table E.8	Zero or one (conditional) ^b
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b One and only one of these four items shall be included.</p>			

Table E.4 — Parts of ValuesUnit data structure

Names ^a	Definition	Data type	Multiplicity
uom UOM	Identifier of unit of measure of this set of values	DomainMetadata data structure, see Table E.7	Zero or one (conditional) ^b
referenceSystem ReferenceSystem	Identifier of reference system used by this set of values	DomainMetadata data structure, see Table E.7	Zero or one (conditional) ^b

a See Table 1 for UML and XML naming conventions.
b One and only one of these items shall be included.

Table E.5 — Parts of AllowedValues data structure

Names ^a	Definition	Data type	Multiplicity and use
value Value	Value for this quantity	Character string type, not empty ^b	Zero or more (optional) One for each separate value ^c
range Range	Range of values of numeric parameter ^d	Range data structure, see Table E.6	Zero or more (optional) One for each separate range ^c

a See Table 1 for UML and XML naming conventions.
b Value shall be string encoding of any value of another data type.
c For numeric parameters, signed values shall be ordered from negative infinity to positive infinity.
d This range can be continuous or discrete, defined by a fixed spacing between adjacent valid values. Inclusion of the specified minimum and maximum values in the range shall be defined by the rangeClosure.

Table E.6 — Parameters in Range data structure

Names ^a	Definition	Data type and values	Multiplicity and use
minimumValue MinimumValue	Minimum value of this range of this numeric parameter	Character String, not empty ^b Default is negative infinity	Zero or one (optional) Include when not default
maximumValue MaximumValue	Maximum value of this range of this numeric parameter	Character String, not empty ^b Default is positive infinity	Zero or one (optional) Include when not default
spacing Spacing	Regular distance or spacing between allowed values in this range ^c	Character String, not empty ^b	Zero or one (optional) Include when range is not continuous
rangeClosure rangeClosure	Specifies which of minimum and maximum values are included in this range	Enumeration type, either: "closed" "open" "open-closed" "closed-open"	Zero or one (optional) Include when not default of "closed"

a See Table 1 for UML and XML naming conventions.
b Parameter value shall be string encoding of any value of another data type.
c This range may be continuous or discrete, defined by this fixed spacing between adjacent valid values.

Table E.7 — Parameters in DomainMetadata data structure

Names ^a	Definition	Data type	Multiplicity and use
name (anonymous)	Human-readable name of metadata described by referenced document	Character String, not empty ^b	One (mandatory)
reference reference	Reference to metadata about this domain	URI	Zero or one (optional) Include when available
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b Reference to metadata recorded elsewhere, either external to this XML document or within it. Whenever practical, this attribute with type anyURI should be a URL from which this metadata can be electronically retrieved. Alternately, this attribute can reference a definition URN for well-known metadata. For example, such a URN could be a URN defined in the "ogc" URN namespace.</p>			

NOTE 2 Possible definition URNs in the "ogc" URN namespace for data types are specified in Best Practices Paper OGC 06-023r1.

Table E.8 — Parameters in ValuesReference data structure

Names ^a	Definition	Data type	Multiplicity and use
name (anonymous)	Human-readable name of metadata described by referenced document	Character String, not empty	One (mandatory)
reference reference	Reference to list of all valid values and/or ranges of values for this quantity	URI ^b	Zero or one (optional) Include when available
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b Reference to metadata recorded elsewhere, either external to this XML document or within it. Whenever practical, this attribute with type anyURI should be a URL from which this metadata can be electronically retrieved. Alternately, this attribute can reference a URN for well-known metadata. For example, such a URN could be a URN defined in the "ogc" URN namespace.</p>			

E.2 Domain typed parameter encoding

NOTE No KVP encoding of domain typed parameters is now specified, since KVP encoding is probably impractical.

The XML Schema for encoding domain type metadata shall be as specified in the attached file owsDomainType.xsd. That Schema uses the owsCommon.xsd schema, also attached.

EXAMPLE A XML document fragment using the ows:DomainType in the ows:Parameter element specified in Subclause 7.4.5 of [OGC 05-008] is:

```
<Parameter name="Length">
  <AllowedValues>
    <Value>1.0</Value>
    <Range>
      <MinimumValue>4.0</MinimumValue>
      <MaximumValue>7.0</MaximumValue>
    </Range>
    <Value>10.0</Value>
    <Range>
      <MinimumValue>15.0</MinimumValue>
      <MaximumValue>17.0</MaximumValue>
    </Range>
  </AllowedValues>
  <Meaning>TBD definition of parameter.</Meaning>
  <DataType
ows:reference="urn:ogc:def:dataType:OGC:0.0:Double">Double</DataType>
  <UOM ows:reference="urn:ogc:def:uom:OGC:1.0:metre">metre</UOM>
```

```
<Metadata xlink:href="http://my.service.net/metadata?123456"/>  
</Parameter>
```

Annex F (normative)

SOAP transfer

All compliant WCS servers may implement SOAP 1.2 transfer of all WCS operation requests and responses, using the XML encodings specified in the body of this document. When SOAP is implemented, the SOAP Request-Response message exchange pattern shall be used with the HTTP POST binding.

For SOAP transfer, each XML-encoded operation request shall be encapsulated in the body of a SOAP envelope, which shall contain only a body and only this request in that body. Similarly, each XML-encoded operation response shall be encapsulated in the body of a SOAP envelope, which shall contain only a body and only this response in that body. A WCS server shall return operation responses and error messages using only SOAP transfer when the operation request is sent using SOAP.

All compliant WCS servers shall specify the URLs to which SOAP operation requests may be sent, within the OperationsMetadata section of a service metadata (Capabilities) XML document, as specified in Subclause 8.3.2.

When an error is detected while processing an operation request encoded in a SOAP envelope, the WCS server shall generate a SOAP response message where the content of the Body element is a Fault element containing an ExceptionReport element. This shall be done using the following XML fragment:

```
<soap:Envelope xmlns:soap="http://www.w3.org/2003/05/soap-envelope">
  <soap:Body>
    <soap:Fault>
      <soap:Code>
        <soap:Value>soap:Server</soap:Value>
      </soap:Code>
      <soap:Reason>
        <soap:Text>A server exception was encountered.</soap:Text>
      </soap:Reason>
      <soap:Detail>
        <ows:ExceptionReport>
          ...
        </ows:ExceptionReport>
      </soap:Detail>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

The Code element shall have the Value “soap:server” indicating that this is a server exception. The Reason element shall have the Text “Server exception was encountered.” This fixed string is used since the details of the exception shall be specified in the Detail element using an ows:ExceptionReport element as specified in OWS Common [OGC 05-008].

Annex G (normative)

GridCRS

G.1 Introduction

This annex specifies a GridCRS data structure for encoding the definitions of rectangular grids defined within another CRS. This annex also specifies three instances of the `gml:OperationMethod` element that specify rectangular grids (more precisely, parallelogram grids in the other CRS). Such a `gml:OperationMethod` is referenced by the `GridType` that is included in the specified GridCRS data structure.

NOTE This GridCRS data structure is specified here because it is now only used in WCS. However, this GridCRS is likely to be useful in other OWS specifications as well, such as the draft Web Coordinate Transformation Service (WCTS) [OGC 05-013]. When or before this GridCRS is used elsewhere, the GridCRS specification in this Annex should be moved into a separate OGC Implementation Specification.

G.2 GridCRS

G.2.1 GridCRS contents

This subclause specifies a GridCRS data structure for encoding the definition of a rectangular grid that is specified within another CRS, called the `GridBaseCRS`. This GridCRS data structure is simplified from the `DerivedCRS` element that is defined in GML 3.1.1, to define such a rectangular grid in the `GridBaseCRS`.

NOTE 1 The GridCRS XML element can be referenced remotely as the source or target CRS of a `CoordinateOperation` element. However, it is NOT defined by a GML Application Schema, because such a definition would be too complex for use in WCS. This GridCRS element is thus NOT in the `gml:_CRS` substitutionGroup, and so it cannot be included as the baseCRS of a `DerivedCRS`, or included in a GML dictionary. However, this GridCRS element does use a few parts of GML, which are included in the approved GML 3.1.1 CRS support profile [OGC 05-094r1].

The parameters contained in the GridCRS data structure are described in Figure G.1 and specified in Table G.1.

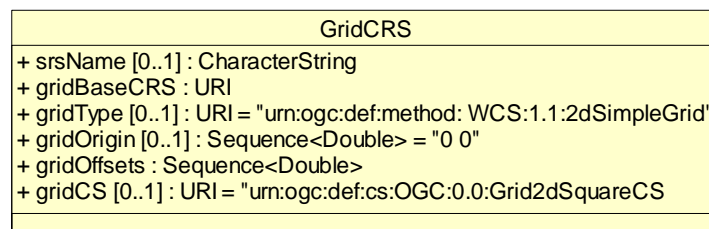


Figure G.1 — GridCRS class diagram

Table G.1 — Parameters in GridCRS data structure

Names ^a	Definition	Data type and value	Multiplicity and use
srsName srsName	Name or identifier of this GridCRS	Character string type, not empty	Zero or one (optional) Include when needed so this GridCRS can be referenced
gridBaseCRS GridBaseCRS	Reference to definition of base CRS used by this GridCRS	URI ^b	One (mandatory)
gridType GridType	Reference to definition of operation method that specifies quadrilateral grid in GridBaseCRS	URI Default is “urn:ogc:def:method: WCS:1.1:2dSimpleGrid”	Zero or one (optional) Include when not default GridType
gridOrigin GridOrigin	Position coordinates of origin of this GridCRS in the GridBaseCRS	Sequence<Double> Default is “0.0 0.0”	Zero or one (optional) Include when not default origin
gridOffsets GridOffsets	Position offsets between adjacent points in rectangular grid, in each dimension of GridCS in each dimension of GridBaseCRS	Sequence<Double>	One (mandatory)
gridCS GridCS	Reference to definition of 2D or 3D CartesianCS used by this GridCRS	URI Default is “urn:ogc:def:cs:OGC: 0.0:Grid2dSquareCS”	Zero or one (optional) Include when not default GridCS

a See Table 1 for UML and XML naming conventions.

b The referenced GridBaseCRS can be a 2D or 3D spatial CRS, most frequently 2D. This baseCRS can be a 2D gml:ProjectedCRS, a 2D or 3D gml:GeographicCRS, or a gml:CompoundCRS that combines a 2D gml:ProjectedCRS with a 1D gml:VerticalCRS, as specified in the GML 3.1.1 common CRSs profile [OGC 05-095r1]. This GridBaseCRS can alternately be a 2D or 3D gml:ImageCRS or another wcs:GridCRS. The two axes of a gml:GeographicCRS are usually named geodetic latitude and geodetic longitude, with coordinates listed in that order and with both latitude and longitude values given in decimal degrees. The two axes of a gml:ProjectedCRS are often named Easting and Northing, not always in that order, with values often given in metres.

In this GridCRS, the GridType remotely references one of several fixed XML documents that each specifies an OperationMethod object. There are many OperationMethod objects that might be referenced, three of which are specified in Subclause G.3.

G.2.2 Default values

As indicated above, the definition of a GridCRS contains five primary parts. However, two of these parts will often have the same values. Those same values will be the values needed for the most common case, namely a 2D grid defined in a 2D CRS, without grid rotation or skewing. Furthermore, when included in the output part of a GetCoverage operation request, a third part will often have the same value, namely a “0.0 0.0” origin position. Default values are thus specified for these three parts, to allow omission of specific values for these parts in the most common cases. These default values are as specified in Table G.2.

Table G.2 — GridCRS default values

GridCRS part	Default value (XML encoded)
GridType	<code><GridType>urn:ogc:def:method:WCS:1.1:2dSimpleGrid</GridType></code> ^a
GridOrigin	<code><GridOrigin>0.0 0.0</GridOrigin></code>
GridCS	<code><GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS></code> ^b
a	This URN and its meaning are specified in Subclause G.3.4.
b	This URN and its meaning are specified in Subclause 10.4 of OGC Best Practices Paper [OGC 05-096r1].

G.2.3 XML encoding

The definition of a GridCRS will often be encoded in XML. This XML encoding shall use the XML Schema fragment specified in `wcs:GridCRS.xsd`.

EXAMPLE 1 An example XML fragment defining a 2D GridCRS without grid rotation or skewing is:

```
<GridCRS>
  <gml:srsName>2D Grid CRS Example 1</gml:srsName>
  <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:63266405</GridBaseCRS>
  <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS</GridType >
  <GridOrigin>1.0 2.0</GridOrigin>
  <GridOffsets>0.1 0 0 0.2</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS >
</GridCRS>
```

NOTE 1 In the above example, the center two Offsets values are zero because there is no grid rotation or skewing relative to the underlying CRS axes (meridians and parallels). These offset values define a grid with a point spacing of 0.1 x 0.2 degree (latitude x longitude, as defined by EPSG 63266405)

EXAMPLE 2 An example XML fragment defining a 2D GridCRS with rotation (but no skewing) in the baseCRS is:

```
<GridCRS>
  <gml:srsName>2D Grid CRS Example 2</gml:srsName>
  <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS >
  <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS</GridType>
  <GridOrigin>1.0 2.0</GridOrigin>
  <GridOffsets>0.0707 -0.0707 0.1414 0.1414</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

NOTE 2 In the above example, the center two Offsets values are non-zero because there is 45 degrees grid rotation clockwise relative to the easting/northing axes of the GridBaseCRS (WGS84 UTM meters, Zone 18N – denoted EPSG 32618). The grid defined by these offsets has a point spacing of 0.1x0.2 meter.

NOTE 3 These two example XML fragments reference the fixed XML document that specifies the `2dGridIn2dCrS` given in Subclause G.3.2.

G.2.4 KVP encoding

This GridCRS may be KVP encoded in a GetCoverage operation request. That KVP encoding shall be as specified in Table G.3.

Table G.3 — GridCRS encoded as Key Value Pairs

Name and example	Optionality & use	Definition and format
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618	Optional, include all five GridCRS parameters, except for defaulted parameters, to request output in a CRS other than the ImageCRS of stored coverage	Reference to baseCRS of desired output GridCRS, a URN or URL ^a
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS		Reference to grid type of desired output GridCRS, a URN or URL ^{a, b}
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS		Reference to coordinate system of desired output GridCRS, a URN or URL ^{a, c}
GridOrigin=1.0,2.0		Position coordinates of one possible grid origin, in baseCRS of desired output GridCRS ^{a, d}
GridOffsets=0.0707,-0.0707,0.1414,0.1414&		Offsets between adjacent grid points, in baseCRS of desired output GridCRS ^{a, e}
<p>a The GridType parameter shall reference the known definition of a coordinate OperationMethod that defines a grid in the referenced GridBaseCRS. That GridType definition shall specify the meanings of the other “Grid...” parameters listed, including the number of dimensions in each other “Grid...” parameter.</p> <p>b This GridType may be omitted when the GridBaseCRS is 2D and the grid is not rotated or skewed in the GridBaseCRS, in which case the default value shall be “urn:ogc:def:method:WCS:1.1:2dSimpleGrid”.</p> <p>c This GridCS may be omitted when the GridBaseCRS is 2D (in a 2D or 3D GridBaseCRS), in which case the default value shall be “urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS”.</p> <p>d This GridOrigin may be omitted when the GridBaseCRS is 2D, in which case the default value shall be “0.0 0.0”.</p> <p>e When the GridType is “urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS”, there shall be four values in this GridOffsets. The center two of these offsets will be zero when the GridCRS is not rotated or skewed in the GridBaseCRS.</p>		

NOTE 1 This KVP encoding does not include the srsName of the GridCRS, since KVP encoding is only used in the Output part of a GetCoverage operation request, and any GridCRS defined there is never referenced.

EXAMPLE 1 Here is an example 2D GridCRS definition without grid rotation or skewing, defined using KVP encoding:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridType=urn:ogc:def:method:WCS:1.1:2dSimpleGrid&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0&
GridOffsets=0.1,0.2&
```

NOTE 2 This example and the next define a grid in the EPSG:63266405 CRS, that is, WGS84 latitude and longitude expressed in decimal degrees. The grid spacing is 0.1 degree of latitude and 0.2 degree of longitude.

EXAMPLE 2 Here is the same example GridCRS definition, taking advantage of all three applicable defaults:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridOffsets=0.1,0.2&
```

EXAMPLE 3 Here is an example 2D GridCRS with 45 degrees clockwise rotation (but no skewing) in the GridBaseCRS, defined using KVP encoding:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

NOTE 3 This example and the next define a grid in the EPSG:32618 CRS, that is, WGS84 UTM Zone 18 North easting and northing expressed in meters. The grid spacing is 0.1m (NW-SE) and 0.2m (SW-NE)

EXAMPLE 4 Here is the same GridCRS definition, taking advantage of the two applicable defaults:

```

GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&

```

G.3 Operation methods for rectangular grids

G.3.1 Introduction

This subclause specifies instances of the `gml:OperationMethod` element that specify rectangular grids in a manner similar to the `gml:RectifiedGrid` element. As previously stated, such a `gml:OperationMethod` is expected to normally be used by the `GridType` that is used by the specified `GridCRS` element. However, neither these `gml:OperationMethods` nor the `GridCRSs` using them define the extent of the grid (e.g., by extending `gml:Grid` as `gml:RectifiedGrid` does).

G.3.2 2D grid in 2D CRS (urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs)

The following `OperationMethod` specifies a 2D grid that can be rotated or skewed within a 2D CRS:

```

<?xml version="1.0" encoding="UTF-8"?>
<OperationMethod xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gml
../../../../gml/3.1.1/Profiles/GridCRSs/1.0.0/gmlGridCRSsProfile.xsd"
gml:id="2dGridIn2dCrs">
  <!-- Last updated 2006-02-27 -->
  <!-- ===== -->
  <methodName>2D Rectified Grid in 2D CRS Method</methodName>
  <methodID>
    <name>urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs</name>
  </methodID>
  <remarks>Method for use by a Coordinate Conversion defining a 2D grid
CRS in a 2D baseCRS, based on CV_RectifiedGrid in ISO 19123 so the grid
can be rotated and skewed in this baseCRS. </remarks>
  <methodFormula>The reverse direction (from the grid CRS to the
baseCRS) shall be as specified by the following equations:
  BaseX = origin(1) + offsets(1,1) * GridX + offsets(1,2) * GridY
  BaseY = origin(2) + offsets(2,1) * GridX + offsets(2,2) * GridY
where
  GridX, GridY are position coordinates in the grid 2D CRS
  BaseX, BaseY are position coordinates in the 2D base CRS
  origin(1), origin(2) are the two coordinates of the grid origin
position in the 2D base CRS
  offsets(1,1), offsets(1,2) are the grid point offsets of the first
grid axis in the 2D base CRS
  offsets(2,1), offsets(2,2) are the grid point offsets of the second
grid axis in the 2D base CRS.

The forward direction (from the baseCRS to the GridCRS) equations are not
included here. </methodFormula>
  <sourceDimensions>2</sourceDimensions>
  <targetDimensions>2</targetDimensions>
  <!-- ===== -->
  <usesParameter>
    <OperationParameter gml:id="origin2d">
      <parameterName>origin2d</parameterName>
      <parameterID>

```

```

        <name>urn:ogc:def:parameter:WCS:1.1:origin2d</name>
    </parameterID>
    <remarks>This parameter shall be encoded as a wcs:Origin element,
giving the two origin coordinates in the 2D base CRS of this GridCRS.
</remarks>
</OperationParameter>
</usesParameter>
<!-- ===== -->
<usesParameter>
    <OperationParameter gml:id="offsets2d">
        <parameterName>offset2d</parameterName>
        <parameterID>
            <name>urn:ogc:def:parameter:WCS:1.1:offsets2d</name>
        </parameterID>
        <remarks>This parameter shall be encoded as a wcs:Offsets
element, giving the grid offsets in the two grid axes, each in the 2D
base CRS relative to the specified origin. This wcs:Offsets shall contain
four values, the first two values shall specify the grid offset for the
first grid axis, and the second pair of values shall specify the grid
offset for the second grid axis. </remarks>
        </OperationParameter>
    </usesParameter>
</OperationMethod>

```

This “2D grid in 2D CRS method” shall be referenced using the above-specified URN “**urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs**”. Several examples in Subclauses G.2.2 and G.2.3 above reference this OperationMethod.

G.3.3 2D grid in 3D CRS (urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs)

The following OperationMethod specifies a 2D grid within a 3D CRS:

```

<?xml version="1.0" encoding="UTF-8"?>
<OperationMethod xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/gml
http://schemas.opengis.net/gml/3.1.1/profiles/GridCRSs/1.0.0/gmlGridCRSSp
rofile.xsd" gml:id="2dGridIn3dCrs">
    <!-- Last updated 2006-02-28 -->
    <!-- ===== -->
    <methodName>2D Rectified Grid in 3D CRS Method</methodName>
    <methodID>
        <name>urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs</name>
    </methodID>
    <remarks>Method for use by a Coordinate Conversion defining a 2D grid
CRS in a 3D baseCRS, based on CV_RectifiedGrid in ISO 19123 so the grid
can be rotated and skewed in this baseCRS. </remarks>
    <methodFormula>The reverse direction (from the grid CRS to the
baseCRS) shall be as specified by the equations:
    BaseX = origin(1) + offsets(1,1) * GridX + offsets(2,1) * GridY
    BaseY = origin(2) + offsets(1,2) * GridX + offsets(2,2) * GridY
    BaseZ = origin(3) + offsets(1,3) * GridX + offsets(2,3) * GridY
    where
    GridX, GridY are position coordinates in the 2D grid CRS
    BaseX, BaseY, BaseZ are position coordinates in the 3D base CRS
    origin(1), origin(2), origin(3) are the three coordinates of the grid
origin position in the 3D base CRS
    offsets(1,1),offsets(1,2), offsets(1,3) are the grid spacings of the
first grid axis in the 3D base CRS

```

offsets(2,1), offsets(2,2), offsets(2,3) are the grid spacings of the second grid axis in the 3D base CRS

The forward direction (from the baseCRS to the GridCRS) equations are not included here. `</methodFormula>`

```

<sourceDimensions>3</sourceDimensions>
<targetDimensions>2</targetDimensions>
<!-- ===== -->
<usesParameter>
  <OperationParameter gml:id="origin3d">
    <parameterName>origin3d</parameterName>
    <parameterID>
      <name>urn:ogc:def:parameter:WCS:1.1:origin3d</name>
    </parameterID>
    <remarks>This parameter shall be encoded as a wcs:Origin element,
giving the three origin coordinates in the 3D base CRS of this GridCRS.
</remarks>
  </OperationParameter>
</usesParameter>
<!-- ===== -->
<usesParameter>
  <OperationParameter gml:id="offset3d">
    <parameterName>offset3d</parameterName>
    <parameterID>
      <name>urn:ogc:def:parameter:WCS:1.1:offset3d</name>
    </parameterID>
    <remarks>This parameter shall be encoded as a wcs:Offsets
element, giving the grid offsets in the two grid axes, each in the 3D
base CRS relative to the specified origin. This wcs:Offsets shall contain
six values, the first three values shall specify the grid offset for the
first grid axis, and the second three values shall specify the grid
offset for the second grid axis. </remarks>
  </OperationParameter>
</usesParameter>
</OperationMethod>

```

This “2D grid in 3D CRS method” shall be referenced using the above-specified URN “urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs”.

EXAMPLE 1 Here is an example GridCRS that references this OperationMethod encoded in XML:

```

<?xml version="1.0" encoding="UTF-8"?>
<GridCRS xmlns="http://www.opengis.net/wcs/1.1"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGridCRS.xsd"
  gml:id="example2dGridCrs2">
  <gml:srsName>2D Grid CRS Example 3</gml:srsName>
  <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:7405</GridBaseCRS>
  <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn3dCrs</GridType >
  <GridOrigin>0 0 0</GridOrigin>
  <GridOffsets>0.0707 0.0707 0 -0.1414 0.1414 0</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS >
</GridCRS>

```

NOTE This example and the next define a grid in the EPSG:7405 CRS (a 3D, Cartesian reference that uses east / north meters in the British National Grid, and vertical meters relative to the Ordnance Survey Datum). The six numbers in GridOffsets define a grid that is level to the ground plane but rotated at 45 degrees counterclockwise relative to the easting / northing axes, with a grid spacing of 0.1m (in the SW-NE direction) and 0.2m (NW-SE).

EXAMPLE 2 The same example GridCRS encoded in KVP is:

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:7405&
```

```

GridType=urn:ogc:def:method:WCS:1.1:2dGridIn3dCr&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0,0&
GridOffsets=0.0707,0.0707,0,-0.1414,0.1414,0&

```

EXAMPLE 3 Here is the same example GridCRS encoded in KVP using the default GridCS and GridOrigin:

```

GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:7405&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn3dCr&
GridOffsets=0.0707,0.0707,0,-0.1414,0.1414,0&

```

G.3.4 2D simple grid (urn:ogc:def:method:WCS:1.1:2dSimpleGrid)

The following OperationMethod specifies a 2D simple grid (with no rotation or skew) within a 2D CRS:

```

<?xml version="1.0" encoding="UTF-8"?>
<OperationMethod xmlns="http://www.opengis.net/gml"
xmlns:gml="http://www.opengis.net/gml"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:schemaLocation="http://www.opengis.net/gml
../../../../gml/3.1.1/Profiles/GridCRSs/1.0.0/gmlGridCRSsProfile.xsd"
gml:id="2dSimpleGrid">
  <!-- Last updated 2006-08-28 -->
  <!-- ===== -->
  <methodName>2D Simple Grid Method</methodName>
  <methodID>
    <name>urn:ogc:def:method:WCS:1.1:2dSimpleGrid</name>
  </methodID>
  <remarks>Method for use by a Coordinate Conversion defining a simple
  2D GridCRS in a 2D baseCRS, where this grid is not rotated or skewed in
  this baseCRS. </remarks>
  <methodFormula>The reverse direction (from the grid CRS to the
  baseCRS) shall be as specified by the equations:
  BaseX = origin(1) + offsets(1) * GridX
  BaseY = origin(2) + offsets(2) * GridY
  Where
  GridX, GridY are position coordinates in the grid 2D CRS
  BaseX, BaseY are position coordinates in the 2D base CRS
  origin(1), origin(2) are the two coordinates of the grid origin
  position in the 2D base CRS
  offsets(1) is the grid point offsets of the first grid axis in the 2D
  base CRS
  offsets(2) is the grid point offsets of the second grid axis in the
  2D base CRS

```

The forward direction (from the baseCRS to the GridCRS) equations are not included here. </methodFormula>

```

<sourceDimensions>2</sourceDimensions>
<targetDimensions>2</targetDimensions>
  <!-- ===== -->
  <usesParameter>
    <OperationParameter gml:id="origin2d">
      <parameterName>origin2d</parameterName>
      <parameterID>
        <name>urn:ogc:def:parameter:WCS:1.1:origin2d</name>
      </parameterID>
      <remarks>This parameter shall be encoded as a wcs:Origin
  element, giving the two origin coordinates in the 2D base CRS of this
  GridCRS. </remarks>
    </OperationParameter>

```

```

</usesParameter>
<!-- ===== -->
<usesParameter>
  <OperationParameter gml:id="offsets2d">
    <parameterName>offsets2d</parameterName>
    <parameterID>
      <name>urn:ogc:def:parameter:WCS:1.1:offsets2d</name>
    </parameterID>
    <remarks>This parameter shall be encoded as a wcs:Offsets
element, giving the grid offsets in the two grid axes, relative to the
specified origin. This wcs:Offsets shall contain two values, the first
value shall specify the grid offset for the first grid axis in the first
baseCRS axis, and the second value shall specify the grid offset for the
second grid axis in the second baseCRS axis. </remarks>
  </OperationParameter>
</usesParameter>
</OperationMethod>

```

This “2D simple grid method” shall be referenced using the above-specified URN “**urn:ogc:def:method:WCS:1.1:2dSimpleGrid**”.

EXAMPLE 1 Here is an example GridCRS that references this OperationMethod encoded in XML. (This example and the next define a grid in the EPSG:63266405 CRS, that is, WGS84 latitude and longitude expressed in decimal degrees.)

```

<?xml version="1.0" encoding="UTF-8"?>
<GridCRS xmlns="http://www.opengis.net/wcs/1.1"
xmlns:gml="http://www.opengis.net/gml" xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xmlns:schemaLocation="http://www.opengis.net/wcs/1.1 ../wcsGridCRS.xsd"
gml:id="example2dGridCrs1">
  <gml:srsName>2D Grid CRS Example 1</gml:srsName>
  <gml:baseCRS xlink:href="urn:ogc:def:crs:EPSG:6.6:63266405"/>
  <GridType>urn:ogc:def:method:WCS:1.1:WCS:1.1:2dSimpleGrid</GridType>
  <GridOrigin>0 0</GridOrigin>
  <GridOffsets>0.1 0.2</GridOffsets>
  <gridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</gridCS >
</GridCRS>

```

NOTE This 2D simple grid method uses a GridOffsets (KVP or XML) parameter containing only two values: the grid spacing along each of its axes, expressed in the units and the order given by the underlying CRS (in this case, degrees of latitude and longitude, in that order). Those two values correspond to the RESX and RESY parameters used in the GetCoverage request KVP encoding in WCS 1.0.

EXAMPLE 2 Here is the same example GridCRS encoded in KVP:

```

GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridType=urn:ogc:def:method:WCS:1.1:WCS:1.1:2dSimpleGrid&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=0,0&
GridOffsets=0.1,0.2&

```

EXAMPLE 2 Here is the same example GridCRS encoded in KVP, taking advantage of all three available defaults:

```

GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:63266405&
GridOffsets=0.1,0.2&

```

Annex H (informative)

GetCoverage use cases

H.1 Introduction

This annex describes a set of use cases for WCS GetCoverage operation requests and responses, followed by descriptions of how this WCS version can be used to satisfy these use cases. These use cases were developed and used by the WCS 1.1 RWG, and consider requesting spatial subsets of grid coverages, but do not discuss temporal subsets.

H.2 Use cases

H.2.1 Overview

These use cases are based on three conditions affecting what a server is required to do to respond to a GetCoverage operation request. These conditions may or may not be significant to the client. These three conditions are:

- a) Is resampling of the stored coverage needed?
- b) Is the stored coverage georectified or unrectified?
- c) Is the desired spatial subset extent defined in ground (or object) or image coordinates?

These 10 use cases are summarized in Table H.1, assigning case numbers for all possible combinations of these three conditions.

Table H.1 — GetCoverage use case numbers for combinations of conditions

Stored coverage type Extent CRS type	Is resampling needed?	
	No resampling	Resampling
Georectified coverage		
Extent in ground CRS	1	6
Extent in image CRS	2	7
Unrectified and un-georeferenced coverage		
Extent in image CRS ^a	3	8
Unrectified and georeferenced coverage		
Extent in image CRS	4	9
Extent in ground CRS	5	10
a The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These use cases also apply when the georeferencing coordinate transformation(s) are not available to the server, cannot be used by the server, or are not needed to satisfy the GetCoverage request.		

In order to allow a client to not know anything more than whether it is defining the extent in a ground or image CRS, the GetCoverage request design for these use cases should use the same request data structures for ALL supported use cases. For example, the GetCoverage

request design should allow including either a ground or image CRS, of one subtype or another, in exactly the same places for ALL supported use cases.

In all these use cases, the server is required to be able to output metadata with the coverage, where all georeferencing information included within an image file is considered to be metadata.

The following subclauses further describe each use case.

H.2.2 Use case 1: No resampling of georectified coverage, defining extent in ground CRS

The GetCoverage request asks a server for an un-resampled subsection of a georectified image or other georectified grid coverage, defining the desired extent in a 2D ground CRS (GeographicCRS or ProjectedCRS). Notice that the specified extent could be skewed and with curved edges in the stored coverage. When such a skewed/curved extent is allowed in the stored coverage, the WCS server should interpret this extent as the minimum extent desired, and should extend that extent to the minimum rectangle surrounding that desired skewed/curved extent.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) CRS of output coverage
- c) Extent of output coverage

H.2.3 Use case 2: No resampling of georectified image, defining extent in image CRS

The GetCoverage request asks a server for an un-resampled subsection of a georectified image, defining the desired extent in an ImageCRS. Notice that the specified extent could be skewed and with curved edges in the stored coverage. When such a skewed/curved extent is allowed in the stored coverage, the WCS server should interpret this extent as the minimum extent desired, and should extend that extent to the minimum rectangle surrounding that desired skewed/curved extent.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) CRS of output coverage
- c) Extent of output coverage

H.2.4 Use case 3: No resampling of unrectified and un-georeferenced image

NOTE This use case also applies when the georeferencing coordinate transformation(s) are not available to the server, cannot be used by the server, or are not needed to satisfy the GetCoverage request.

The GetCoverage request asks a server for an un-resampled subsection of an unrectified and un-georeferenced image, defining the desired extent in an ImageCRS.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) Extent of output image
- c) Image CRS of output image, at least identification of this ImageCRS definition, for possible future use in georeferencing coordinate Transformation(s)

H.2.5 Use case 4: No resampling of unrectified but georeferenced image, defining extent in image CRS

The GetCoverage request asks a server for an un-resampled subsection of an unrectified but georeferenced image, defining the desired extent in an Image CRS.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) Image CRS of output image, at least identification of this ImageCRS definition, for use in georeferencing coordinate Transformation(s)
- c) Extent of output image
- d) Georeferencing coordinate Transformation(s), between identified Image CRS of output image subsection and same Ground CRS of georeferencing coordinate Transformation(s) for stored unrectified image

H.2.6 Use case 5: No resampling of unrectified but georeferenced image, defining extent in ground CRS

The GetCoverage request asks a server for an un-resampled subsection of an unrectified but georeferenced image, defining the desired extent in some 2D or 3D ground CRS (GeographicCRS, ProjectedCRS, or either compounded with a VerticalCRS). Notice that the specified extent could be skewed and with curved edges in the stored coverage. When such a skewed/curved extent is allowed in the stored coverage, the WCS server should interpret this extent as the minimum extent desired, and should extend that extent to the minimum rectangle surrounding that desired skewed/curved extent.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) Image CRS of output image, at least identification of this ImageCRS definition, for use in georeferencing coordinate Transformation(s)
- c) Extent of output image
- d) Georeferencing coordinate Transformation(s), between identified Image CRS of output image subsection and same Ground CRS of georeferencing coordinate Transformation(s) for stored unrectified image

H.2.7 Use case 6: Resampling of georectified coverage, defining extent in ground CRS

The GetCoverage request asks a server for a resampled (differently georectified) subsection of a georectified image or other georectified grid coverage, defining the desired extent in a 2D ground CRS (GeographicCRS or ProjectedCRS). See Addition A below.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) CRS of output coverage
- c) Extent of output coverage

H.2.8 Use case 7: Resampling of georectified image, defining extent in image CRS

The GetCoverage request asks a server for a resampled (differently georectified) subsection of a georectified image, defining the desired extent in an ImageCRS. See Addition A below.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) CRS of output coverage
- c) Extent of output coverage

H.2.9 Use case 8: Resampling of unrectified and un-georeferenced image

NOTE This use case also applies when the georeferencing coordinate transformation(s) are not available to the server, cannot be used by the server, or are not needed to satisfy the GetCoverage request.

The GetCoverage request asks a server for a resampled subsection of an un-georeferenced (and unrectified) image, defining the desired extent in an ImageCRS. See Addition A below.

The needed response from the GetCoverage operation includes all:

- a) Array of output coverage grid/pixel values
- b) Extent of output image
- c) Image CRS of output image, at least identification of this ImageCRS definition, for possible future use in georeferencing coordinate Transformation(s)

H.2.10 Use case 9: Resampling of unrectified but georeferenced image, defining extent in image CRS

The GetCoverage request asks a server for a resampled (rectified) subsection of an unrectified but georeferenced image, defining the desired extent in an Image CRS. See Addition A below.

The needed response from the GetCoverage operation includes all:

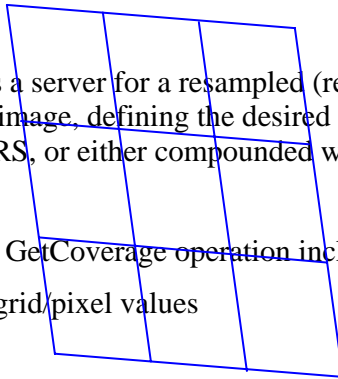
- a) Array of output coverage grid/pixel values
- b) CRS of output image
- c) Image CRS of output image, at least identification of this ImageCRS definition

H.2.11 Use case 10: Resampling of unrectified but georeferenced image, defining extent in ground CRS

The GetCoverage request asks a server for a resampled (rectified) subsection of an unrectified but georeferenced image, defining the desired extent in a 2D or 3D Ground CRS (GeographicCRS, ProjectedCRS, or either compounded with a VerticalCRS). See Addition A below.

The needed response from the GetCoverage operation includes all:

- Array of output coverage grid/pixel values
- CRS of output image
- Image CRS of output image, at least identification of this ImageCRS definition



H.2.12 Addition A

The following addition applies to uses cases 6 through 10 listed above

The “differently georectified” or “rectified” or “resampled” output grid requested must be able to be resampled into a client-defined grid that is rotated and skewed, with client-defined grid spacings and origin position, with respect to the ground base CRS for that grid. This client-defined grid in a 3D ground base CRS could be specified by a rectified grid as specified in Figure 14 and CV_RectifiedGrid in Clause 8 of ISO 19123. Figure 14 in ISO 19123 represents a quadrilateral (a parallelogram in general, not necessarily a square or rectangle) grid in a plane within the 3D XYZ space, where the axes of the grid can be at any angles in the 3D XYZ space. A 2D version of that Figure 14 is shown in Figure H.1.

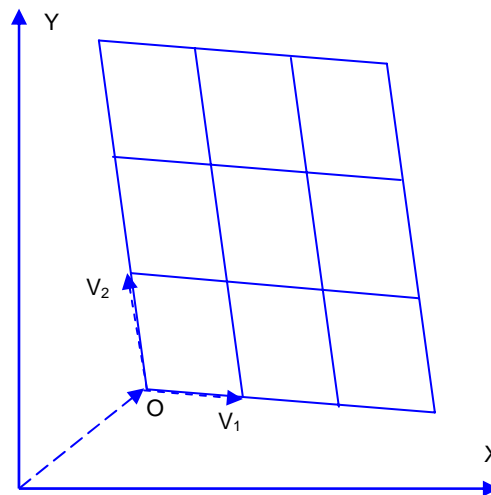


Figure H.1 — Parallelogram grid within a 2D X-Y space

NOTE This client-defined grid in a 2D ground base CRS could be specified by EPSG coordinate operation method code 9623, which is specified in Subclause 2.4.2.2 and Figure 8 in EPSG Guidance Note 7.

Notice that the specified extent will often be skewed with curved edges in the desired resampled output coverage. A specified skewed and curved extent in the desired resampled

output coverage should be interpreted as the minimum extent desired, and the server should extend this extent to the minimum rectangle surrounding that desired skewed/curved extent.

The squared-off or original rectangle can be transformed into the corresponding extent in the stored georectified coverage. Notice that the transformed extent will often be skewed with curved edges in the stored georectified coverage. When the transformed skewed/curved extent does not lie wholly within the stored georectified coverage, padding with null values is required at all output grid points falling outside the stored georectified coverage.

H.3 Handling of use cases

H.3.1 Overview

This subclause describes how the 10 use cases described above can be handled by the GetCoverage operation design specified in Clause 10. For each use case, Table H.2 lists the type of CRS that can be referenced in the BoundingBox in the DomainSubset part of a GetCoverage operation request.

Table H.2 — GetCoverage request DomainSubset BoundingBox CRS for use cases

Is resampling needed? Stored coverage type Extent CRS type used	Use case	CRS referenced in BoundingBox in DomainSubset part of GetCoverage request
No resampling		
Georectified coverage	1	any ground CRS ^b
Extent in ground CRS	2	ImageCRS of stored georectified coverage
Extent in image CRS		
Unrectified and un-georeferenced coverage	3	ImageCRS of stored unrectified coverage
Extent in image CRS ^a		
Unrectified and georeferenced coverage	4	ImageCRS of stored unrectified coverage
Extent in image CRS	5	any ground CRS ^b
Extent in ground CRS		
Resampling		
Georectified coverage	6	any ground CRS ^b
Extent in ground CRS	7	ImageCRS of stored georectified coverage
Extent in image CRS		
Unrectified and un-georeferenced coverage	8	ImageCRS of stored unrectified coverage
Extent in image CRS ^a		
Unrectified and georeferenced coverage	9	ImageCRS of stored unrectified coverage
Extent in image CRS	10	any ground CRS ^b
Extent in ground CRS		
<p>^a The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These two use cases also apply when the georeferencing coordinate transformation(s) are not available to the server, cannot be used by the server, or are not needed to satisfy the GetCoverage request.</p> <p>^b This DomainSubset BoundingBoxextent can be skewed with curved edges in the stored coverage.</p>		

For each use case, Table H.3 lists the type of CRS that can be included in the Output part of a GetCoverage request.

Table H.3 — GetCoverage request Output CRS for use cases

Is resampling needed? Stored coverage type Extent CRS type used	Use case	CRS in Output part of GetCoverage request
No resampling		
Georectified coverage		
Extent in ground CRS	1	no CRS or wcs:GridCRS of stored georectified coverage ^d
Extent in image CRS	2	no CRS or wcs:GridCRS of stored georectified coverage ^d
Unrectified and un-georeferenced coverage		
Extent in image CRS ^a	3	no CRS
Unrectified and georeferenced coverage		
Extent in image CRS	4	no CRS
Extent in ground CRS	5	no CRS
Resampling		
Georectified coverage		
Extent in ground CRS	6	any wcs:GridCRS that does not specify the same grid points as the wcs:GridCRS of stored georectified coverage ^b
Extent in image CRS	7	any wcs:GridCRS that does not specify the same grid points as the wcs:GridCRS of stored georectified coverage ^b
Unrectified and un-georeferenced coverage		
Extent in image CRS ^a	8	any wcs:GridCRS that is based on ImageCRS of stored unrectified coverage ^c
Unrectified and georeferenced coverage		
Extent in image CRS	9	any wcs:GridCRS
Extent in ground CRS	10	any wcs:GridCRS
<p>a The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These two use cases also apply when the georeferencing coordinate transformation(s) are not available to the server, cannot be used by the server, or are not needed to satisfy the GetCoverage request.</p> <p>b If only the origin in this output wcs:GridCRS is different from the GridCRS of the stored georectified coverage, but that origin is at a different grid point in the GridCRS of the stored georectified coverage, then this output GridCRS specifies the same grid positions and no resampling is needed.</p> <p>c No resampling is needed when this wcs:GridCRS, which is based on the ImageCRS of the stored georectified coverage, has parameters that define grid positions which exactly match the grid positions in that ImageCRS.</p> <p>d This use case will also be obtained when no GridCRS is specified in the Output part of the GetCoverage request.</p>		

For each use case, Table H.4 lists the type of CRS that can be referenced in the BoundingBox in the GetCoverage response.

Table H.4 — GetCoverage response BoundingBox CRS for use cases

Is resampling needed? Stored coverage type Extent CRS type used	Use case number	CRS referenced in BoundingBox in GetCoverage response
No resampling		
Georectified coverage Extent in ground CRS	1	wcs:GridCRS for output coverage that references ground baseCRS of stored georectified coverage ^b
Extent in image CRS	2	wcs:GridCRS for output coverage that references ground baseCRS of stored georectified coverage ^b
Unrectified and un-georeferenced coverage Extent in image CRS ^a	3	ImageCRS for output image ^b
Unrectified and georeferenced coverage Extent in image CRS	4	ImageCRS for output image ^{b, c}
Extent in ground CRS	5	ImageCRS for output image ^{b, c}
Resampling		
Georectified coverage Extent in ground CRS	6	wcs:GridCRS for output coverage based on same ground baseCRS and defining same grid positions as requested output wcs:GridCRS ^b
Extent in image CRS	7	wcs:GridCRS for output coverage based on same ground baseCRS and defining same grid positions as requested output wcs:GridCRS ^b
Unrectified and un-georeferenced coverage Extent in image CRS ^a	8	wcs:GridCRS for output coverage based on ImageCRS of stored georectified coverage and defining same grid positions as requested output wcs:GridCRS ^b
Unrectified and georeferenced coverage Extent in image CRS	9	wcs:GridCRS for output coverage based on same ground baseCRS and defining same grid positions as requested output wcs:GridCRS ^{b, c}
Extent in ground CRS	10	wcs:GridCRS for output coverage based on same ground baseCRS and defining same grid positions as requested output wcs:GridCRS ^{b, c}
<p>a The extent cannot be defined in a ground CRS for an unrectified and un-georeferenced coverage. These two use cases also apply when the georeferencing coordinate transformation(s) are not available to the server, cannot be used by the server, or are not needed to satisfy the GetCoverage request.</p> <p>b Included unless all this information is included in the coverage format used.</p> <p>c For this use case, the GetCoverage response shall also include georeferencing coordinate Transformation(s) for the ImageCRS of the output images.</p>		

The following subclauses further describe how each use case can be handled.

H.3.2 Use case 1: No resampling of georectified coverage, defining extent in ground CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the base CRS of stored georectified coverage). Any ground CRS can be used that the server can transform into the baseCRS of both the stored coverage and the desired output coverage. Notice that this DomainSubset BoundingBox extent can be rotated and skewed, without curved edges, in the stored georectified coverage.

NOTE 1 The two specified corners of this BoundingBox are not required to fall exactly on stored pixels in order to avoid resampling, since the Output part of the GetCoverage request specifies (or implies) the desired wcs:GridCRS of the output coverage.

The corners of this BoundingBox are first transformed into positions in the stored georectified coverage, which in this use case is the same wcs:GridCRS as the desired output coverage. When that extent is rotated and/or skewed (without curved edges) in the stored georectified coverage, a server interprets this extent as the minimum extent desired, and next extends that extent to the minimum rectangle surrounding that desired rotated/skewed extent. After that transformation and possible extension, the corners of this modified BoundingBox are extended to the next grid position in the stored georectified coverage. The actual output coverage uses the cell values from the source coverage (when they exist).

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
    <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
    <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405&
```

For this use case, the Output part of the GetCoverage request can specify either no CRS or the wcs:GridCRS of the stored georectified coverage (not the same CRS as referenced in the DomainSubset BoundingBox). When no output grid is specified, this is interpreted by servers as requesting the grid of stored georectified coverage.

Note: The origin of this wcs:GridCRS is NOT required or expected to be the actual position of the origin in the GetCoverage operation output, since that origin position is indirectly specified by the OWS BoundingBox in the DomainSubset.

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
</GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCr&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox, referencing the wcs:GridCRS for the output coverage that references the ground CRS of the stored georectified coverage (required unless all this information is included in the coverage format used)
- c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>7 7</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 2 In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123. The modified wcs:GridCRS origin is at (589, 172).

H.3.3 Use case 2: No resampling of georectified image, defining extent in image CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS of the stored georectified coverage. Notice that this extent cannot be skewed or with curved edges in the stored georectified image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID"
  dimension="2">
    <ows:LowerCorner>1000 2000</ows:LowerCorner>
    <ows:UpperCorner>1511 2511</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID&
```

For this use case, the Output part of the GetCoverage request can specify either no CRS or the wcs:GridCRS of the stored georectified coverage (not the same CRS as referenced in the DomainSubset BoundingBox). When no output grid is specified, this is interpreted by servers as requesting the grid of stored georectified coverage.

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
  </GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox, referencing the wcs:GridCRS for the output coverage that references the ground CRS of the stored georectified coverage (required unless all this information is included in the coverage format used)
- c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>511 511</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.4 Use case 3: No resampling of unrectified and un-georeferenced image

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS of the stored unrectified coverage. Notice that this extent cannot be skewed or with curved edges in the stored unrectified image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID"
  dimension="2">
    <ows:LowerCorner>1000 2000</ows:LowerCorner>
    <ows:UpperCorner>1511 2511</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID&
```

For this use case, the Output part of the GetCoverage request does not specify any wcs:GridCRS. The CRS used by the actual output coverage will be the ImageCRS of the stored image usually with different origin point or pixel.

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox (XML encoded), referencing the ImageCRS for output image (required unless all this information is included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="imageCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>511 511</ows:UpperCorner>
</ows:BoundingBox>
```

NOTE In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123. The wcs:GridCRS origin is at (1000, 2000).

H.3.5 Use case 4: No resampling of unrectified but georeferenced image, defining extent in image CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS of the stored georectified coverage. Notice that this extent cannot be skewed or with curved edges in the stored unrectified image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID"
  dimension="2">
    <ows:LowerCorner>1000 2000</ows:LowerCorner>
    <ows:UpperCorner>1511 2511</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID&
```

For this use case, the Output part of the GetCoverage request does not specify any wcs:GridCRS. The CRS used by the actual output coverage will be the ImageCRS of the stored image, usually with different origin point or pixel.

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox (XML encoded), referencing the ImageCRS of output image (required unless all this information is included in the coverage format used)
- c) Mandatory georeferencing coordinate Transformation(s), referencing the ImageCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="imageCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>511 511</ows:UpperCorner>
</ows:BoundingBox>
```

NOTE In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.6 Use case 5: No resampling of unrectified but georeferenced image, defining extent in ground CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the base CRS of the georeferencing Transformations). Any ground CRS can be used that the server can transform into the baseCRS of both the stored coverage and the desired re-rectified output coverage. Notice that this extent can be skewed with curved edges in the stored coverage.

NOTE 1 The two specified corners of this BoundingBox are not required to fall exactly on stored pixels in order to avoid resampling, since the Output part of the GetCoverage request implies the ImageCRS of the stored image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
    <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
    <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405&
```

For this use case, the Output part of the GetCoverage request does not specify any wcs:GridCRS. The CRS used by the actual output coverage will be the ImageCRS of the stored image, usually with different origin point or pixel.

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox (XML encoded), referencing the ImageCRS of output image (required unless all this information is included in the coverage format used)
- c) Mandatory georeferencing coordinate Transformation(s), referencing the ImageCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="imageCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>7 7</ows:UpperCorner>
</ows:BoundingBox>
```

NOTE 2 In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123. The wcs:GridCRS origin is at (589, 172).

H.3.7 Use case 6: Resampling of georectified coverage, defining extent in ground CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the base of desired re-rectified output coverage) Any ground CRS can be used that the server can transform into the baseCRS of both the stored coverage and the desired re-rectified output coverage. Notice that this extent can be skewed with curved edges in the stored georectified coverage and in the desired output image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
    <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
    <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405&
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS that does not specify the same grid points as the wcs:GridCRS of the stored georectified coverage. This wcs:GridCRS is the desired CRS of the output coverage. The baseCRS of this wcs:GridCRS (roughly) corresponds to the old ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The baseCRS of this wcs:GridCRS can be any ground CRS, either the same as or different from the stored georectified coverage.

NOTE 1 If only the origin in this output wcs:GridCRS is different from the GridCRS of the stored georectified coverage, but that origin is at a different grid point in the GridCRS of the stored georectified coverage, then this output GridCRS specifies the same grid positions and no resampling is needed, see use case 1

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
</GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrS&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)
- c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>987 987</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS gml:id="TBD"> See Addition B below
```

NOTE 2 In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.8 Use case 7: Resampling of georectified image, defining extent in image CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS of the stored georectified coverage. Notice that this extent cannot be skewed or with curved edges in the stored unrectified coverage. However, this extent can be skewed with curved edges in the desired output image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID"
  dimension="2">
    <ows:LowerCorner>1000 2000</ows:LowerCorner>
    <ows:UpperCorner>1511 2511</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSpixelCenter:ImageCrsID&
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS that does not specify the same grid points as wcs:GridCRS of the stored georectified coverage (not the same CRS as referenced in the DomainSubset BoundingBox). This wcs:GridCRS is the desired CRS of the output coverage. The baseCRS of this wcs:GridCRS (roughly) corresponds to the old ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The baseCRS of this wcs:GridCRS can be any ground CRS, either the same as or different from the stored georectified coverage.

Note: If only the origin in this output wcs:GridCRS is different from the GridCRS of the stored georectified coverage, but that origin is at a different grid point in the GridCRS of the stored georectified coverage, then this output GridCRS specifies the same grid positions and no resampling is needed, see use case 2

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
</GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

- Array of output coverage grid/pixel values
- ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS
- Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>765 765</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.9 Use case 8: Resampling of unrectified and un-georeferenced image

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS of stored unrectified coverage. Notice that this extent cannot be skewed or with curved edges in the stored unrectified coverage. However, this extent can be skewed with straight edges in the desired output image.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID"
  dimension="2">
    <ows:LowerCorner>1000 2000</ows:LowerCorner>
    <ows:UpperCorner>1511 2511</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS based on the ImageCRS of the stored unrectified coverage (is the same CRS as referenced in the DomainSubset BoundingBox). This wcs:GridCRS is the desired CRS of the output coverage. The baseCRS of this wcs:GridCRS (roughly) corresponds to the ResponseCRS from WCS 1.0.0. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0.0 RESX, RESY, and RESZ.

NOTE 1 No resampling is needed when this wcs:GridCRS (that is based on the ImageCRS of the stored georectified coverage) has parameters that define grid positions which exactly match the grid positions in that ImageCRS, but this should be rare.

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
</GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage based on the ImageCRS of stored unrectified coverage and defines same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)

- c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>765 765</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 2 In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.10 Use case 9: Resampling of unrectified but georeferenced image, defining extent in image CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing the ImageCRS of stored unrectified coverage. Notice that this extent cannot be skewed or with curved edges in the stored unrectified coverage. However, this extent can be skewed with curved edges in the desired output coverage.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID"
  dimension="2">
    <ows:LowerCorner>1000 2000</ows:LowerCorner>
    <ows:UpperCorner>1511 2511</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=1000,2000,1511,2511,urn:ogc:def:crs:OGC:0.0:ImageCRSPixelCenter:ImageCrsID&
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS (not the same CRS as referenced in the DomainSubset BoundingBox). This wcs:GridCRS is the desired CRS of the output coverage. The baseCRS of this wcs:GridCRS (roughly) corresponds to the old ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The baseCRS of this wcs:GridCRS can be any ground CRS, either the same as or different from the stored georectified coverage.

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
  </GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCrs&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,-0.0707,0.1414,0.1414&
```


For this use case, the GetCoverage response contains:

- a) Array of output coverage grid/pixel values
- b) ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)
- c) Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)
- d) Mandatory georeferencing coordinate Transformation(s), referencing the ImageCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>765 765</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.11 Use case 10: Resampling of unrectified but georeferenced image, defining extent in ground CRS

For this use case, the GetCoverage request can specify a DomainSubset that includes an OWS BoundingBox (XML or KVP encoded) referencing a ground CRS (not necessarily the base CRS of desired rectified output coverage). Any ground CRS can be used that the server can transform into the baseCRS of both the stored coverage and the desired re-rectified output coverage. Notice that this extent can be skewed with curved edges in the stored unrectified coverage. However, this extent cannot be skewed or with curved edges in the desired output coverage.

EXAMPLE 1 DomainSubset XML fragment (2D):

```
<DomainSubset>
  <ows:BoundingBox crs="urn:ogc:def:crs:EPSG:6.6:63266405" dimension="2">
    <ows:LowerCorner>30.25 110.15</ows:LowerCorner>
    <ows:UpperCorner>30.75 110.65</ows:UpperCorner>
  </ows:BoundingBox>
</DomainSubset>
```

EXAMPLE 2 BoundingBox KVP fragment (2D):

```
BoundingBox=30.25,110.15,30.75,110.65,urn:ogc:def:crs:EPSG:6.6:63266405
```

For this use case, the Output part of the GetCoverage request can specify any wcs:GridCRS (not the same CRS as referenced in the DomainSubset BoundingBox). This wcs:GridCRS is the desired CRS of the output coverage. The baseCRS of this wcs:GridCRS (roughly) corresponds to the old ResponseCRS. This wcs:GridCRS contains additional information, including the information corresponding to the WCS 1.0 RESX, RESY, and RESZ. The baseCRS of this wcs:GridCRS can be any ground CRS.

EXAMPLE 3 XML fragment (2D):

```
<Output...>
  <GridCRS> <!-- See Addition B below for examples -->
</GridCRS>
</Output>
```

EXAMPLE 4 KVP fragment (2D):

```
GridBaseCRS=urn:ogc:def:crs:EPSG:6.6:32618&
GridType=urn:ogc:def:method:WCS:1.1:2dGridIn2dCr&
GridCS=urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS&
GridOrigin=1.0,2.0&
GridOffsets=0.0707,10.0707,0.1414,0.1414&
```

For this use case, the GetCoverage response contains:

- Array of output coverage grid/pixel values
- ows:BoundingBox (XML encoded), referencing the wcs:GridCRS for the output coverage that is based on the same ground CRS and defines same grid positions as requested output wcs:GridCRS (required unless all this information is included in the coverage format used)
- Definition of this output coverage wcs:GridCRS (required unless all this information is included in the coverage format used)
- Mandatory georeferencing coordinate Transformation(s), referencing the ImageCRS of the output image (this Transformation(s) might be included in the coverage format used)

EXAMPLE 5 XML fragment (2D): (when XML encoded)

```
<ows:BoundingBox crs="gridCRSofOutputCoverage" dimension="2">
  <ows:LowerCorner>0 0</ows:LowerCorner>
  <ows:UpperCorner>987 987</ows:UpperCorner>
</ows:BoundingBox>
<wcs:GridCRS> <!-- See Addition B below for examples -->
</wcs:GridCRS>
```

NOTE 2 In this example, the LowerCorner is the (0, 0) origin of this output grid, in compliance with ISO 19123.

H.3.12 Addition B

EXAMPLE 1 An XML document fragment for a 2D wcs:GridCRS without rotation or skewing, using the 2dGridIn2dCr method is:

```
<GridCRS gml:id="example2dGridCr1">
  <gml:srsName>2D Grid CRS Example 1</gml:srsName>
  <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:63266405</GridBaseCRS>
  <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCr</GridType>
  <GridOrigin>1.0 2.0</GridOrigin>
  <GridOffsets>0.1 0 0 0.2</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
</GridCRS>
```

EXAMPLE 2 An XML document fragment for a 2D wcs:GridCRS with rotation but no skewing using the 2dGridIn2dCr method is:

```
<GridCRS gml:id="example2dGridCr2">
  <GridBaseCRS>urn:ogc:def:crs:EPSG:6.6:32618</GridBaseCRS>
  <GridType>urn:ogc:def:method:WCS:1.1:2dGridIn2dCr</GridType>
  <GridOrigin>1.0 2.0</GridOrigin>
  <GridOffsets>0.0707 -0.0707 0.1414 0.1414</GridOffsets>
  <GridCS>urn:ogc:def:cs:OGC:0.0:Grid2dSquareCS</GridCS>
```

</GridCRS>

Annex I (normative)

Other data structures

I.1 Overview

This annex specifies the Coverages data structure that is used by this WCS, with the Manifest data structure that it uses. This annex also specifies the InterpolationMethods data structure.

NOTE The Coverages, Manifest, and InterpolationMethods are proposed additions to OWS Common [OGC 05-008], and are thus in the “ows” XML namespace. This Manifest is proposed in OWS Common change requests [OGC 06-015r3]. Although discussed here as additions to OWS Common, the Coverages and InterpolationMethods might be placed in a separate document specifying aspects specific to coverages that are used by multiple Implementation specifications.

I.2 Manifest

I.2.1 Manifest contents

A manifest is a document describing the contents of a package of documents. A manifest document shall be structured as shown in the UML class diagram in Figure I.1. The attributes and associations of the four new classes shall include the parameters and data structures listed and defined in Table I.1 through Table I.4.

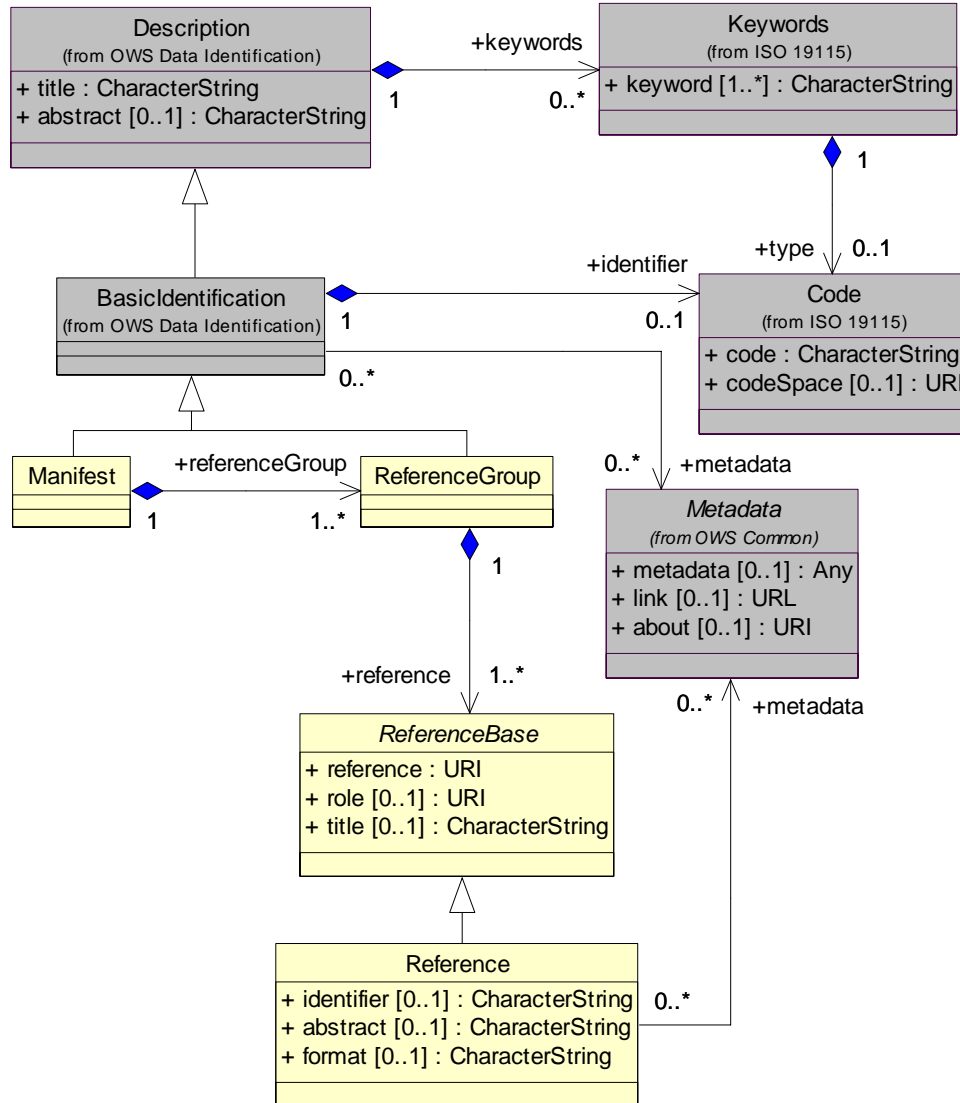


Figure I.1 — Manifest UML package

NOTE 1 To reduce the need for readers to refer to other documents, the first five parameters listed below are largely copied from Table 23 in Subclause 10.6.1 of [OGC 05-008].

Table I.1 — Manifest data structure

Names ^a	Definition	Data type	Multiplicity and use
identifier Identifier	An unambiguous identifier of this Manifest document, normally used by software	MD_Identifier class in ISO 19115 ^b	Zero or one (optional) Include when available and useful
title Title	Title of this Manifest document, normally used for display to a human	Character string type, not empty	Zero or one (optional) Include when available and useful ^c
abstract Abstract	Brief narrative description of this Manifest document, normally available for display to a human	Character string type, not empty	Zero or one (optional) Include when available and useful
keywords Keywords	Unordered list of one or more commonly used or formalised word(s) or phrase(s) used to describe this Manifest document	MD_Keywords class in ISO 19115	Zero or more (optional) One for each keyword authority used
metadata Metadata	Additional metadata about this Manifest document ^d	reference to metadata or metadata contents, see gml:metaDataProperty ^e	Zero or more (optional) One for each useful metadata object
reference Group Reference Group	References to a logical group of documents or resources within this manifest document	ReferenceGroup data structure, see Table I.2	Zero or more (optional) One for each group included
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b The optional codeSpace attribute in the ows:CodeType is expected to rarely be used in the short term. Wherever a specific OWS specification expects this codeSpace attribute to be used, that specification shall specify how it should be used there, including its values and meanings.</p> <p>c Software may display the “Identifier” value when the “Title” is absent.</p> <p>d This Metadata should be used primarily by specific servers, not by specific OWSs. The specification editors for each specific OWS should decide what additional metadata to require or encourage be included. That specific OWS should then specify additional parameters and groups as needed to contain this additional metadata, with specific names and meanings.</p> <p>e Use or adaptation of the gml:metaDataProperty data type is specified in order to include references to the type of metadata that is pointed to (or included), and to what aspect of the data(set) this metadata applies to.</p>			

NOTE 2 To reduce the need for readers to refer to other documents, the first five parameters listed below are largely copied from Table 23 in Subclause 10.6.1 of [OGC 05-008].

Table I.2 — ReferenceGroup data structure

Names ^a	Definition	Data type	Multiplicity and use
identifier Identifier	An unambiguous identifier of this Reference Group, normally used by software	MD_Identifier class in ISO 19115 ^b	Zero or one (optional) Include when available and useful
title Title	Title of this Reference Group, normally used for display to a human	Character string type, not empty	Zero or one (optional) Include when available and useful ^c
abstract Abstract	Brief narrative description of this Reference Group, normally available for display to a human	Character string type, not empty	Zero or one (optional) Include when available and useful
keywords Keywords	Unordered list of one or more commonly used or formalised word(s) or phrase(s) used to describe this Reference Group	MD_Keywords class in ISO 19115	Zero or more (optional) One for each keyword authority used
metadata Metadata	Additional metadata about this Manifest document ^d	reference to metadata or metadata contents, see gml:metaDataProperty ^e	Zero or more (optional) One for each useful metadata object
reference Reference	Reference to one document or resource	Reference data structure, see Table 4	One or more (mandatory) One for each reference in this group
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b The optional codeSpace attribute in the ows:CodeType is expected to rarely be used in the short term. Wherever a specific OWS specification expects this codeSpace attribute to be used, that specification shall specify how it should be used there, including its values and meanings.</p> <p>c Software may display the “Identifier” value when the “Title” is absent.</p> <p>d This Metadata should be used primarily by specific servers, not by specific OWSs. The specification editors for each specific OWS should decide what additional metadata to require or encourage be included. That specific OWS should then specify additional parameters and groups as needed to contain this additional metadata, with specific names and meanings.</p> <p>e Use or adaptation of the gml:metaDataProperty data type is specified in order to include references to the type of metadata that is pointed to (or included), and to what aspect of the data(set) this metadata applies to.</p>			

Table I.3 — ReferenceBase data structure

Names ^a	Definition	Data type	Multiplicity and use
reference xlink:href	Reference to a remote resource or local payload.	URI ^b	One (mandatory)
role xlink:role	Reference to a resource that describes the role of this reference.	URI	Zero or one (optional) Include when available and useful ^c
title xlink:title	Describes the meaning of the referenced resource in a human-readable fashion	Character String, not empty	Zero or one (optional) Include when available and useful
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b A remote resource is typically addressed by a URL.</p> <p>c When no value is supplied, no particular role value is to be inferred.</p>			

This ReferenceBase class (or data structure) is abstract, and thus is not intended to be used without some modification. This class is defined to permit future adaptation as needed. The Reference class (or data structure) extension of this ReferenceBase class is provided for typical uses.

NOTE 3 To reduce the need for readers to refer to other documents, the first four parameters listed below are largely copied from Table 23 in Subclause 10.6.1 of [OGC 05-008]. The last three parameters are copied from Table I.3 above.

Table I.4 — Reference data structure

Names ^a	Definition	Data type	Multiplicity and use
identifier Identifier	An unambiguous identifier to this document or resource, normally used by software	MD_Identifier class in ISO 19115 ^b	Zero or one (optional) Include when available and useful
abstract Abstract	Brief narrative description of this document or resource, normally available for display to a human	Character string type, not empty	Zero or one (optional) Include when available and useful
format format	Reference to a format in which document or resource is encoded	Character string type, not empty	Zero or more (optional) ^f Include when relevant and available
metadata Metadata	Additional metadata about this document or resource ^c	reference to metadata or metadata contents, see gml:metaDataProperty ^d	Zero or more (optional) One for each useful metadata object
reference xlink:href	Reference to a document or resource in this package or outside it	URI	One (mandatory)
role xlink:role	Reference to a resource that describes the role of this reference	URI	Zero or one (optional) Include when available and useful ^b
title xlink:title	Describes the meaning of the referenced resource in a human-readable fashion	Character String, not empty	Zero or one (optional) Include when available and useful
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b The optional codeSpace attribute in the ows:CodeType is expected to rarely be used in the short term. Wherever a specific OWS specification expects this codeSpace attribute to be used, that specification shall specify how it should be used there, including its values and meanings.</p> <p>c This Metadata should be used primarily by specific servers, not by specific OWSs. The specification editors for each specific OWS should decide what additional metadata to require or encourage be included. That specific OWS should then specify additional parameters and groups as needed to contain this additional metadata, with specific names and meanings.</p> <p>d Use or adaptation of the gml:metaDataProperty data type is specified in order to include references to the type of metadata that is pointed to (or included), and to what aspect of the data(set) this metadata applies to.</p>			

I.2.2 XML encoding

A manifest document or data structure shall be encoded as specified in the attached XML Schema Document named owsManifest.xsd. The ReferenceBase class (or data structure) is XML encoded by adapting the xlink:simpleLink attribute group (which is used by GML).

EXAMPLE A simple manifest for a zip file is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Manifest xmlns="http://www.opengis.net/ows/1.1"
```



```

xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/ows/1.1
http://schemas.opengis.net/ows/1.1.0/owsManifest.xsd">
  <ReferenceGroup>
    <Abstract>Coverage created from a GetCoverage operation request to a
WCS</Abstract>
    <Reference xlink:href="coverage/image.tiff"
xlink:role="urn:ogc:role:coverage" />
    <Reference xlink:href="coverage/metadata.xml"
xlink:role="urn:ogc:role:metadata" />
  </ReferenceGroup>
</Manifest>

```

Assuming this manifest is contained in a zip file containing two files plus the manifest.xml document, the zip file would contain:

```

ogc-coverage.zip
  ows-manifest.xml
  coverage/image.tiff
  coverage/metadata.xml

```

I.3 Coverages data structure

I.3.1 Introduction

In some cases, an operation response can consist of a group of coverages using essentially the same contents as the Manifest specified above. Similarly, part of an operation request can contain a group of coverages using essentially the same contents. That is, this “Coverages” data structure can contain multiple coverages, where each coverage can be a group of related data items. In this use, all of these data items might be contained in one package that is transferred as the operation response or in the operation request message, or some of these data items might be stored elsewhere for subsequent retrieval.

EXAMPLE 1 Some responses to a WCS GetCoverage operation can contain multiple groups of related data items, where each group contains an output coverage with its metadata.

EXAMPLE 2 The request for a WCS Transaction operation will contain multiple groups of related data items, where each group is an input coverage with its metadata.

When used for operation inputs, all or some of these data items might be accessible from another OGC or other web service using an operation. If this operation request is KVP encoded, that complete operation request shall be encoded as a URL value, in the xlink:href attribute in the AbstractReferenceBaseType. However, if this operation request is XML encoded, a XML encoded operation request must be sent to the other web server whose operation URL is referenced by the xlink:href attribute.

In these cases, the Manifest (UML class and XML element) specified above could be used in an operation request to a specific OWS. However, the needed operation request(s) to the other server would not be directly associated with the corresponding instances providing xlink:href attribute values (in the AbstractReferenceBaseType). Furthermore, the Manifest name could be misleading when used in the operation request, where a more-specific name may be preferred for a specific OWS.

This document thus defines a ServiceReference (class and element) that shall be used whenever applicable in operation requests that are defined by specific OWSs. This class is a

subclass of the Reference class, which adds a UML attribute that either contains or references the XML encoded operation request needed to retrieve this input data from the other web server.

This subclass also defines a Coverages class, and corresponding element using the ManifestType, which may be used by specific OWSs. This Coverages class is a subclass of the Manifest class, with no additions or changes in the class attributes except that the ReferenceGroup is renamed Coverage. These classes are included in the Coverage Group package UML class diagram in Figure I.2.

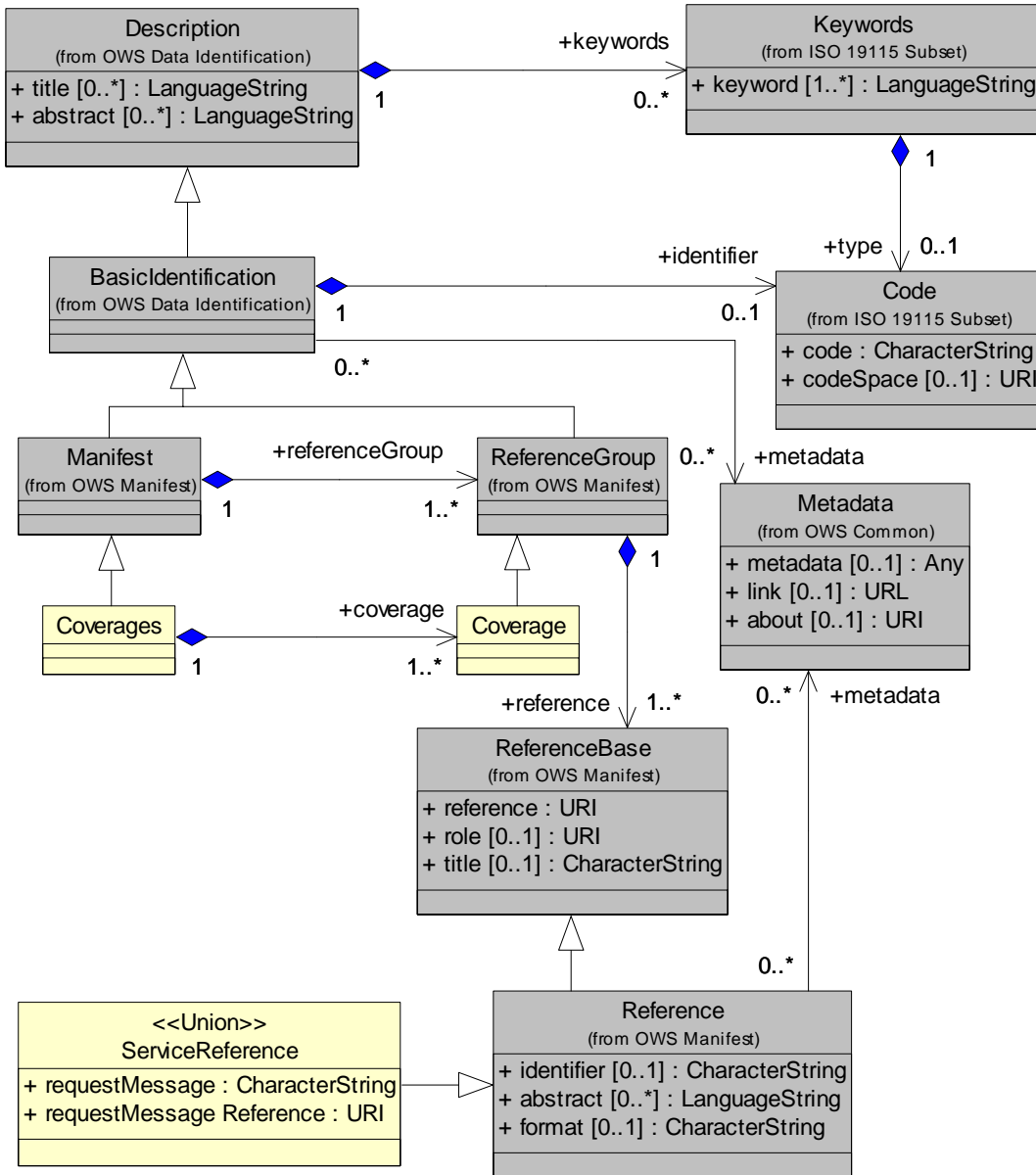


Figure I.2 — Coverage Group package partial UML diagram

I.3.2 XML encoding

A Coverages data structure shall be encoded as specified in the attached owsCoverages.xsd file.

EXAMPLE Here is an example XML document using the Coverages element:

```
<?xml version="1.0" encoding="UTF-8"?>
<Coverages xmlns="http://www.opengis.net/wcs/1.1/ows"
  xmlns:ows="http://www.opengis.net/ows"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:schemaLocation="http://www.opengis.net/wcs/1.1/ows ../owsCoverages.xsd
http://www.opengis.net/ows ../../../../ows/1.0.0/ows19115subset.xsd">
  <Coverage>
    <ows:Title>TBD</ows:Title>
    <ows:Abstract>Coverage created from GetCoverage operation request to a
WCS</ows:Abstract>
    <Identifier>TBD</Identifier>
    <Reference xlink:href="coverage/image.tiff"
xlink:role="urn:ogc:def:role:WCS:1.1:coverage"/>
    <Reference xlink:href="coverage/metadata.xml"
xlink:role="urn:ogc:def:role:WCS:1.1:metadata"/>
  </Coverage>
</Coverages>
```

I.4 Interpolation methods

I.4.1 Interpolation methods contents

The InterpolationMethods data structure lists one or more spatial interpolation methods, which can each be used for interpolation of a range record field in a continuous grid coverage. The InterpolationMethods shall use the data structure described in Figure I.3 and specified in Tables I.5 through Table I.8.

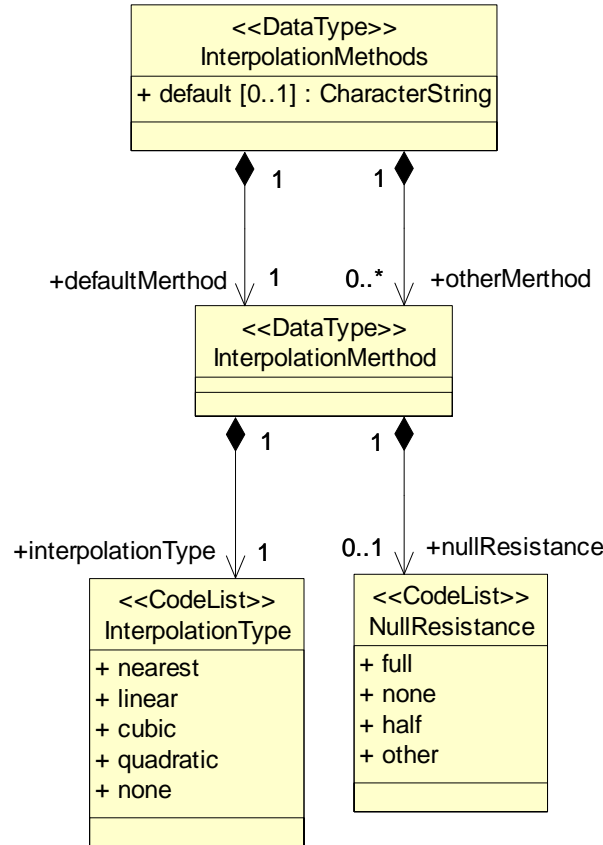


Figure I.3 — Interpolation Methods class diagram

Table I.5 — Parts of the InterpolationMethods data structure

Names ^a	Definition	Data type	Multiplicity and use
defaultMethod DefaultMethod	Spatial interpolation method used for this range field when client doesn't specify one ^b	InterpolationMethod data structure, see Table I.6	One (mandatory)
otherMethod OtherMethod	Another spatial Interpolation method that server can apply to this range field	InterpolationMethod data structure, see Table I.6	Zero or more (optional) Include for each other method supported ^c
<p>a See Table 1 for UML and XML naming conventions.</p> <p>b This default interpolation method should be the recommended or normal method for this coverage range field.</p> <p>c When the default interpolation method is “none”, no other methods should be listed.</p>			

InterpolationMethods has a mandatory **DefaultMethod** parameter that identifies the spatial interpolation method used for GetCoverage requests that don't specify one.

InterpolationMethods may also have zero or more **OtherMethod** parameters whose values identify other spatial interpolation methods that the server can use for this coverage range field. When the only listed (default) interpolation method is “none”, clients may only retrieve subsets of this coverage in its native CRS and at its native resolution.

Table I.6 — Parts of the InterpolationMethod data structure

Names ^a	Definition	Data type and value	Multiplicity and use
Interpolation MethodType (<i>anonymous</i>)	Identifier of spatial interpolation method.	CodeList type One of the strings defined in Table I.7	One (mandatory)
nullResistance nullResistance	How server handles null values when spatially interpolating values in this field	CodeList type One of the strings listed in Table I.8.	Zero or one (optional) Include when known
a See Table 1 for UML and XML naming conventions.			

For a continuous grid coverage, the normal spatial interpolation types are listed in Table I.7 and in the attached default InterpolationMethods.xml dictionary. Servers may implement and list other interpolation methods, defined in dictionaries other than the default.

Table I.7 — Normal spatial interpolation types

Interpolation type identifier	Interpolation method	Description
nearest	nearest neighbour	These are defined in ISO 19123 (Schema for Coverage Geometry and Functions), Annex B.
linear	bilinear ^a	
quadratic	biquadratic ^a	
cubic	bicubic ^a	
none	None	No interpolation is available; requests must be for locations that are among the original domain locations.
a ISO 19123 specifies two-dimensional bilinear, biquadratic, and bicubic interpolation types. When also interpolating in the elevation dimension of a 3D CRS, the 3D extensions of these interpolation types shall be used.		

Each **InterpolationMethod** element has an optional **nullResistance** summarizing how the server treats null-valued points in interpolation. The normal values for this parameter are listed in Table I.8 and in the default InterpolationMethods.xml dictionary.

Table I.8 — Effect of null values in spatial interpolation

nullResistance	Description
full	Nulls in the input grid(s) have no effect on interpolated value; resampling uses only non-null valued cells.
none	Interpolated value is Null when any one input value is Null.
half	Interpolated value is Null when half or more input values are Null.
other	Resampling treats nulls differently than any method described above

I.4.2 Interpolation methods XML encoding

A set of interpolation methods can be XML encoded, in each Field in the Range part of each CoverageDescription. The XML encoding of InterpolationMethods shall use the XML Schema fragment in the attached owsInterpolationMethod.xsd file. The InterpolationType and nullResistance are UML <CodeList> stereotyped classes, which are thus converted into XML Schema following the GML pattern for XML encoding of <CodeList> classes.

That is, in `owsInterpolationMethod.xsd`, the normal values of `InterpolationType` and `nullResistance` are NOT encoded in the XML Schema, but in the default-referenced GML Dictionary. That Dictionary document is named `interpolationMethods.xml` and also attached to this document. This approach is used since not all useful values of these quantities are currently known, and this approach allows additional values to be more easily specified and used.

EXAMPLE Here is an example `InterpolationMethods` data structure encoded in XML. It lists four interpolation methods: the first three (“cubic,” “linear,” “quadratic”) are defined in the default codespace dictionary; the fourth (“sinusoidal”) is defined at the URL given in the `codeSpace` attribute.

```
<?xml version="1.0" encoding="UTF-8"?>
<InterpolationMethods xmlns="http://www.opengis.net/wcs/1.1/ows"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="..\..\wcs\1.1.0\owsInterpolationMethod.xsd">
  <DefaultMethod>cubic</DefaultMethod>
  <OtherMethod>linear</OtherMethod>
  <OtherMethod>quadratic</OtherMethod>
  <OtherMethod codeSpace="http://my.server.org/OtherInterpolationMethods.xml">
    sinusoidal</OtherMethod>
</InterpolationMethods>
```

I.4.3 Interpolation method KVP encoding

One selected interpolation method can be KVP encoded for each range field in a `GetCoverage` operation request as specified in Subclause 10.2.2.

Bibliography

- [1] OGC 00-014r1, Guidelines for Successful OGC Interface Specifications
- [2] ISO 19103, Geographic Information – Conceptual schema language
- [3] OGC 04-024, Web Map Service (WMS), version 1.3
- [4] OGC 05-007r4, Web Processing Service (WPS) Implementation Specification
- [5] OGC 06-015r2, OWS Common change request: Add Manifest specification
- [6] OGC 06-023r1, Definition identifier URNs in OGC namespace
- [7] OGC 06-056, OWS Common change request: Add input and output specifications
- [8] OGC 06-094, Web Coverage Service (WCS) Implementation Specification
- [9] OMG Unified Modeling Language Specification (UML), Version 1.5, March 2003, <http://www.omg.org/docs/formal/03-03-01.pdf>
- [10] ISO 8601:2000, Data elements and interchange formats — Information interchange — Representation of dates and times