

# Open Geospatial Consortium

Approval Date: 2013-09-06

Publication Date: 2014-03-11

Reference number of this Document: OGC 11-053r1

External reference URI for this standard: <http://www.opengis.net/doc/IS/WCS-CRS-extension/1.0>

Version: 1.0

Category: Interface Standard

Editor: Peter Baumann, Jinsongdi Yu

## OGC® Web Coverage Service Interface Standard - CRS Extension

Copyright © 2014 Open Geospatial Consortium.

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

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

Document type:	OGC Standard
Document subtype:	Interface extension
Document stage:	Adopted
Document language:	English

## License Agreement

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

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

THIS LICENSE IS A COPYRIGHT LICENSE ONLY, AND DOES NOT CONVEY ANY RIGHTS UNDER ANY PATENTS THAT MAY BE IN FORCE ANYWHERE IN THE WORLD.

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

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

Except as contained in this notice, the name of LICENSOR or of any other holder of a copyright in all or part of the Intellectual Property shall not be used in advertising or otherwise to promote the sale, use or other dealings in this Intellectual Property without prior written authorization of LICENSOR or such copyright holder. LICENSOR is and shall at all times be the sole entity that may authorize you or any third party to use certification marks, trademarks or other special designations to indicate compliance with any LICENSOR standards or specifications.

This Agreement is governed by the laws of the Commonwealth of Massachusetts. The application to this Agreement of the United Nations Convention on Contracts for the International Sale of Goods is hereby expressly excluded. In the event any provision of this Agreement shall be deemed unenforceable, void or invalid, such provision shall be modified so as to make it valid and enforceable, and as so modified the entire Agreement shall remain in full force and effect. No decision, action or inaction by LICENSOR shall be construed to be a waiver of any rights or remedies available to it.

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

<b>Contents</b>		<b>Page</b>
1	Scope.....	1
2	Conformance.....	1
3	Normative references.....	2
4	Terms and definitions.....	2
4.1	CRS 2.....	
4.2	Native CRS [of a coverage].....	2
4.3	CRS Identifier.....	3
5	Conventions.....	3
5.1	UML notation.....	3
5.2	Data dictionary tables.....	3
5.3	Namespace prefix conventions.....	3
5.4	Multiple representations.....	4
6	<i>Crs</i> requirements class.....	4
6.1	Overview.....	4
6.2	Modifications to <i>GetCapabilities</i> .....	4
6.3	Modifications to <i>DescribeCoverage</i> .....	5
6.4	Modifications to <i>GetCoverage</i> .....	6
6.4.1	Modifications to the <i>GetCoverage</i> request.....	6
6.4.2	Modifications to the <i>GetCoverage</i> response.....	7
6.5	Exceptions.....	8
6.6	Encodings.....	8
6.5.1	GET/KVP Encoding.....	8
6.5.2	XML/POST Encoding.....	9
6.5.3	SOAP Encoding.....	9
7	<i>Crs-gridded-coverage</i> requirements class.....	10
7.1	Overview.....	10
7.2	Modifications to <i>GetCapabilities</i> .....	10
7.3	Modifications to <i>GetCoverage</i> .....	10
8	<i>Crs-discrete-coverage</i> requirements class.....	11
8.1	Overview.....	11
8.2	Modifications to <i>GetCapabilities</i> .....	11
8.3	Modifications to <i>GetCoverage</i> .....	11
	Bibliography.....	12
	Annex A (normative) Abstract test suite.....	13
A.1	Conformance Test Class: <i>crs</i> .....	13
A.1.1	CRS extension identifier.....	13
A.1.2	Capabilities.....	13
A.1.3	CrsMetadataelement.....	13
A.1.4	Output CRS.....	14
A.1.5	Unique output CRS.....	14

A.1.6	GetCoverage CRS.....	14
A.1.7	Service Metadata Output CRS.....	15
A.1.8	Unique GetCoverageSubsettingCrs CRS.....	15
A.1.9	Available GetCoveragesubsettingCRS.....	15
A.1.10	Default GetCoverageOutputCRS.....	16
A.1.11	Unique GetCoverageOutputCRS.....	16
A.1.12	Available GetCoverageoutputCRS.....	16
A.1.13	GetCoverage Response CRS.....	17
A.1.14	GetCoverage Exception.....	17
A.1.15	GetCoverageGetKVPSubsettingcrs.....	17
A.1.16	GetCoverageOutputCRS.....	18
A.1.17	GetCoverage XML/POST.....	18
A.1.18	GetCoverage SOAP.....	18
A.2	Conformance Test Class: <i>crs-gridded-coverage</i> .....	18
A.2.1	CRS Gridded Coverage Dependency.....	19
A.2.2	Gridded Coverage CRS.....	19
A.2.3	GetCoverage Range Set.....	19
A.3	Conformance Test Class: <i>crs-discrete-coverage</i> .....	20
A.3.1	CRS Discrete Coverage Dependency.....	20
A.3.2	Crs Discrete Coverage Identifier.....	20
A.3.3	Crs Discrete Coverage Range Set.....	20

<b>Tables</b>	<b>Page</b>
Table 1 — Conformance class dependencies .....	2
Table 2— Namespace mappings .....	3
Table 3 — Components of CRS : : CRS structure.....	5
Table 4 — Components of CRS : : GetCoverageCrs structure .....	6
Table 5 — Exception codes for use of outputCrs .....	8

## i. Abstract

This document specifies parameters to the OGC Web Coverage Service (WCS) *GetCoverage* request that allows a client, a service, or other application to specify the Coordinate Reference System (CRS) in which coverages are delivered. Note that the CRS of the input bounding box is already defined in the OGC WCS Core Implementation Standard [OGC 09-110r3].

## ii. Preface

Suggested additions, changes, and comments on this draft document are welcome and encouraged. Such suggestions may be submitted by email message or by making suggested changes in an edited copy of this document.

## iii. Terms

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

## iv. Submitting organizations

The following organizations have submitted this Interface Specification to the Open Geospatial Consortium:

- Jacobs University Bremen

## v. Document Contributor Contact Points

Name	Organization
Peter Baumann	Jacobs University Bremen, rasdaman GmbH
Jinsongdi Yu	Jacobs University Bremen

## vi. Revision history

Date	Release	Author	Paragraph modified	Description
2011-07-10	0.0.2	Peter Baumann	All	Created, from earlier CR draft
2011-12-09	0.1.0	Peter Baumann, Jinsongdi Yu	Several, in particular: Annex A	Finalized and added ATS

2012-08-30	1.0.0	Peter Baumann, Jinsongdi Yu	First pages Tables Subclause 6.2 Subclause 6.5 Clause 7, 8  Annex A	Final editing for release of adopted version Corrected table numbering Disentangled Req2 and Req 3 Added CrsMismatch Switched Grid and Discrete Coverage Performed adjustments required Add A.3.3
------------	-------	--------------------------------	---	---

## vii. Changes to the OGC<sup>®</sup> Abstract Specification

The OGC<sup>®</sup> Abstract Specification does not require any changes to accommodate the technical contents of this (part of this) document.

## viii. Future Work

Among the topics for future development are the following items:

- Replacement of 07-092r3 through creation of an OGC-NA Name Type Specification (NTS) for predefined, combined, and parameterized CRSs to augment the /def/ NTS (which defines http URIs for standard CRS).
- Integrate AUTO CRS definitions in the OGC-NA NTS for CRSs.
- CR to GML: EnvelopeWithCRSType (similar to EnvelopeWithTimePeriodType).
- CR to GML to allow string coordinates (needed for representation of temporal coordinates).

## Foreword

This WCS CRS extension is an OGC Interface Standard which relies on WCS Core [OGC 09-110r4] and the GML Application Schema for Coverages [OGC 09-146r2].

This document includes one normative Annex.

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

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



## Introduction

The OGC *Web Coverage Service (WCS)–CRS Extension* defines an extension to the WCS Core [OGC 09-146r2] allowing to obtain coverages in various Coordinate Reference Systems (CRSs).

While the WCS Core [OGC 09-110r4] defines access only in the coverage's *Native CRS* (i.e., the one in which the coverage is stored on the server), this WCS CRS Extension defines how to request and obtain a coverage in CRSs different from the Native CRS, and also how to provide a subsetting bounding box with coordinates in a CRS different from the Native CRS.

CRSs in this WCS Extension are identified by *CRS Identifiers* such as defined in the OGC CRS Name Type Definition [OGC 11-135]. A WCS server supporting this WCS CRS Extension announces the CRSs supported by listing their CRS Identifiers in its Capabilities document.



# OGC® Web Coverage Service Interface Standard - CRS Extension

## 1 Scope

This OGC WCS CRS Extension – in short: *CRS Extension* – defines an extension to the WCS Core [OGC 09-110r4] allowing to obtain coverages and to specify subsets in various Coordinate Reference Systems (CRSs).

While the WCS Core defines access only in the coverage's Native CRS (i.e., the one in which the coverage is stored on the server), this WCS CRS Extension defines how to request and obtain a coverage in CRSs different from the Native CRS, and also how to provide a subsetting bounding box with coordinates in a CRS different from the Native CRS.

CRSs in this WCS Extension are identified by *CRS Identifiers* such as defined in the OGC CRS Name Type Definition [OGC 11-135]. A WCS server supporting this WCS CRS Extension announces the CRSs supported by listing them in its Capabilities document.

## 2 Conformance

This document establishes the following requirements and conformance classes:

- *crs*, of URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/req/crs](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/req/crs); the corresponding conformance class is *crs*, with URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs).
- *crs-gridded-coverage*, of URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/req/crs-gridded-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/req/crs-gridded-coverage); the corresponding conformance class is *crs-gridded-coverage*, with URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs-gridded-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-gridded-coverage).
- *crs-discrete-coverage*, of URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/req/crs-discrete-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/req/crs-discrete-coverage); the corresponding conformance class is *crs-discrete-coverage*, with URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs-discrete-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-discrete-coverage).

Standardisation target of all requirements and conformance classes are WCS implementations (currently: servers).

Requirements URIs defined in this document are relative to [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/req](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/req), conformance test URIs are relative to [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf).

Annex A lists the conformance tests which shall be exercised on any software artefact claiming to implement WCS.

### 3 Normative references

This *OGC WCS CRS Extension* specification consists of the present document and an XML Schema. The complete specification is identified by OGC URI [http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0](http://www.opengis.net/spec/WCS_service-extension_crs/1.0), the document has OGC URI [http://www.opengis.net/doc/IS/WCS\\_service-extension\\_crs/1.0](http://www.opengis.net/doc/IS/WCS_service-extension_crs/1.0).

The complete specification is available for download from <http://www.opengeospatial.org/standards/wcs>; additionally, the XML Schema is posted online at <http://schemas.opengis.net/wcs/crs/1.0> as part of the OGC schema repository. In the event of a discrepancy between bundled and schema repository versions of the XML Schema files, the schema repository shall be considered authoritative.

The normative documents listed in Table 1 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.

**Table 1 — Conformance class dependencies**

CRS conformance class	Dependency document	Dependency conformance class
<i>crs</i>	OGC 09-146r2, <i>GML 3.2.1 Application Schema for Coverages</i> , version 1.0  OGC 09-110r4, <i>OGC® Web Coverage Service 2.0 Interface Standard - Core</i> , version 2.0	<i>gml-coverage</i>  <i>core</i>
<i>crs-discrete-coverage</i>	This document	<i>crs</i>
<i>crs-gridded-coverage</i>	This document	<i>crs</i>

### 4 Terms and definitions

For the purposes of this document, the terms and definitions given in the above references apply. In addition, the following terms and definitions apply. An arrow “ $\square$ ” indicates that the following term is defined in this Clause.

#### 4.1 CRS

Coordinate Reference System

#### 4.2 Native CRS [of a coverage]

$\square$  CRS indicated in the `srsName` attribute of the `GML::DomainSet` [of a coverage]

### 4.3 CRS Identifier

URL, conforming to the OGC CRS Name Type Definition [OGC 11-135], which resolves to a well-known  $\square$ CRS

Example 1 WGS84 has this CRS Name registered with OGC-NA [2]:  
<http://www.opengis.net/def/crs/EPSG/0/4326>

Example 2 A hypothetical temporal CRS<sup>1</sup> might have this CRS Name registered with OGC-NA:  
<http://www.opengis.net/def/crs/OGC/1.0/Gregorian-Date>

Example 3 A 3-dimensional CRS with dimensions latitude, longitude, and time can be represented as follows, according to the OGC CRS Name Type Definition:  
[http://www.opengis.net/def/crs-compound?  
 1=http://www.opengis.net/def/crs/EPSG/0/4326  
 &2=http://www.opengis.net/def/crs/OGC/1.0/Gregorian-Date](http://www.opengis.net/def/crs-compound?1=http://www.opengis.net/def/crs/EPSG/0/4326&2=http://www.opengis.net/def/crs/OGC/1.0/Gregorian-Date)

## 5 Conventions

### 5.1 UML notation

Unified Modeling Language (UML) static structure diagrams appearing in this specification are used as described in Subclause 5.2 of OGC Web Services Common [OGC 06-121r9].

### 5.2 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 Subclause 5.5 of [OGC 06-121r9]. The contents of these data dictionary tables are normative, including any table footnotes.

### 5.3 Namespace prefix conventions

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

**Table 2— Namespace mappings**

Prefix	Namespace URI	Description
xsd	<a href="http://www.w3.org/2001/XMLSchema">http://www.w3.org/2001/XMLSchema</a>	XML Schema namespace
gml	<a href="http://www.opengis.net/gml/3.2">http://www.opengis.net/gml/3.2</a>	GML 3.2.1
gmlcov	<a href="http://www.opengis.net/gmlcov/1.0">http://www.opengis.net/gmlcov/1.0</a>	GML Application Schema for Coverages 1.0

<sup>1</sup> Temporal CRS definitions are in preparation by OGC at the time of this writing, based on the syntax of ISO 8601 [1].

wcs	<a href="http://www.opengis.net/wcs/2.0">http://www.opengis.net/wcs/2.0</a>	WCS 2.0 Core
crs	<a href="http://www.opengis.net/wcs/crs/1.0">http://www.opengis.net/wcs/crs/1.0</a>	this WCS 2.0 CRS Extension

## 5.4 Multiple representations

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

## 6 Crs requirements class

### 6.1 Overview

Clients and servers supporting this CRS extension are CRS aware, that is: they can request and deliver coverages in CRSs other than only the one in which they are originally stored. A server announces both CRS support and the concrete CRSs known in the Capabilities document; a client can request delivery of a coverage obtained through a *GetCoverage* request in one of these CRSs announced by the server.

This Clause 6 establishes the CRS Extension core conformance class, *crs*, which defines how Web Coverage Services shall deal with CRSs.

### 6.2 Modifications to *GetCapabilities*

A server announces support of the CRS extension to a client by adding the URL identifying this extension to the list of supported extensions delivered in the Capabilities document.

#### Requirement 1 *crs/extension-identifier*:

A WCS service implementing conformance class *crs* of this CRS Extension **shall** include the following URI in the *Profile* element of the *ServiceIdentification* in a *GetCapabilities* response:

[http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs)

**Dependency:** WCS Core [OGC 09-110r4],

<http://www.opengis.net/spec/WCS/2.0/req/core>

A *Crs* item is added to the WCS service metadata using the extension mechanism provided by the WCS Core [OGC 09-110r3]. In its *crsSupported* component it contains the list of CRSs in which the server can return coverages.

#### Requirement 2 *crs/capabilities-crs*:

The response to a successful *GetCapabilities* request **shall** adhere to Figure 1, Table 3, and the XML schema defined for this CRS Extension.

Figure 1 presents the UML diagram of the extended Capabilities document. Table 3 details the components added.

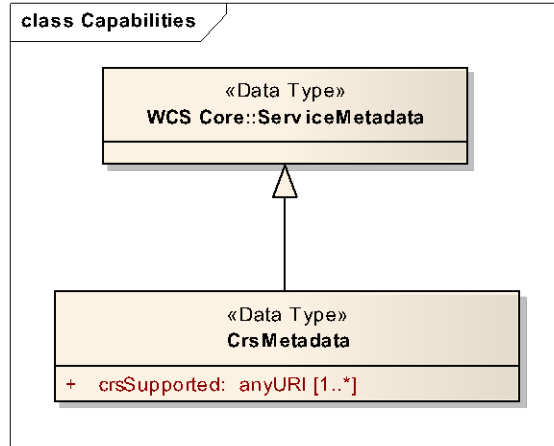


Figure 1 —Capabilities UML diagram with CrsMetadata component

Table 3 — Components of CRS : :CRS structure

Name	Definition	Data type	Multiplicity
crsSupported	Identifiers of CRSs a result coverage can bear	List of anyURI	one or more (optional)

**Requirement 3 crs/wcsServiceMetadata-crs:**

The response to a successful *GetCapabilities* request **shall** contain a `crs:CrsMetadata` element in the `wcs:Extension` element of `wcs:ServiceMetadata`.

**Requirement 4 crs/wcsServiceMetadata-outputCrs:**

The `crsSupported` item(s) delivered in the `CrsMetadata` element of the response to a successful *GetCapabilities* request **shall** consist of URLs, each one resolving to an instantiatable subtype of `GML::AbstractCRS`.

Note This does not per se mean that every CRS reported in the Capabilities document is applicable to every coverage available from the server or to every possible trim or slice subset of such a coverage. For example, a server may offer 2D and 3D coverages, in which case 2D coverages cannot be retrieved in a 3D CRS, and vice versa. Further, 2D slices from a 3D coverage cannot be returned in a 3D CRS, but require a 2D output CRS.

**Requirement 5 crs/wcsServiceMetadata-outputCrs-unique:**

The list of `crsSupported` element(s) delivered in the `CrsMetadata` element of the response to a successful *GetCapabilities* request **shall** not contain any duplicates.

A client should not assume that a CRS list is delivered in some particular order.

**6.3 Modifications to DescribeCoverage**

None.

## 6.4 Modifications to *GetCoverage*

### 6.4.1 Modifications to the *GetCoverage* request

The *GetCoverage* request is extended with two optional parameters indicating the CRSs for subsetting coordinates and for the result coverage. If these parameters are omitted then all access will be performed in the coverage’s Native CRS (as reported in its Coverage Description).

Note This normally ensures that the coverage’s unchanged range set values are retrieved, assuming that no other operations than subsetting is performed in the *GetCoverage* request.

**Requirement 6 crs/getCoverage-crs:**

A *GetCoverage* request **shall** adhere to Figure 2, Table 4, and the XML schema defined for this CRS Extension whereby, in the XML request encoding, the *GetCoverage* `wcs:Extension` element **shall** contain exactly one *GetCoverageCrs* element.

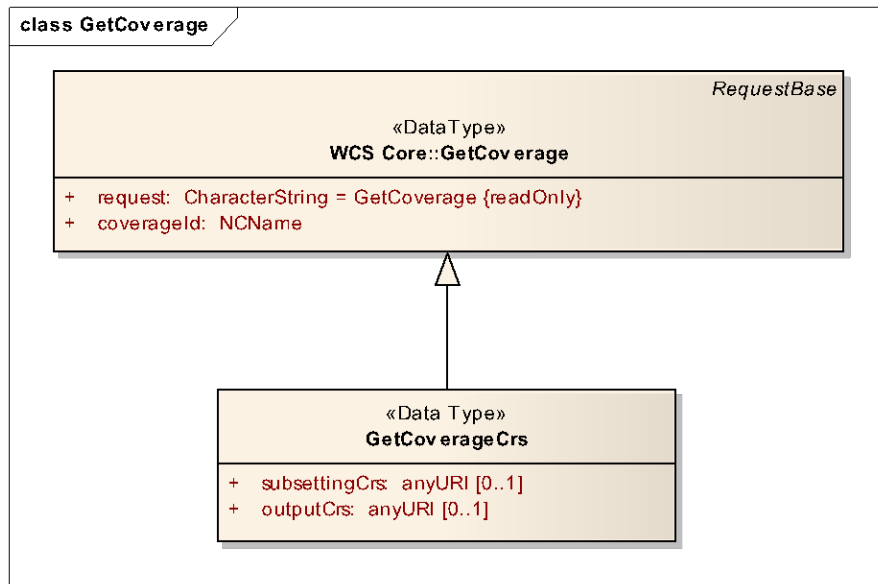


Figure 2 —*GetCoverage* UML diagram with *GetCoverageCrs* extension

Table 4 — Components of `CRS::GetCoverageCrs` structure

Name	Definition	Data type	Multiplicity
subsettingCrs	CRS Identifier indicating the CRS in which the request subsetting coordinates are expressed	anyURI	zero or one (optional)
outputCrs	CRS Identifier indicating the CRS of the result coverage	anyURI	zero or one (optional)

A *GetCoverage* request may contain an additional parameter, `subsettingCrs`, to indicate the CRS in which the subsetting coordinates passed are expressed. Admissible values for this parameter are URIs which resolve to exactly one CRS.



**Requirement 7 crs/getCoverage-subsettingCrs-default:**

If the `subsettingCrs` parameter is not present in a *GetCoverage* request then the request **shall** be evaluated as if a `subsettingCrs` parameter equal to the coverage's Native CRS (as provided in the Coverage Description) had been provided.

**Requirement 8 crs/getCoverage-subsettingCrs-uniqueCrsName:**

The `subsettingCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier.

**Requirement 9 crs/getCoverage-subsettingCrs-availableCrs:**

The `subsettingCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier described in the `crsSupported` list delivered in the `CrsMetadata` element of the Capabilities document of the server addressed.

Note As per WCS Core [OGC 09-110r3] every coverage has a Native CRS associated which will be contained in this list among potentially more CRSs; therefore, this list will never be empty as long as the server offers at least one coverage.

A *GetCoverage* request may contain an additional parameter, `outputCrs`, to indicate a particular CRS in which the client wants to receive the result coverage produced by the server. Admissible values for this parameter are URIs which resolve to exactly one CRS.

**Requirement 10 crs/getCoverage-outputCrs-default:**

If the `outputCrs` parameter is not present in a *GetCoverage* request then the request **shall** be evaluated as if an `outputCrs` parameter equal to the `subsettingCrs` parameter value assumed for this request's evaluation had been provided.

**Requirement 11 crs/getCoverage-outputCrs-uniqueCrsName:**

The `outputCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier.

**Requirement 12 crs/getCoverage-outputCrs-availableCrs:**

The `outputCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier described in the `crsSupported` list delivered in the `CrsMetadata` element of the Capabilities document of the server addressed.

Note As per WCS Core [OGC 09-110r3] every coverage has a Native CRS associated which will be in this list among potentially more CRSs; therefore, this list will never be empty as long as the server offers at least one coverage.

**6.4.2 Modifications to the *GetCoverage* response****Requirement 13 crs/getCoverage-response-crs:**

The contents of the response to a successful *GetCoverage* request containing an `outputCrs` parameter **shall** have, in the `gml:boundedBy` element, an `srsName` value identical to the `outputCrs` element in the request.

Example The following is a valid `srsName` specification:

```
<gml:boundedBy ... srsName="http://www.opengis.net/def/crs/EPSG/0/4326" ...>
```

Note If a request contains neither `subsettingCrs` nor `outputCrs` then, as per Requirement 10, the `outputCrs` likewise defaults to the coverage's Native CRS.

No statement is made in this specification about the interpolation method applied by the server in the course of the CRS transformation. A server is free to employ any interpolation method during CRS transformation, and information about the interpolation applied may not be accessible to the client. While the interpolation method applied may change over requests, and different interpolation methods may be applied to different range components of a particular coverage, a server should not change interpolation across the domain of coverage during execution of a single coverage CRS transformation.

Note The WCS Interpolation Extension allows controlling interpolation behaviour.

## 6.5 Exceptions

### Requirement 14 `crs/getCoverage-exception`:

When a WCS server encounters an error while evaluating the `outputCrs` parameter in a *GetCoverage* operation it **shall** return an exception report message chosen as indicated in Table 4. For each listed `exceptionCode`, the contents of the `locator` parameter value **shall** be as specified in the right column in Table 5.

Table 5 — Exception codes for use of `outputCrs`

<code>exceptionCode</code> value	HTTP code	Meaning of exception code	<code>locator</code> value
<code>NotACrs</code>	404	CRS parameter indicated is not a valid CRS	<code>subsettingCrs</code> request parameter value
<code>CrsMismatch</code>	404	CRS referenced by the parameter indicated is not compatible with the coverage (e.g., because the number of dimensions do not match, or because corresponding axes are incompatible, like latitude vs. height or longitude vs. time)	defaulting CRS parameter value in the request
<code>SubsettingCrs-NotSupported</code>	404	CRS indicated in the <code>subsettingCrs</code> parameter is not supported by this server	<code>subsettingCrs</code> request parameter value
<code>OutputCrs-NotSupported</code>	404	CRS indicated in the <code>outputCrs</code> parameter is not supported by this server	<code>outputCrs</code> request parameter value

## 6.6 Encodings

### 6.5.1 GET/KVP Encoding

#### Requirement 15 `crs/getCoverage-getkvp-subsettingcrs`:

In a *GetCoverage* request using the GET/KVP protocol, a `Crs::subsettingCrs` parameter shall be represented as

SUBSETTINGCRS=s  
 where s is a URL.

**Example** The following is a valid KVP fragment, assuming that the coverage addressed has a lat and long axis:

```
...& SUBSETTINGCRS=http://www.opengis.net/def/crs/EPSSG/0/4326 &...
```

**Requirement 16 crs/getCoverage-outputcrs:**

In a *GetCoverage* request using the GET/KVP protocol, a `Crs::outputCrs` parameter shall be represented as

OUTPUTCRS=s  
 where s is a URL.

**Example** The following is a valid KVP fragment, assuming that the coverage addressed has a lat and longaxis:

```
...& OUTPUTCRS=http://www.opengis.net/def/crs/EPSSG/0/4326 &...
```

## 6.5.2 XML/POST Encoding

**Requirement 17 crs/getCoverage-xmlpost:**

In a *GetCoverage* request using the XML/POST protocol, a `Crs::GetCoverageCrs` parameter **shall** be represented by a `crs::GetCoverageCrs` element.

**Example** The XML fragments below resemble the same example cases as the GET/KVP fragments above.

```
<wcs:Extension>
  <wscrs:outputCrs>
    http://www.opengis.net/def/crs/EPSSG/0/4326
  </wscrs:outputCrs>
</wcs:Extension>
```

```
<wcs:Extension>
  <wscrs:subsettingCrs>
    http://www.opengis.net/def/crs/EPSSG/0/4326
  </wscrs:subsettingCrs>
  <wscrs:outputCrs>
    http://www.opengis.net/def/crs/EPSSG/0/4326
  </wscrs:outputCrs>
</wcs:Extension>
```

## 6.5.3 SOAP Encoding

**Requirement 18 crs/getCoverage-soap:**

In a *GetCoverage* request using the SOAP protocol, a `Crs::GetCoverageCrs` parameter shall be represented by a `crs::GetCoverageCrs` element.

**Example** See previous subclause.

## 7 *Crs-gridded-coverage* requirements class

### 7.1 Overview

This Clause 7 establishes the CRS Extension conformance class *crs-gridded-coverage* which defines how Web Coverage Services shall deal with CRSs when Grid Coverages (in the definition of GML 3.2.1 [OGC 09-146r1]) are addressed.

Note As per GML 3.2.1 [OGC 09-146r1], this encompasses the coverage classes `GML::GridCoverage`, `GML::RectifiedGridCoverage`, and `GML::ReferenceableGridCoverage`.

#### **Requirement 19 *crs-gridded-coverage/dependency*:**

A WCS service implementing conformance class *crs-gridded-coverage* of this CRS Extension **shall** also implement the *crs* conformance class of this CRS Extension.

#### **Dependency:**

Clause 6, [http://www.opengis.net/doc/IS/WCS\\_service-extension\\_crs/1.0/Clause-6](http://www.opengis.net/doc/IS/WCS_service-extension_crs/1.0/Clause-6)

### 7.2 Modifications to *GetCapabilities*

#### **Requirement 20 *crs-gridded-coverage/extension-identifier*:**

A WCS service implementing conformance class *crs-gridded-coverage* of this CRS Extension **shall** include the following URI in the `Profile` element of the `ServiceIdentification` in a *GetCapabilities* response:

`http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-gridded-coverage`

### 7.3 Modifications to *GetCoverage*

For Grid Coverages, a conceptual evaluation model is specified. Implementations can choose to employ different algorithms as long as the result data are identical.

#### **Requirement 21 *crs-gridded-coverage/getCoverage-rangeSet*:**

The range values of the response to a successful *GetCoverage* request containing an `outputCrs` parameter **shall** be determined according to the following conceptual evaluation model: Take the bounding box of the request's subsetting parameter (or the full coverage bounding box if no subsetting was specified in the request); re-project this bounding box to the output CRS provided; generate the result image extent as the smallest bounding box around the reprojected subsetting box; fill this new image extent with pixel values re-projected from the coverage's Native CRS to the output CRS in a way that, for each axis, the smallest distance between any two reprojected grid points is used as offset (including all interpolation and resampling necessary) unless specified otherwise by further request parameters.

Note Request parameters dictating a different behaviour wrt. grid point offset are defined in the WCS Scaling & Interpolation extension specification.

## 8 *Crs-discrete-coverage* requirements class

### 8.1 Overview

This Clause 8 establishes the CRS Extension conformance class *crs-discrete-coverage* which defines how Web Coverage Services shall deal with CRSs when Discrete Coverages (in the definition of GML 3.2.1 [OGC 09-146r1]) are addressed.

Note As per GML 3.2.1 [OGC 09-146r1], the GML::DiscreteCoverage class consists of the coverage sub-classes GML::MultiPointCoverage, GML::MultiCurveCoverage, GML::MultiSurfaceCoverage, and GML::MultiSolidCoverage.

#### **Requirement 22 *crs-discrete-coverage/dependency*:**

A WCS service implementing conformance class *crs-discrete-coverage* of this CRS Extension **shall** also implement the *crs* conformance class of this CRS Extension.

#### **Dependency:**

Clause 6, [http://www.opengis.net/doc/IS/WCS\\_service-extension\\_crs/1.0/Clause-6](http://www.opengis.net/doc/IS/WCS_service-extension_crs/1.0/Clause-6)

### 8.2 Modifications to *GetCapabilities*

#### **Requirement 23 *crs-discrete-coverage/extension-identifier*:**

A WCS service implementing conformance class *crs-discrete-coverage* of this CRS Extension **shall** include the following URI in the `Profile` element of the `ServiceIdentification` in a *GetCapabilities* response:

```
http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-
discrete-coverage
```

Note Currently, no particular requirements are imposed. Further revisions of this standard may add ones, though.

### 8.3 Modifications to *GetCoverage*

For Discrete Coverages, a conceptual evaluation model is specified. Implementations can choose to employ different algorithms as long as the result data are identical.

#### **Requirement 24 *crs-gridded-coverage/getCoverage-rangeSet*:**

The range values of the response to a successful *GetCoverage* request containing an `outputCrs` parameter **shall** be determined according to the following conceptual evaluation model: Take the bounding box of the request's subsetting parameter (or the full coverage bounding box if no subsetting was specified in the request); re-project this bounding box to the output CRS provided; generate the result image extent as the smallest bounding box around the reprojected subsetting box; fill this new image extent with pixel values re-projected from the coverage's Native CRS to the output CRS in a way that, for each axis, the smallest distance between any two reprojected grid points is used as offset (including all interpolation and resampling necessary) unless specified otherwise by further request parameters.

Note Request parameters dictating a different behaviour wrt. grid point offset are defined in the WCS Scaling & Interpolation extension specification.

## Bibliography

- [1] ISO 8601:2004(E) *Data elements and interchange formats — Information interchange — Representation of dates and time*
- [2] [www.epsg.org](http://www.epsg.org)

## Annex A (normative)

### Abstract test suite

A CRS Extension implementation must satisfy the following system characteristics to be conformant with this specification.

Test identifiers below are relative to [http://www.opengis.net/spec/WCS/2.0/WCS\\_service-extension\\_crs/1.0/conf](http://www.opengis.net/spec/WCS/2.0/WCS_service-extension_crs/1.0/conf).

#### A.1 Conformance Test Class: *crs*

The OGC URI identifier of this conformance class is:

[http://www.opengis.net/spec/WCS/2.0/conf/WCS\\_service-extension\\_crs/1.0/conf/crs](http://www.opengis.net/spec/WCS/2.0/conf/WCS_service-extension_crs/1.0/conf/crs).

##### A.1.1 CRS extension identifier

**Test id:** `crs/extension-identifier:`

**Test Purpose:** A WCS service implementing conformance class *crs* of this CRS Extension **shall** include the following URI in the `Profile` element of the `ServiceIdentification` in a *GetCapabilities* response:  
`http://www.opengis.net/spec/WCS_service-extension_crs/1.0`

**Test method:** Determine the supported extension via a valid *GetCapabilities* request; check that the URI required is listed accordingly.

Test passes if all individual tests pass.

##### A.1.2 Capabilities

**Test id:** `crs/capabilities-crs:`

**Test Purpose:** The response to a successful *GetCapabilities* request **shall** adhere to Figure 1, Table 3, and the XML schema defined for this CRS Extension.

**Test method:** Send a valid *GetCapabilities* request to the server under test, check the result consists of the appropriate components, as defined in the places referenced.

Test passes if all individual tests pass.

##### A.1.3 CrsMetadataelement

**Test id:** `crs/wcsServiceMetadata-crs:`

**Test Purpose:** The response to a successful *GetCapabilities* request **shall** contain a `crs:CrsMetadata` element in the `wcs:Extension` element of `wcs:ServiceMetadata`.

**Test method:** Send a valid *GetCapabilities* request to the server under test, check the result consists of a `CrsMetadata` element.

Test passes if all individual tests pass.

#### A.1.4 Output CRS

**Test id:** `crs/wcsServiceMetadata-outputCrs:`

**Test Purpose:** The `crsSupported` item(s) delivered in the `CrsMetadata` element of the response to a successful *GetCapabilities* request **shall** consist of URLs, each one resolving to an instantiatable subtype of `GML : :AbstractCRS`.

**Test method:** Send a valid *GetCapabilities* request to the service under test. Check that the `crsSupported` item(s) delivered in the `CrsMetadata` element of the response consist of URLs, each one resolving to an instantiatable subtype of `GML : :AbstractCRS`.

Test passes if all individual tests pass.

#### A.1.5 Unique output CRS

**Test id:** `crs/wcsServiceMetadata-outputCrs-unique:`

**Test Purpose:** The list of `crsSupported` element(s) delivered in the `CrsMetadata` element of the response to a successful *GetCapabilities* request **shall not** contain any duplicates.

**Test method:** Send a valid *GetCapabilities* request to the service under test. Check that the `crsSupported` item(s) delivered in the `CrsMetadata` element of the response do not contain any duplicates.

Test passes if all individual tests pass.

#### A.1.6 GetCoverage CRS

**Test id:** `crs/getCoverage-crs:`

**Test Purpose:** A *GetCoverage* request **shall** adhere to Figure 2, Table 4, and the XML schema defined for this CRS Extension whereby, in the XML request encoding, the *GetCoverage* `wcs : Extension` element **shall** contain exactly one `GetCoverageCrs` element.

**Test method:** Send a valid *GetCoverage* request to server under test which conforms to the references in the requirement. Check that the response is not an exception.

Test passes if all individual tests pass.



### A.1.7 Service Metadata Output CRS

- Test id:** `crs/wcsServiceMetadata-outputCrs:`
- Test Purpose:** The `crsSupported` item(s) delivered in the `CrsMetadata` element of the response to a successful *GetCapabilities* request **shall** consist of URLs, each one resolving to an instantiatable subtype of `GML::AbstractCRS`.
- Test method:** Send a valid *GetCoverage* request without the `subsettingCrs` parameter to the service under test. Check that the response coverage is evaluated according to its Native CRS (as provided in the Coverage Description).
- Test passes if all individual tests pass.

### A.1.8 Unique GetCoverageSubsettingCrs CRS

- Test id:** `crs/getCoverage-subsettingCrs-uniqueCrsName:`
- Test Purpose:** The `subsettingCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier.
- Test method:** Send valid *GetCoverage* requests with a `subsettingCrs` parameter to server under test where the parameter contains
- a CRS Identifier (as provided in the `CrsMetadata` element of the Capabilities document);
  - two or more CRS Identifiers.
- Pass test if appropriate valid results or exceptions, resp., are delivered

### A.1.9 Available GetCoveragesubsettingCRS

- Test id:** `crs/getCoverage-subsettingCrs-availableCrs:`
- Test Purpose:** The `subsettingCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier described in the `crsSupported` list delivered in the `CrsMetadata` element of the Capabilities document of the server addressed.
- Test method:** Send valid *GetCoverage* requests with a `subsettingCrs` parameter to server under test where the parameter contains
- a CRS Identifier which is in the `crsSupportedlist` delivered in the `CrsMetadata` element of the Capabilities document of the server addressed;
  - a CRS Identifier which is not in the `crsSupportedlist`.

Pass test if appropriate valid results or exceptions, resp., are delivered.

### A.1.10 Default GetCoverageOutputCRS

**Test id:** `crs/getCoverage-outputCrs-default:`

**Test Purpose:** If the `outputCrs` parameter is not present in a *GetCoverage* request then the request **shall** be evaluated as if an `outputCrs` parameter equal to the `subsettingCrs` parameter value assumed for this request's evaluation had been provided.

**Test method:** Send a valid *GetCoverage* request without the `outputCrs` parameter to the service under test. Check that the response coverage is provided according to the `subsettingCrs` parameter value assumed for this request's evaluation had been provided.

Test passes if all individual tests pass.

### A.1.11 Unique GetCoverageOutputCRS

**Test id:** `crs/getCoverage-outputCrs-uniqueCrsName:`

**Test Purpose:** The `outputCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier.

**Test method:** Send valid *GetCoverage* requests with an `outputCrs` parameter to server under test where the parameter contains

- a CRS Identifier (as provided in the `CrsMetadata` element of the Capabilities document);
- two or more CRS Identifiers.

Pass test if appropriate valid results or exceptions, resp., are delivered.

### A.1.12 Available GetCoverageoutputCRS

**Test id:** `crs/getCoverage-outputCrs-availableCrs:`

**Test Purpose:** The `outputCrs` parameter in a *GetCoverage* request, if present, **shall** have as its value a CRS Identifier described in the `crsSupported` list delivered in the `CrsMetadata` element of the Capabilities document of the server addressed.

**Test method:** Send valid *GetCoverage* requests with an `outputCrs` parameter to server under test where the parameter contains

- a CRS Identifier which is in the `crsSupportedlist` delivered in the `CrsMetadata` element of the Capabilities document of the server

addressed;

- a CRS Identifier which is not in the `crsSupported` list.

Pass test if appropriate valid results or exceptions, resp., are delivered.

### A.1.13 GetCoverage Response CRS

**Test id:** `crs/getCoverage-response-crs:`

**Test Purpose:** The contents of the response to a successful *GetCoverage* request containing an `outputCrs` parameter **shall** have, in the `gml:boundedBy` element, an `srsName` value identical to the `outputCrs` element in the request.

**Test method:** Send a valid *GetCoverage* request with an `outputCrs` parameter to the service under test. Check that a `srsName` of the `gml:boundedBy` element of the response coverage is identical to the `outputCrs` element in the request.

Test passes if all individual tests pass.

### A.1.14 GetCoverage Exception

**Test id:** `crs/getCoverage-exception:`

**Test Purpose:** When a WCS server encounters an error while evaluating the `outputCrs` parameter in a *GetCoverage* operation it **shall** return an exception report message chosen as indicated in Table 4. For each listed `exceptionCode`, the contents of the `locator` parameter value **shall** be as specified in the right column in Table 5.

**Test method:** For each exception referenced in the requirement: Send an erroneous *GetCoverage* request to the server under test provoking this exception, as per its definition. Check for proper exception reporting. Pass test if all checks succeed.

Test passes if all individual tests pass.

### A.1.15 GetCoverageGetKVPSubsettingcrs

**Test id:** `crs/getCoverage-getkvp-subsettingcrs:`

**Test Purpose:** In a *GetCoverage* request using the GET/KVP protocol, a `Crs::subsettingCrs` parameter shall be represented as `SUBSETTINGCRS=s` where `s` is a URL.

**Test method:** Send a valid *GetCoverage* request in GET/KVP with the `Crs::subsettingCrs` parameter being represented as `SUBSETTINGCRS=s` where `s` is a URL. Check that the result is not an

exception.

Test passes if all individual tests pass.

### A.1.16 GetCoverageOutputCRS

**Test id:** `crs/getCoverage-outputcrs:`

**Test Purpose:** In a *GetCoverage* request using the GET/KVP protocol, a `Crs::outputCrs` parameter shall be represented as `OUTPUTCRS=s` where `s` is a URL.

**Test method:** Send a valid GET/KVP *GetCoverage* request with the `Crs::outputCrs` parameter being represented as `OUTPUTCRS=s` where `s` is a URL. Check that the result is not an exception.

Test passes if all individual tests pass.

### A.1.17 GetCoverage XML/POST

**Test id:** `crs/getCoverage-xmlpost:`

**Test Purpose:** In a *GetCoverage* request using the XML/POST protocol, a `Crs::GetCoverageCrs` parameter **shall** be represented by a `crs::GetCoverageCrs` element.

**Test method:** Send a valid XML/POST *GetCoverage* request with the `crs::GetCoverageCrs` element. Check that the result is not an exception.

Test passes if all individual tests pass.

### A.1.18 GetCoverage SOAP

**Test id:** `crs/getCoverage-soap:`

**Test Purpose:** In a *GetCoverage* request using the SOAP protocol, a `Crs::GetCoverageCrs` parameter shall be represented by a `crs::GetCoverageCrs` element.

**Test method:** Send a valid SOAP *GetCoverage* request with the `crs::GetCoverageCrs` element. Check that the result is not an exception.

Test passes if all individual tests pass.

## A.2 Conformance Test Class: *crs-gridded-coverage*

The OGC URI identifier of this conformance class is:

[http://www.opengis.net/spec/WCS/2.0/conf/WCS\\_service-extension\\_crs/1.0/conf/crs-gridded-coverage](http://www.opengis.net/spec/WCS/2.0/conf/WCS_service-extension_crs/1.0/conf/crs-gridded-coverage).

### A.2.1 CRS Gridded Coverage Dependency

- Test id:** **crs-gridded-coverage/dependency:**
- Test Purpose:** A WCS service implementing conformance class *crs-gridded-coverage* of this CRS Extension **shall** also implement the *crs* conformance class of this CRS Extension.
- Dependency:**  
Clause 6, [http://www.opengis.net/doc/IS/WCS\\_service-extension\\_crs/1.0/Clause-6](http://www.opengis.net/doc/IS/WCS_service-extension_crs/1.0/Clause-6)
- Test method:** Determine the supported dependency via a valid *GetCapabilities* request; check that the URI required is listed accordingly.
- Test passes if all individual tests pass.

### A.2.2 Gridded Coverage CRS

- Test id:** **crs-gridded-coverage/extension-identifier:**
- Test Purpose:** A WCS service implementing conformance class *crs-gridded-coverage* of this CRS Extension **shall** include the following URI in the `Profile` element of the `ServiceIdentification` in a *GetCapabilities* response:  
[http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs-gridded-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-gridded-coverage)
- Test method:** Determine the supported extension via a valid *GetCapabilities* request; check that the URI required is listed accordingly.
- Test passes if all individual tests pass.

### A.2.3 GetCoverage Range Set

- Test id:** **crs-gridded-coverage/getCoverage-rangeSet:**
- Test Purpose:** The range values of the response to a successful *GetCoverage* request containing an `outputCrs` parameter **shall** be determined according to the following conceptual evaluation model: Take the bounding box of the request's subsetting parameter (or the full coverage bounding box if no subsetting was specified in the request); re-project this bounding box to the output CRS provided; generate the result image extent as the smallest bounding box around the reprojected subsetting box; fill this new image extent with pixel values re-projected from the coverage's Native CRS to the output CRS in a way that, for each axis, the smallest distance between any two reprojected grid points is used as offset (including all interpolation and resampling necessary) unless specified otherwise by further request parameters.
- Test method:** Send valid *GetCoverage* requests to server under test
- without a `subsettingCrs` parameter;
  - with a `subsettingCrs` parameter contains a CRS Identifier (as

provided in the `CrsMetadata` element of the Capabilities document);

- without an `outputCrs` parameter;
- with an `outputCrs` parameter contains a CRS Identifier (as provided in the `CrsMetadata` element of the Capabilities document).

Check that these results are evaluated according to the specified model.

Test passes if all individual tests pass.

### A.3 Conformance Test Class: *crs-discrete-coverage*

The OGC URI identifier of this conformance class is:

[http://www.opengis.net/spec/WCS/2.0/conf/WCS\\_service-extension\\_crs/1.0/conf/crs-discrete-coverage](http://www.opengis.net/spec/WCS/2.0/conf/WCS_service-extension_crs/1.0/conf/crs-discrete-coverage).

#### A.3.1 CRS Discrete Coverage Dependency

**Test id:** `crs-discrete-coverage/dependency:`

**Test Purpose:** A WCS service implementing conformance class *crs-discrete-coverage* of this CRS Extension **shall** also implement the *crs* conformance class of this CRS Extension.

**Dependency:**

Clause 6, [http://www.opengis.net/doc/IS/WCS\\_service-extension\\_crs/1.0/Clause-6](http://www.opengis.net/doc/IS/WCS_service-extension_crs/1.0/Clause-6)

**Test method:** Determine the supported dependency via a valid *GetCapabilities* request; check that the URI required is listed accordingly.

Test passes if all individual tests pass.

#### A.3.2 Crs Discrete Coverage Identifier

**Test id:** `crs-discrete-coverage/extension-identifier:`

**Test Purpose:** A WCS service implementing conformance class *crs-discrete-coverage* of this CRS Extension **shall** include the following URI in the `Profile` element of the `ServiceIdentification` in a *GetCapabilities* response:  
[http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs-discrete-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-discrete-coverage)

**Test method:** Determine the supported extension via a valid *GetCapabilities* request; check that the URI required is listed accordingly.

Test passes if all individual tests pass.

#### A.3.3 Crs Discrete Coverage Range Set

**Test id:** `crs-gridded-coverage/getCoverage-rangeSet:`

**Test Purpose:** The range values of the response to a successful *GetCoverage* request containing an `outputCrs` parameter **shall** be determined according to the following conceptual evaluation model: Take the bounding box of the request's subsetting parameter (or the full coverage bounding box if no subsetting was specified in the request); re-project this bounding box to the output CRS provided; generate the result image extent as the smallest bounding box around the reprojected subsetting box; fill this new image extent with pixel values re-projected from the coverage's Native CRS to the output CRS in a way that, for each axis, the smallest distance between any two reprojected grid points is used as offset (including all interpolation and resampling necessary) unless specified otherwise by further request parameters.**crs-discrete-coverage/extension-identifier:**

A WCS service implementing conformance class *crs-discrete-coverage* of this CRS Extension **shall** include the following URI in the `Profile` element of the `ServiceIdentification` in a *GetCapabilities* response:  
[http://www.opengis.net/spec/WCS\\_service-extension\\_crs/1.0/conf/crs-discrete-coverage](http://www.opengis.net/spec/WCS_service-extension_crs/1.0/conf/crs-discrete-coverage)

**Test method:** Send valid *GetCoverage* requests to server under test

- without a `subsettingCrs` parameter;
- with a `subsettingCrs` parameter contains a CRS Identifier (as provided in the `CrsMetadata` element of the Capabilities document);
- without an `outputCrs` parameter;
- with an `outputCrs` parameter contains a CRS Identifier (as provided in the `CrsMetadata` element of the Capabilities document).

Check that these results are evaluated according to the specified model.

Test passes if all individual tests pass.

-- end of ATS --