Open Geospatial Consortium

Publication Date: 2014-02-26

Approval Date: 2014-01-17

Submission Date: 2013-08-20

Reference number of this Document: OGC 12-039

External Reference for this document: http://www.opengis.net/doc/IS/WCS_scaling_extension/1.0

Version: 1.0

Category: OGC® Interface Standard

Editor: Peter Baumann, Jinsongdi Yu

OGC® Web Coverage Service Interface Standard - Scaling Extension

Warning

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

Document type:	OGC Implementation Standard
Document subtype:	Interface
Document stage:	Approved
Document language:	English

License Agreement

Contents

Page

Tables

Page

i. Abstract

This document specifies parameters to the OGC Web Coverage Service (WCS) *GetCoverage* request which allow scaling of a coverage during its server-side processing in a *GetCoverage* request.

ii. Keywords

ogcdoc, wcs, scaling

iii. Terms and definitions

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, Inc.:

- Jacobs University Bremen
- Fuzhou University

v. Document Contributor Contact Points

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

vi. Changes to the OGC® Abstract Specification

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

vii. Future Work

Among the topics for future development are the following items:

- None defined right now

Foreword

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

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.

OGC® Web Coverage Service Interface Standard - Scaling Extension

1 Scope

This OGC Web Coverage Service (WCS) – Scaling Extension (in short: Scaling Extension) defines an extension to the WCS Core [OGC 09-110] for the scaling of grid coverages along one or more of its axes during a GetCoverage retrieval.

To this end, the GetCoverage request is extended with a parameter allowing to control output resolution of the grid coverage returned. The scale parameter always is expressed relative to the grid coverage's Grid CRS, that is: in array indices.

2 Conformance

This document establishes the following requirements and conformance classes:

- *scaling*, of URI http://www.opengis.net/spec/WCS_service-extension_scaling/1.0/req/scaling; the corresponding conformance class is *scaling*, with URI http://www.opengis.net/spec/WCS_service-extension_scaling/1.0/conf/scaling.

  This is the mandatory core conformance class of this extension.

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_scaling/1.0/req, conformance test URIs are relative to http://www.opengis.net/spec/WCS_service-extension_scaling/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 Scaling 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_scaling/1.0, the document has OGC URI http://www.opengis.net/doc/IS/WCS_service-extension_scaling/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/scaling/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

Range subsetting conformance class	**Dependency document**	**Dependency conformance class**
scaling	OGC 09-146, *GML 3.2.1 Application Schema for Coverages*, version 1.0	*gml-coverage*
	OGC 09-110, *OGC® Web Coverage Service 2.0 Interface Standard - Core*, version 2.0	*core*

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 “□“ indicates that the following term is defined in this Clause.

4.1 Scaling (of a grid coverage)

linear transformation that changes resolution of a grid coverage by a scale factor that may or may not be the same in all directions

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	http://www.w3.org/2001/XMLSchema	XML Schema namespace
gml	http://www.opengis.net/gml/3.2	GML 3.2.1
gmlcov	http://www.opengis.net/gmlcov/1.0	GML Application Schema for Coverages 1.0
wcs	http://www.opengis.net/wcs/2.0	WCS 2.0 Core
scal	http://www.opengis.net/wcs/scaling/1.0	WCS 2.0 Scaling 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 *Scaling* requirements class

6.1 Overview

Clients and servers supporting this scaling requirements class shall support retrieval of scaled versions of grid coverages offered by a WCS server.

This Clause 6 establishes the Scaling Extension core requirements class, *scaling*.

6.2 Modifications to *GetCapabilities*

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

Requirement 1 extension-identifier:
A WCS service implementing requirements class *scaling* of this Scaling 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_scaling/1.0/conf/scaling`

6.3 Modifications to *DescribeCoverage*

None.

6.4 Modifications to *GetCoverage*

6.4.1 Modifications to the *GetCoverage* request

An additional request parameter indicates the scale factor. A scale factor can be given as

- a single number indicating the scale factor to be applied to all axes of the coverage addressed, or
- a list of coverage axes and their individual scale factors, or
- a list of coverage axes and their individual target size, or
- a target domain to which the coverage addressed is to be scaled.

Note It is recommended to use the target size in a scaling request, as scale factors sometimes can lead to numerical errors in the result domain extent, which can confuse clients.

No statement is made in this specification about the interpolation method applied by the server in the course of the scaling operation. A server is free to employ any interpolation method during scaling, 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 scaling operation.

Note The WCS Interpolation Extension allows controlling interpolation behaviour.

Requirement 2 getCoverage-scaling:
A *GetCoverage* request **shall** adhere to Figure 1, Table 3 through Table 6, and the XML schema defined for this Scaling Extension, containing zero or one `Scaling` element.

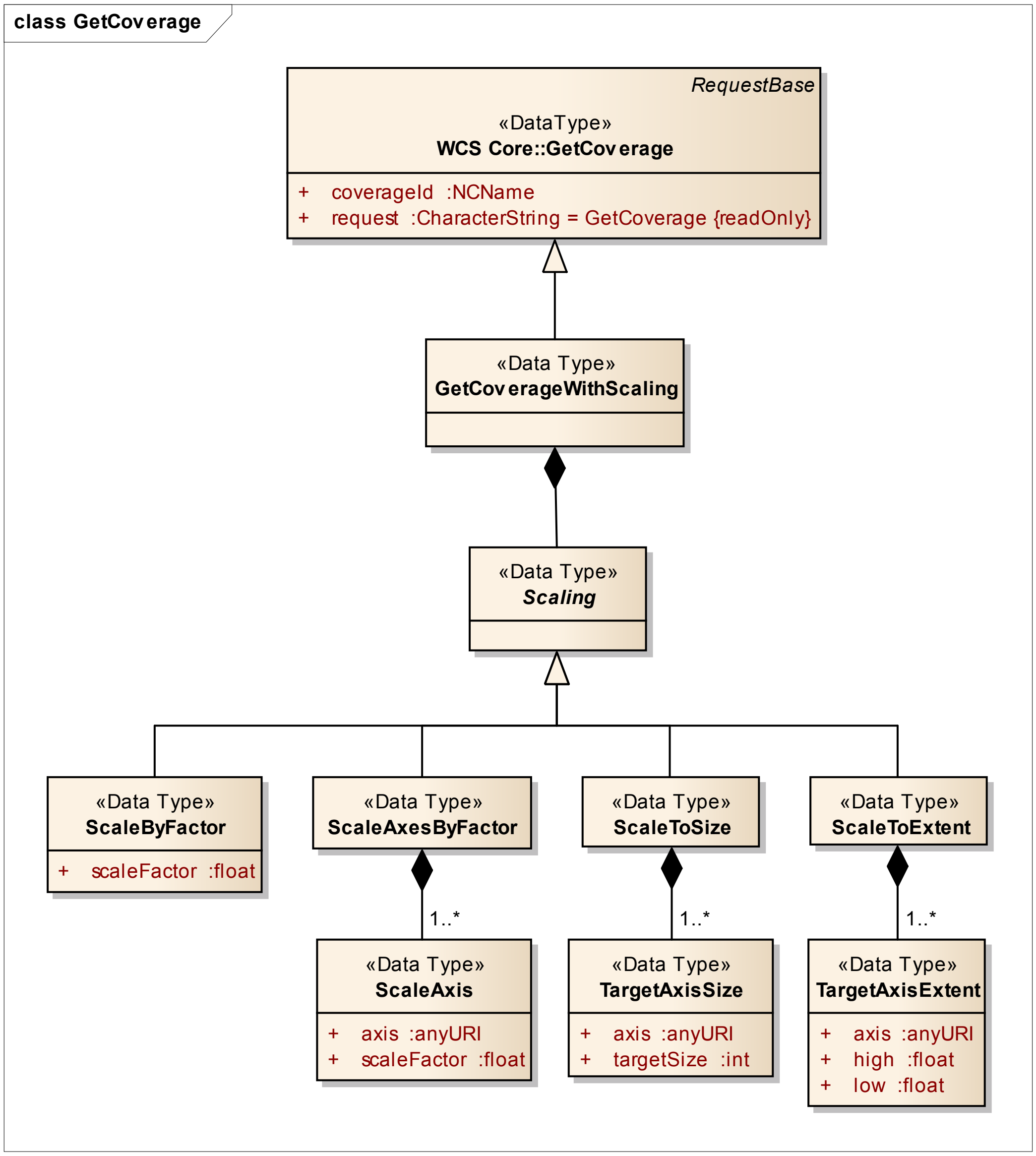

Figure 1 — `GetCoverage` with *scaling* support UML diagram

Table 3 — Components of `Scal::ScaleByFactor` structure

Name	Definition	Data type	Multiplicity
`scaleFactor`	Scale factor to be applied to all coverage axes	`float`	zero or one (optional)

Requirement 3 grid-coverage:
The coverage identifier in a *GetCoverage* request **shall** address a coverage which is a Grid-Coverage, RectifiedGridCoverage, ReferenceableGridCoverage, or a subtype of one of these.

Requirement 4 getCoverage-mutually-exclusive:
A *GetCoverage* request containing a scaling operation **shall** contain exactly one of

`Scal::scaleByFactor`, `Scal::scaleAxesByFactor`, `Scal::scaleToSize`, and `Scal::scaleToExtent`.

Alternative 1: Scaling can be expressed by indicating a scale factor to be applied uniformly along all axes.

Requirement 5 getCoverage-scale-by-factor-positive:
The `Scal::scaleByFactor` parameter in a *GetCoverage* request, if present, **shall** have a positive float value.

Note A value of 1.0 leaves the coverage unscaled, a value between 0 and 1 scales down (reduces target domain), a value greater than 1 scales up (enlarges target domain).

Alternative 2: Scaling can be expressed by indicating scale factors individually per axis; axes not mentioned will remain unaffected.

Requirement 6 getCoverage-scale-axes-by-factor-present:
The `Scal::scaleAxesByFactor` parameter in a *GetCoverage* request, if present, **shall** consist of a non-empty sequence of `Scal::scaleAxis` elements with a structure as defined in Table 4.

Note In literature, this is sometimes referred to as *anisotropic scaling*.

Table 4 — Components of `Scal::ScaleAxis` structure

Name	Definition	Data type	Multiplicity
`sxis`	Axis to which scaling is to be applied	`anyURI`	one (mandatory)
`scaleFactor`	Scale factor for this axis	`Float`	one (mandatory)

Note See Subclause 6.6 for examples.

Alternative 3: Scaling can be expressed by indicating, per domain axis given, the target domain extent to which the coverage should be rescaled; this leaves the lower bound of the grid unchanged while the upper bound is adjusted accordingly. Axes not mentioned remain unaffected.

Requirement 7 getCoverage-scale-to-size-positive:
The `targetSize` of every `Scal::targetAxisSize` parameter in a *GetCoverage* request **shall** have a positive value.

Requirement 8 getCoverage-scale-to-size-existing-axis:
Each `Scal::targetAxisSize` parameter in a *GetCoverage* request **shall** contain an `axis` element which is identical to the `axisAbbrev` element of some domain axis in the Grid CRS of the coverage addressed.

Requirement 9 getCoverage-scale-to-size-unique-axis:
The `Scal::scaleToSize` parameter in a *GetCoverage* request, if present, **shall** contain no two `Scal::targetAxisSize` elements with identical `axis` names.

Table 5 — Components of `Scal::TargetAxisSize` structure

Name	Definition	Data type	Multiplicity
`axis`	Axis to which scaling is to be applied	`anyURI`	one (mandatory)
`targetSize`	Target size of the coverage in the axis indicated	`int`	one (mandatory)

Note See Subclause 6.6 for examples.

Alternative 4: Scaling can be expressed by indicating, per domain axis given, the target domain to which the coverage should be rescaled. Axes not mentioned remain unchanged.

Requirement 10 getCoverage-scale-to-extent:
The `Scal::scaleToExtent` parameter in a *GetCoverage* request, if present, **shall** have a structure as defined in Table 6.

Table 6 — Components of `Scal::TargetAxisExtent structure`

Name	Definition	Data type	Multiplicity
`axis`	Axis to which scaling is to be applied	`anyURI`	one (mandatory)
`low`	Target lower bound of domain extent in this axis	`int`	one (mandatory)
`high`	Target upper bound of domain extent in this axis	`int`	one (mandatory)

Requirement 11 getCoverage-scale-to-extent-existing-axis:
The `Scal::TargetAxisExtent` parameter in a *GetCoverage* request, if present, **shall** contain an `axis` element which is identical to the identifier of some domain axis in the Grid CRS of the coverage addressed.

6.4.2 Modifications to the *GetCoverage* response

The scaling parameter effectuates that a coverage is returned which has

- a domain obtained by scaling the original coverage's domain according to the scaling parameters;

- a range type which is identical to the original coverage's range type;

- a range set where the elements are rescaled, possibly involving resampling and interpolation.

Note How the new range values are derived is left open, but it should resemble some interpolation of the original value and its neighborhood. The server is not required to disclose the interpolation method applied, but should normally use, per coverage, one and the same method upon subsequent scaling requests executed against this coverage. The WCS Interpolation Extension [1] defines how clients can influence interpolation behavior, provided the server supports this Interpolation Extension.

Requirement 12 getCoverage-scale-by-factor-result:
The response to a successful *GetCoverage* request scaling by scale factor f **shall** be identical to that of a request with scale vector $s = (s_1,\ldots,s_d)$ where $s_i = f$ for all $1 \leq i \leq d$ and d is the dimension of the original coverage.

Requirement 13 getCoverage-scale-axes-by-factor-result:
Let c be a coverage with
dimension d,
grid domain $X = \{l_1,\ldots,h_1\} \times\ldots\times \{l_d,\ldots,h_d\}$, and
range set $R = \{ (v,p) \mid \text{where } p = (x_1,..,x_d) \text{ and } l_i \leq x_i \leq h_i \text{ for } 1 \leq i \leq d \}$;
let further $s=(s_1,\ldots,s_d)$ be a valid *GetCoverage* request scale vector.
Then,
the response coverage to a successful *GetCoverage* request scaling coverage c by scale vector s **shall** have
dimension d,
Grid domain $X' = \{ \text{floor}(l_1 / s_1),\ldots, \text{floor}(h_1 / s_1) \}$
$\times\ldots\times$
$\{ \text{floor}(l_d / s_d),\ldots, \text{floor}(h_d / s_d) \}$, and
range set $R' = \{ (v',p') \mid p' = (x_1',..,x_d') □ X'$ and
v' is a function of the corresponding original location p
and possibly a neighborhood of p },
where any eventual subsetting of the coverage is done before this scale operation.

Note The above dimension implies that a scale operation cannot make reference to axis directions that are “sliced away” by a WCS Core subsetting operation. Note that “directions” in a coverage can be identified through different axes, such as “latitude” (using geo coordinates) and “j” (using iteger grid coordinates), but still mean the same “direction” in the sense of this discussion: slicing away “latitude”, in this example, disqualifies “j” from getting scaled.

Example The following list shows sample 2-D coverage grid domains and the output coverage’s grid domain when scaled uniformly by a factor of 2.0:

[0:99,0:199]	[0:49,0:99]
[100:199,100:199]	[50:99,50:99]
[-10:10,-20:20]	[-5:5,-10:10]

Requirement 14 getCoverage-scale-to-size-result:
Let c be a coverage with
dimension d,
grid domain $X = \{l_1,\ldots,h_1\} \times\ldots\times \{l_d,\ldots,h_d\}$,
range set $R = \{ (v,p) \mid \text{where } p = (x_1,..,x_d) \text{ and } l_i \leq x_i \leq h_i \text{ for } 1 \leq i \leq d \}$;
let further $s=(s_1,\ldots,s_d)$ be a valid *GetCoverage* request scale vector.
Then,
the response coverage to a successful *GetCoverage* request scaling coverage c by scale vector s **shall** have

dimension d,
Grid domain $X' = \{ l_1, \dots, l_1 + s_1 - 1 \,) \}$
$\times \dots \times$
$\{ l_d, \dots, l_d + s_d - 1 \}$,
range set $R' = \{ (v',p') \mid p' = (x_1',..,x_d') \,\square\, X'$ and
v' is a function of the corresponding original location p
and possibly a neighborhood of p },
where any eventual subsetting of the coverage is done before this scale operation.

Note This implies that axes not addressed by the scale factor will not be scaled at all (in other words, their scale factor is 1).

Example The following list shows sample 2-D coverage grid domains and the output coverage's grid domain when scaled uniformly to size 500:

[0:99,0:199]	[0:499,0:499]
[100:199,100:199]	[100:599,100:599]
[-10:10,-20:20]	[-10:489,-20:479]

Requirement 15 getCoverage-scale-to-extent-result:
Let c be a coverage with
dimension d,
grid domain $X = \{l_1,\dots,h_1\} \times\dots\times \{l_d,\dots,h_d\}$,
range set $R = \{ (v,p) \mid$ where $p = (x_1,..,x_d)$ and $l_i \leq x_i \leq h_i$ for $1 \leq i \leq d \}$;
let further s={ (a,sl,sh) } be a valid *GetCoverage* request scale interval set with each a_i being the identifier of a Grid axis in c, and $sl_i \leq sh_i$ being target domain intervals.
Then,
the response coverage to a successful *GetCoverage* request scaling coverage c by scale vector s **shall** have
dimension d,
Grid domain X' of dimension d where the domain extent is the same as in c for all those axes not mentioned in s, and set to $[sl_i,\dots,sh_i]$ for all axes a_i occurring in s,
range set $R' = \{ (v',p') \mid p' = (x_1',..,x_d') \,\square\, X'$ and
v' is a function of the corresponding original location p
and possibly a neighborhood of p },
where any eventual subsetting of the coverage is done before this scale operation.

If no `Scal::Scaling` parameter is present in the *GetCoverage* request then the coverage will be delivered unscaled, as per WCS Core [OGC 09-110].

Requirement 16 getCoverage-unscaled:
The response to a successful *GetCoverage* request containing no `Scal::Scaling` parameter **shall** consist of a coverage with no scaling applied to the original coverage.

The type of the coverage is preserved by the scaling operation.

Requirement 17 getCoverage-keep-type:
The response to a successful *GetCoverage* request containing a `Scal::Scaling` parameter **shall** consist of a coverage whose type is not changed by the scaling operation.

Note Other *GetCoverage* parameters might change the output coverage type indeed.

6.5 Exceptions

Requirement 18 getCoverage-exception:
When a WCS server encounters an error while evaluating a **scaleFactor** or **scaleExtent** parameter in a *GetCoverage* operation it **shall** return an exception report message chosen as indicated in Table 7 with a locator parameter value as specified in the right column of Table 7 for each exceptionCode listed.

Table 7 — Exception codes for use of scaleFactor and scaleExtent

exceptionCode value	**HTTP code**	**Meaning of exception code**	**locator value**
InvalidCoverageType	404	Coverage addressed is not a grid coverage	Coverage identifier
InvalidScaleFactor	404	Scale factor passed is not valid (no number or less than or equal to zero)	**scaleFactor** request parameter value
InvalidExtent	404	Extent interval passed has upper bound smaller than lower bound	**high** request parameter value
ScaleAxisUndefined	404	CRS axis indicated is not an axis occurring in this coverage	**scaleFactor** or **scaleExtent** request parameter value

6.6 Encodings

6.5.1 GET/KVP Encoding

Requirement 19 getCoverage-getkvp-scale-by-factor:
In a *GetCoverage* request using the GET/KVP protocol, a Scal::ScaleByFactor parameter **shall** be represented as

SCALEFACTOR=s

where s is a scaleFactor represented as the ASCII representation of a positive floating-point number.

Requirement 20 getCoverage-getkvp-scale-axes-by-factor:
In a *GetCoverage* request using the GET/KVP protocol, a Scal::ScaleAxesByFactor parameter containing n>0 axis scaling specifications **shall** be represented as

SCALEAXES=a_1(s_1),…,a_n(s_n)

where, for 1≤i≤n,

- a_i is an axis abbreviation;
- s_i is a scaleFactor expressed as the ASCII representation of a positive floating-point number.

Requirement 21 getCoverage-getkvp-scale-to-size:
In a *GetCoverage* request using the GET/KVP protocol, a Scal::ScaleToSize parameter

containing n>0 axis extent specifications **shall** be represented as

SCALESIZE=a_1(s_1),…,a_n(s_n)

where, for $1 \leq i \leq n$,
- a_i is an axis abbreviation;
- s_i are `sizes`.

Requirement 22 getCoverage-getkvp-scale-to-extent:
In a *GetCoverage* request using the GET/KVP protocol, a `Scal::ScaleToExtent` parameter containing n>0 axis extent specifications **shall** be represented as

SCALEEXTENT=a_1(lo_1:hi_1),…,a_n(lo_n:hi_n)

where, for $1 \leq i \leq n$,
- a_i is an axis abbreviation;
- lo_i and hi_i are `low` and `high`, respectively, each of them represented as either a string, enclosed in double quotes, or a number.

Example Below are three individually (not in combination) valid KVP fragments, assuming that the Grid CRS of the (gridded) coverage addressed contains the axes mentioned:

```
…& SCALEFACTOR=2.0 &…

…& SCALEAXES=i(3.5),j(3.5),k(2.0) &…

…& SCALESIZE=i(1000),j(1000),k(10) &…

…& SCALEEXTENT=i(10:20),j(20:30) &…
```

6.5.2 XML/POST Encoding

Requirement 23 getCoverage-xmlpost:
In a *GetCoverage* request using the XML/POST protocol, the following mapping **shall** be used:
- Alternative 1 **shall** be encoded by a `scal:ScaleByFactor` element;
- Alternative 2 **shall** be encoded by a `scal:ScaleAxesByFactor` element;
- Alternative 3 **shall** be encoded by a `scal:ScaleToSize` element;
- Alternative 4 **shall** be encoded by a `scal:ScaleToExtent` element.

Note The alternatives refer to Clause 6.4.

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

```
<scal:ScaleByFactor>
  <scal:scaleFactor>2.0</scal:scaleFactor>
</scal:ScaleByFactor>

<scal:ScaleAxesByFactor>
  <scal:ScaleAxis>
    <scal:axis>i</scal:axis>
    <scal:scaleFactor>3.5</scal:scaleFactor>
  </scal:ScaleAxis>
  <scal:ScaleAxis>
    <scal:axis>j</scal:axis>
```

```
      <scal:scaleFactor>3.5</scal:scaleFactor>
    </scal:ScaleAxis>
    <scal:ScaleAxis>
      <scal:axis>k</scal:axis>
      <scal:scaleFactor>2.0</scal:scaleFactor>
    </scal:ScaleAxis>
  </scal:ScaleAxesByFactor>

  <scal:ScaleToSize>
    <scal:TargetAxisSize>
      <scal:axis>i</scal:axis>
      <scal:targetSize>1000</scal:targetSize>
    </scal:TargetAxisSize>
    <scal:TargetAxisSize>
      <scal:axis>j</scal:axis>
      <scal:targetSize>1000</scal:targetSize>
    </scal:TargetAxisSize>
    <scal:TargetAxisSize>
      <scal:axis>k</scal:axis>
      <scal:targetSize>10</scal:targetSize>
    </scal:TargetAxisSize>
  </scal:ScaleToSize>

  <scal:ScaleToExtent>
    <scal:TargetAxisExtent>
      <scal:axis>i</scal:axis>
      <scal:low>10</scal:low>
      <scal:high>20</scal:high>
    </scal:TargetAxisExtent>
    <scal:TargetAxisExtent>
      <scal:axis>j</scal:axis>
      <scal:low>20</scal:low>
      <scal:high>30</scal:high>
    </scal:TargetAxisExtent>
  </scal:ScaleToExtent>
```

6.5.3 SOAP Encoding

Requirement 24 getCoverage-soap:
In a *GetCoverage* request using the SOAP protocol, scaling parameters **shall** be mapped as in Requirement 23 above.
Dependency: [09-149r1] (http://www.opengis.net/spec/WCS_protocol-binding_soap/1.0/soap)

Example   See previous subclause.

Bibliography

[1] OGC 12-049, *OGC® WCS Extension – Interpolation*, version 1.0.

Annex A
(normative)

Abstract test suite

A Scaling 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_scaling/1.0/conf.

A.1 Conformance Test Class: *scaling*

The OGC URI identifier of this conformance class is: http://www.opengis.net/spec/WCS/2.0/conf/WCS_service-extension_scaling/1.0/conf/scaling.

A.1.1 Scaling/extension identifier

Test id:	**extension-identifier:**
Test Purpose:	A WCS service implementing requirements class *scaling* of this Scaling 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_scaling/1.0/conf/scaling`
Test method:	Send a *GetCapabilities* request to server under test, verify that the response contains a `Profile` element with said URI. Test passes if result is as expected.

A.1.2 scaling/getCoverage scaling

Test id:	**getCoverage-scaling:**
Test Purpose:	A *GetCoverage* request **shall** adhere to Figure 1, Table 3 through Table 6, and the XML schema defined for this Scaling Extension, containing zero or one `Scaling` element.
Test method:	Send *GetCoverage* requests testing server response on the cases distinguished in said reference. Check proper response. Test passes if expected result is delivered.

A.1.3 grid coverage

Test id:	**grid-coverage:**
Test Purpose:	The coverage identifier in a *GetCoverage* request **shall** address a coverage which is a GridCoverage, RectifiedGridCoverage, ReferenceableGridCov-

erage, or a subtype of one of these.

Test method: Send *GetCoverage* requests to the service under test, check whether responses are adequate. Exercise tests for each of the following situations:

- with the coverage identifier of a GridCoverage. Verify that request succeeds;
- with the coverage identifier of a RectifiedCoverage. Verify that request succeeds;
- with the coverage identifier of a ReferenceableCoverage. Verify that request succeeds.

Test passes if expected result is delivered.

A.1.4 getCoverage mutually exclusive

Test id: **getCoverage-mutually-exclusive:**
Test Purpose: A *GetCoverage* request containing a scaling operation **shall** contain exactly one of `Scal::scaleByFactor`, `Scal::scaleAxesByFactor`, `Scal::scaleToSize`, and `Scal::scaleToExtent`.

Test method: Send *GetCoverage* requests to the service under test containing:

- No scaling parameter. Verify that request succeeds.
- a `Scal::scaleByFactor`. Verify that request succeeds.
- a `Scal::scaleAxesByFactor`. Verify that request succeeds.
- a `Scal::scaleToSize`. Verify that request succeeds.
- a `Scal::scaleToExtent`. Verify that request succeeds.
- more than one scaling element. Verify that request fails.

Test passes if expected result is delivered.

A.1.5 getCoverage scale by factor positive

Test id: **getCoverage-scale-by-factor-positive:**
Test Purpose: The `Scal::scaleByFactor` parameter in a *GetCoverage* request, if present, **shall** have a positive float value.

Test method: Send *GetCoverage* requests to the service under test with the `Scal::scaleByFactor` parameter containing:

- a positive float value. Verify that request succeeds.
- 0. Verify that request fails.
- a random negative value. Verify that request fails.

Test passes if result is as expected.

A.1.6 getCoverage scale axes by factor present

Test id: **getCoverage-scale-axes-by-factor-present:**

Test Purpose: The `Scal::scaleAxesByFactor` parameter in a *GetCoverage* request, if present, **shall** consist of a non-empty sequence of `Scal::scaleAxis` elements with a structure as defined in Table 4.

Test method: Send *GetCoverage* requests testing server response on the cases distinguished in said reference. Check proper response.

Test passes if expected result is delivered.

A.1.7 getCoverage scale to size positive

Test id: **getCoverage-scale-to-size-positive:**

Test Purpose: The `targetSize` of every `Scal::targetAxisSize` parameter in a *GetCoverage* request **shall** have a positive value.

Test method: Send *GetCoverage* requests to the service under test with a `Scal::targetAxisSize` parameter containing:

- a positive float value. Verify that request succeeds.
- 0. Verify that request fails.
- a random negative value. Verify that request fails.

Test passes if expected result is delivered.

A.1.8 getCoverage scale to size existing axis

Test id: **getCoverage-scale-to-size-existing-axis:**

Test Purpose: Each `Scal::targetAxisSize` parameter in a *GetCoverage* request **shall** contain an `axis` element which is identical to the `axisAbbrev` element of some domain axis in the Grid CRS of the coverage addressed.

Test method: Send *GetCoverage* requests to the service under test with a `Scal::targetAxisSize` parameter containing an `axis` element which is:

- a domain axis in the Grid CRS of the coverage addressed. Verify that request succeeds.
- a domain axis not in the Grid CRS of the coverage addressed. Verify that request fails.

Test passes if expected result is delivered.

A.1.9 getCoverage scale to size unique axis

Test id: **getCoverage-scale-to-size-unique-axis:**

Test Purpose: The `Scal::scaleToSize` parameter in a *GetCoverage* request, if present, **shall** contain no two `Scal::targetAxisSize` elements with identical `axis` names.

Test method: Send *GetCoverage* requests to the service under test, with a `Scal::scaleToSize` parameter contains:

- two `Scal::targetAxisSize` elements with different `axis` names. Verify that request succeeds.
- two `Scal::targetAxisSize` elements with identical `axis` names. Verify that request fails.

Test passes if expected result is delivered.

A.1.10 getCoverage scale to extent

Test id: **getCoverage-scale-to-extent:**

Test Purpose: The `Scal::scaleToExtent` parameter in a *GetCoverage* request, if present, **shall** have a structure as defined in Table 6.

Test method: Send *GetCoverage* requests testing server response on the cases distinguished in said reference. Check proper response.

Test passes if expected result is delivered.

A.1.11 getCoverage scale to extent existing axis

Test id: **getCoverage-scale-to-extent-existing-axis:**

Test Purpose: The `Scal::TargetAxisExtent` parameter in a *GetCoverage* request, if present, **shall** contain an `axis` element which is identical to the identifier of some domain axis in the Grid CRS of the coverage addressed.

Test method: Send *GetCoverage* requests to the service under test, with a `Scal::TargetAxisExtent` parameter contains an `axis` element which is:

- a domain axis in the Grid CRS of the coverage addressed. Verify that request succeeds.
- a domain axis not in the Grid CRS of the coverage addressed. Verify that request fails.

Test passes if expected result is delivered.

A.1.12 getCoverage scale by factor result

Test id: **getCoverage-scale-by-factor-result:**

Test Purpose: The response to a successful *GetCoverage* request scaling by scale factor f **shall** be identical to that of a request with scale vector $s = (s_1,\ldots,s_d)$ where $s_i = f$ for all $1 \le i \le d$ and d is the dimension of the original coverage.

Test method: Send separate *GetCoverage* requests to the service under test with:

- a scale factor f, resulting in response d_1;
- scale vector $s = (s_1,\ldots,s_d)$ where $s_i = f$ for all $1 \le i \le d$ and d is the dimension of the original coverage, resulting in response d_2;
- check if d_1 and d_2 are equal.

Test passes if expected result is delivered.

A.1.13 getCoverage scale axes by factor result

Test id: **getCoverage-scale-axes-by-factor-result:**

Test Purpose: Let c be a coverage with
dimension d,
grid domain $X = \{l_1,\ldots,h_1\} \times\ldots\times \{l_d,\ldots,h_d\}$, and
range set $R = \{ (v,p) \mid \text{where } p = (x_1,..,x_d) \text{ and } l_i \le x_i \le h_i \text{ for } 1 \le i \le d \}$;
let further $s=(s_1,\ldots,s_d)$ be a valid *GetCoverage* request scale vector.
Then,
the response coverage to a successful *GetCoverage* request scaling coverage c by scale vector s **shall** have
dimension d,
Grid domain $X' = \{ \text{floor}(l_1 / s_1),\ldots, \text{floor}(h_1 / s_1) \}$
$\times\ldots\times$
$\{ \text{floor}(l_d / s_d),\ldots, \text{floor}(h_d / s_d) \}$, and
range set $R' = \{ (v',p') \mid p' = (x_1',..,x_d') \in X'$ and
v' is a function of the corresponding original location p
and possibly a neighborhood of p $\}$,
where any eventual subsetting of the coverage is done before this scale operation.

Test method: Send a *GetCoverage* request to the service under test using a scale vector, check that response is correct.

Test passes if result is as expected.

A.1.14 getCoverage scale to size result

Test id: **getCoverage-scale-to-size-result:**

Test Purpose: Let c be a coverage with
dimension d,
grid domain $X = \{l_1,\ldots,h_1\} \times\ldots\times \{l_d,\ldots,h_d\}$,
range set $R = \{ (v,p) \mid \text{where } p = (x_1,..,x_d) \text{ and } l_i \leq x_i \leq h_i \text{ for } 1 \leq i \leq d \}$;
let further s=($s_1,\ldots,s_d$) be a valid *GetCoverage* request scale vector.
Then,
the response coverage to a successful *GetCoverage* request scaling coverage c by scale vector s **shall** have
dimension d,
Grid domain $X' = \{ l_1, \ldots, l_1 + s_1 \text{ -1 }) \}$
$\times\ldots\times$
$\{ l_d, \ldots, l_d + s_d \text{ -1 } \}$,
range set $R' = \{ (v',p') \mid p' = (x_1',..,x_d')$ □ X' and
v' is a function of the corresponding original location p
and possibly a neighborhood of p },
where any eventual subsetting of the coverage is done before this scale operation.

Test method: Send a *GetCoverage* request to the service under test using scale to size, check that response is correct.

Test passes if result is as expected.

A.1.15 getCoverage scale to extent result

Test id: **getCoverage-scale-to-extent-result:**

Test Purpose: Let c be a coverage with
dimension d,
grid domain $X = \{l_1,\ldots,h_1\} \times\ldots\times \{l_d,\ldots,h_d\}$,
range set $R = \{ (v,p) \mid \text{where } p = (x_1,..,x_d) \text{ and } l_i \leq x_i \leq h_i \text{ for } 1 \leq i \leq d \}$;
let further s={ (a,sl,sh) } be a valid *GetCoverage* request scale interval set with each a_i being the identifier of a Grid axis in c, and $sl_i \leq sh_i$ being target domain intervals.
Then,
the response coverage to a successful *GetCoverage* request scaling coverage c by scale vector s **shall** have
dimension d,
Grid domain X' of dimension d where the domain extent is the same as in c for all
those axes not mentioned in s, and set to [$sl_i,\ldots,sh_i$] for all axes a_i occurring in s,
range set $R' = \{ (v',p') \mid p' = (x_1',..,x_d')$ □ X' and
v' is a function of the corresponding original location p
and possibly a neighborhood of p },

where any eventual subsetting of the coverage is done before this scale operation.

Test method: Send a *GetCoverage* request to the service under test using scale to extent, check that response is correct.

Test passes if result is as expected.

A.1.16 getCoverage unscaled

Test id: **getCoverage-unscaled:**

Test Purpose: The response to a successful *GetCoverage* request containing no `Scal::Scaling` parameter **shall** consist of a coverage with no scaling applied to the original coverage.

Test method: Send a *GetCoverage* request to the service under test without a `Scal::Scaling` parameter, send a further *GetCoverage* request with a uniform scaling of 1.0; check that both responses are identical.

Test passes if result is as expected.

A.1.17 getCoverage keep type

Test id: **getCoverage-keep-type:**

Test Purpose: The response to a successful *GetCoverage* request containing a `Scal::Scaling` parameter **shall** consist of a coverage whose type is not changed by the scaling operation.

Test method: Send a *GetCoverage* request to the service under test, containing a `Scal::Scaling` parameter, check that the coverage type is the same as reported by *DescribeCoverage*.

Test passes if result is as expected.

A.1.18 getCoverage exception

Test id: **getCoverage-exception:**

Test Purpose: When a WCS server encounters an error while evaluating a **`scaleFactor`** or **`scaleExtent`** parameter in a *GetCoverage* operation it **shall** return an exception report message chosen as indicated in Table 7 with a `locator` parameter value as specified in the right column of Table 7 for each `exceptionCode` listed.

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.

Test passes if result is as expected.

A.1.19 getCoverage getkvp scale by factor

Test id: **getCoverage-getkvp-scale-by-factor:**

Test Purpose: In a *GetCoverage* request using the GET/KVP protocol, a `Scal::ScaleByFactor` parameter **shall** be represented as

```
SCALEFACTOR=s
```

where `s` is a `scaleFactor` represented as the ASCII representation of a positive floating-point number.

Test method: Send a GET/KVP *GetCoverage* request containing a `Scal::ScaleByFactor` parameter represented as `SCALEFACTOR=s`, where `s` is a `scaleFactor` represented as the ASCII representation of a positive floating-point number. Check proper response.

Test passes if result is as expected.

A.1.20 getCoverage getkvp scale axes by factor

Test id: **getCoverage-getkvp-scale-axes-by-factor:**

Test Purpose: In a *GetCoverage* request using the GET/KVP protocol, a `Scal::ScaleAxesByFactor` parameter containing n>0 axis scaling specifications **shall** be represented as

SCALEAXES=a_1(s_1),…,a_n(s_n)

where, for 1≤i≤n,

- a_i is an axis abbreviation;

- s_i is a `scaleFactor` expressed as the ASCII representation of a positive floating-point number.

Test method: Send a GET/KVP *GetCoverage* containing a `Scal::ScaleAxesByFactor` parameter represented as SCALEAXES=a_1(s_1),…,a_n(s_n), where, for 1≤i≤n, a_i is an `axis` identifier (i.e., URL); s_i is a `scaleFactor` expressed as the ASCII representation of a positive floating-point number scale axes by factor. Check proper response.

Test passes if result is as expected.

A.1.21 getCoverage getkvp scale to size

Test id: **getCoverage-getkvp-scale-to-size**

Test Purpose: **getCoverage-getkvp-scale-to-size:**
In a GetCoverage request using the GET/KVP protocol, a `Scal::ScaleToSize` parameter containing n>0 axis extent specifications **shall** be represented as

`SCALESIZE=`*`a1`*`(`*`s1`*`),…,`*`an`*`(`*`sn`*`)`
where, for $1 \leq i \leq n$,
- *`ai`* is an `axis` abbreviation;
- *`si`* are `sizes`.

Test method: Send a GET/KVP *GetCoverage* containing a `Scal::ScaleToSize` parameter represented as `SCALESIZE=`$a_1(s_1),…,a_n(s_n)$ where, for $1 \leq i \leq n$, a_i is an `axis` name; s_i are `sizes`. Check proper response.

Test passes if result is as expected.

A.1.22 getCoverage getkvp scale to extent

Test id: **getCoverage-getkvp-scale-to-extent:**

Test Purpose: In a *GetCoverage* request using the GET/KVP protocol, a `Scal::ScaleToExtent` parameter containing n>0 axis extent specifications **shall** be represented as
`SCALEEXTENT=`$a_1(lo_1:hi_1),…,a_n(lo_n:hi_n)$
where, for $1 \leq i \leq n$,
- a_i is an axis abbreviation;
- lo_i and hi_i are `low` and `high`, respectively, each of them represented as either a string, enclosed in double quotes, or a number.

Test method: Send a GET/KVP *GetCoverage* containing a `Scal::ScaleToExtent` parameter represented as `SCALEEXTENT=`$a_1(lo_1,hi_1),…,a_n(lo_n,hi_n)$, where, for $1 \leq i \leq n$, a_i is an `axis` name; lo_i and hi_i are an admissible lower and upper bound value, resp., for this coverage. Check proper response.

Test passes if result is as expected.

A.1.23 getCoverage xmlpost

Test id: **getCoverage-xmlpost:**

Test Purpose: In a *GetCoverage* request using the XML/POST protocol, the following mapping **shall** be used:
- Alternative 1 **shall** be encoded by a `scal:ScaleByFactor` element;
- Alternative 2 **shall** be encoded by a `scal:ScaleAxesByFactor` element;
- Alternative 3 **shall** be encoded by a `scal:ScaleToSize` element;
- Alternative 4 **shall** be encoded by a `scal:ScaleToExtent` element.

Test method: Send XML/POST *GetCoverage* requests to the service under test containing:

- □ a `scal:ScaleByFactor` element. Verify that request succeeds;
- □ a `scal:ScaleAxesByFactor` element. Verify that request suc-

ceeds;

- a `scal:ScaleToSize` element. Verify that request succeeds;
- a `scal:ScaleToExtent` element. Verify that request succeeds.

Test passes if expected result is delivered.

A.1.24 getCoverage soap

Test id: **getCoverage-soap:**

Test Purpose: In a *GetCoverage* request using the SOAP protocol, scaling parameters **shall** be mapped as in Requirement 23 above.
Dependency: [09-149r1] (http://www.opengis.net/spec/WCS_protocol-binding_soap/1.0/soap)

Test method: Send SOAP *GetCoverage* requests to the service under test containing:

- a `scal:ScaleByFactor` element. Verify that request succeeds;
- a `scal:ScaleAxesByFactor` element. Verify that request succeeds;
- a `scal:ScaleToSize` element. Verify that request succeeds;
- a `scal:ScaleToExtent` element. Verify that request succeeds.

Test passes if expected result is delivered.

-- end of ATS –

Annex B: Revision history

Date	Release	Author	Paragraph modified	Description
2012-04-20	0.0.1	Peter Baumann	All	Created
2012-12-03	0.1.0	Peter Baumann, JinsongdiYu	Several, in particular: Annex A	Finalized and added ATS
2014-01-06	1.0	Peter Baumann, JinsongdiYu	Req7, Req9 Annex A	Editorial revision