Open Geospatial Consortium

Date: 2013-06-21

Reference number of this Document: OGC 12-174

Version: 0.0.1

Category: OpenGIS© Interface Standard

Editor: Peter Baumann, Mircea Alexandru Dumitru

**OGC® Web Coverage Service Interface Standard - REST Protocol Extension**

Copyright © 2013 Open Geospatial Consortium.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

**Warning**

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

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: OGCStandard

Document subtype: Interface

Document stage: Draft

Document language: English

Contents Page

[1 Scope 1](#_Toc359701231)

[2 Conformance 1](#_Toc359701232)

[3 Normative references 1](#_Toc359701233)

[4 Terms and definitions 2](#_Toc359701234)

[4.1 Resource 2](#_Toc359701235)

[4.2 URL component 2](#_Toc359701236)

[4.3 URL component key 2](#_Toc359701237)

[4.4 URL component value 2](#_Toc359701238)

[5 Conventions 2](#_Toc359701239)

[5.1 UML notation 2](#_Toc359701240)

[5.2 Data dictionary tables 2](#_Toc359701241)

[6 *Rest* requirements class 3](#_Toc359701242)

[6.1 Overview 3](#_Toc359701243)

[*6.2* General 3](#_Toc359701244)

[*6.3* *GetCapabilities* 3](#_Toc359701245)

[*6.4* *DescribeCoverage* 4](#_Toc359701246)

[*6.5* *GetCoverage* 4](#_Toc359701247)

[6.5.1 Request structure 4](#_Toc359701248)

[6.5.2 Output encoding 6](#_Toc359701249)

[7 Exception Codes 7](#_Toc359701250)

[Bibliography 8](#_Toc359701251)

[Annex A (normative) Abstract test suite 9](#_Toc359701252)

**Tables** Page

[Table 1 — Conformance class dependencies 1](#_Toc359701253)

[Table 1—Components of the coverage-description resource request URL 4](#_Toc359701254)

[Table 2 — URL Components of the coverage resource request URL 5](#_Toc359701255)

[Table 3 — Exception codes for rest 7](#_Toc359701256)

Preface

This document specifies an extension to the OGC Web Coverage Service (WCS) 2.0 core toallow for client/server communication using a RESTful interface over HTTP.

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.

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.

Submitting organizations

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

|  |  |
| --- | --- |
| * Jacobs University Bremen
 | * rasdamanGmbH
 |

Document Contributor Contact Points

|  |  |
| --- | --- |
| **Name** | **Organization**  |
| Peter Baumann | Jacobs University Bremen | rasdaman GmbH |
| Mircea Alexandru Dumitru | Jacobs University Bremen | Flanche Creative Labs SRL |

Revision history

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Date | Release | Author | Paragraph modified | Description |
| 2012-12-20 | 0.0.1 | MirceaAlexandruDumitru, Peter Baumann | All | Created |
| 2013-07-01 | 0.0.2 | Peter Baumann | All | Merged KVP into REST |

Changes to the OpenGIS® Abstract Specification

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

 Future Work

* Establish REST protocol conformance class in WCS service extension specifications where this is necessary, based on this REST definition for the WCS Core.

Foreword

This WCS REST Protocol extension is an OGC Interface Standard which relies on WCS Core [OGC 09-110r4].

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) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space/time-varying phenomena.

This document specifies an Extension to the OGC Web Coverage Service (WCS) 2.0 Core to allow for client/server communication using a RESTful interface [1] over HTTP.

OGC® Web Coverage Service Interface Standard - REST Protocol Extension

# Scope

This document specifies how Web Coverage Service (WCS) clients and servers can commu- nicate over the Internet using a RESTful interface over HTTP.

# Conformance

This document establishes a single requirements class *rest*, of URI [http://www.opengis.net/spec/WCS\_protocol-binding\_rest/1.0rest](http://www.opengis.net/spec/WCS_protocol-binding_get-rest/1.0/req/get-rest) with a single pertaining conformance class, *rest*, of URI [http://www.opengis.net/spec/WCS\_protocol-binding\_rest/1.0/conf/rest](http://www.opengis.net/spec/WCS_protocol-binding_get-rest/1.0/conf/get-rest). Requirements and conformance test URIs listed in this document are relative to [http://www.opengis.net/spec/WCS\_protocol-binding\_rest/1.0/](http://www.opengis.net/spec/WCS_protocol-binding_get-rest/1.0/)

Annex A lists the conformance tests which shall be exercised on any software artefact claim- ing to implement an OGC WCS using this extension.

# Normative references

This OGC WCS 2.0 REST Protocol Binding 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\_protocol-binding\_rest/1.0](http://www.opengis.net/spec/WCS_protocol-binding_get-rest/1.0), the document has OGC URI <http://www.opengis.net/doc/IS/WCS_protocol-binding_rest/1.0>.

The complete specification is available for download from [http://www.opengeospatial.org/standards/wcs](http://www.opengeospatial.org/stan%1Fdards/wcs); additionally, the XML Schema is posted online at <http://schemas.opengis.net/wcs/2.0> 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 — Conformance class dependencies

|  |  |  |
| --- | --- | --- |
| **REST conformance class** | **Dependency document** | **Dependency conformance class** |
| *rest* | OGC 09-110rX, *OGC®**Web Coverage Service 2.0 Interface Standard - Core*, version 2.0 | *core* |

# 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.

Further, the following definitions from RFC 3986 apply:

## Resource

abstract Web primitive

Example A coverage resource may be served RESTfully via URL
http://acme.com/MERIS\_Global\_Mosaic\_2009

## URL component

text contained after the *context path* between two forward slashes “/”, that are not placed between two matching parentheses “(...)”, and before any question mark character “?”

Example In the URL below, “subset(lat(10:12))” and “subset(lon(-5:5))” are the two URL components:
http://acme.com/MERIS\_Global\_Mosaic\_2009/
 subset(lat(10:12))/subset(lon(-5:5))

## URL component key

text contained in an URL component that comes before an opening parenthesis “(“

Example In the URL below, “subset” denotes the component key:
http://acme.com/MERIS\_Global\_Mosaic\_2009/
 subset(lat(10:12))/subset(lon(-5:5))

## URL component value

text contained in an URL component that is enclosed between the first opening parenthesis “(“ and the last closing parenthesis “)” in the →component

Example In the URL below, “lat(10:12)” and “lon(-5:5)” denote the two component values, respectively:
http://acme.com/MERIS\_Global\_Mosaic\_2009/
 subset(lat(10:12))/subset(lon(-5:5))

# Conventions

## 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].

## 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.

# *Rest* requirements class

## Overview

This Clause 6 establishes the WCS REST Protocol Binding Extension core conformance class, *rest*.This conformance class specifies how a client can request and obtain resources, such as coverages, from a server implementing this protocol.

## General

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

1. A WCS service implementing the conformance class *rest* of this Extension **shall** include the following URI in the Profile element of the ServiceIdentification in a *GetCapabilities* response:
http://www.opengis.net/spec/WCS\_protocol-binding\_rest/1.0/conf/rest
2. Operation responses **shall** URL-encode special characters as defined in [2].

Example Use “%3F” to represent a question mark, “?”.

1. URL Components **shall** be case sensitive.

Note While this requirement is strict, the corresponding conformance test tentatively does not stamp an implementation non-conformant if it is elastic in also recognizing a wrong case in values, as long as this does not cause a conflict.

## *GetCapabilities*

A *GetCapabilities* request in the *rest* conformance class consists of the *service-path* to which a new URL component is added, “capabilities”, while the response – a capabilities document – is an XML document. The *resource* represented in this URL is the *service* which exposes its capabilities through the document response.

Example To request a capabilities document a client can request the service resource by issuing an HTTP GET request to an URL of this form:
http://acme.com/capabilities

1. The response to a *capabilities* resource request **shall** be a valid XML document of type wcs:CapabilitiesType.

Example See [OGC 09-110r3].

## *DescribeCoverage*

A *DescribeCoverage* operation in the rest conformance class consists of the service-path and the URL Components described in Table 2.The resource represented by this URL is the coverage’s *description*.

1. The *coverage description* URL Components in a request URL **shall** adhere to the syntax defined in **Table 1**.

Table 2—Components of the coverage-description resource request URL

|  |  |  |  |
| --- | --- | --- | --- |
| **URL Component Key** | **Definition** | **Data type** | **Multiplicity** |
| coverage | Identifier for the coverage set of resources | StringFixed to “coverage” | One(mandatory) |
| coverageId | The id of the coverage | NCName | One(mandatory) |
| Description | Identifier for the description resource | StringFixed to “description” | One(mandatory) |

Example To request the description resource of a coverage resource a client can request it by issuing an HTTP GET request to an URL of this form:
http://acme.com/coverage/C0001/description

1. The response to a successful *coverage-description* resource request **shall** be a valid XML document of type wcs:DescribeCoverageType.

Example See [OGC 09-110r3].

## *GetCoverage*

### 6.5.1 Request structure

A *GetCoverage* request in the *rest* conformance class consists of the *service-path* and the URL Components described in Table 3.The *resource* represented in the URL is the actual coverage or, depending on the URL Components, a subset of it.

1. A *coverage* resource request **shall** contain the URL components defined in Table 2­and **.**

Table — URL Components of the coverage resource request URL

|  |  |  |  |
| --- | --- | --- | --- |
| **URL Component Key** | **Definition** | **Data type** | **Multiplicity** |
| coverage | Identifier for the coverage set of resources | StringFixed to “coverage” | One(mandatory) |
| coverageId | The identifier of the coverage on this server | NCName | One(mandatory) |
| subset | Boundaries of the coverage subset | SubsetSpec as defined in Requirement 8 | Zero or more(optional) |

Access by coverageId defines access to a complete coverage. Subsetting (i.e., trimming and slicing) is done by concatenating subsetting specifiers to the path; this follows the philosophy that a coverage contains a set of direct positions (“cells”) which can be accessed through URL paths.

Syntax rules are as follows: underlined tokens represent literals which appear as is (“terminal symbols”), other tokens represent sub-expressions to be substituted (“non- terminals”). A vertical bar (“|”) denotes alternatives, items in brackets (“[ ]”) are optional. Non-terminals NCName, number, and token follow the respective XML definitions.

1. Each SubsetSpec **shall** adhere to this EBNF syntax:
 SubsetSpec: subset = axisName ( intervalOrPoint )
 axisName: NCName
 intervalOrPoint: interval | point
 interval: low : high
 low: point | \*
 high: point | \*
 point: number | "token" //" = ASCII code 0x42

Note Allowed values for points are determined by the CRS used. This ranges from 2009-11-06 for time over -41.5 for lat/long to "41°5’ for lat/long whereby non-numeric values have to be enclosed in double quotes.

Example The following KVP-encoded *GetCoverage* request addresses service path on server www.acme.com retrieves all range fields of coverage C0001 in the domain specified by the bounding box with longitude (-71,47) and latitude (-66,51), expressed in some spatio-temporal CRS being the Native CRS of this coverage:
http://acme.com/coverage/C0001/
 subset=lon(-71:47)/
 subset=lat(-66:51)/
 subset=t("2009-11-06T23:20:52Z")

1. Any path component allowed by some WCS extension **shall** be acceptable also as a query component and vice versa, without change in syntax for this component (except for possible entity encoding).

Example The following two *GetCoverage* requests are equivalent:
http://acme.com/coverage/C0001?rangesubset=red,green,blue
http://acme.com/coverage/C0001/rangesubset=red,green,blue

1. Any path component except coverage (cf. ) **shall** be allowed to occur in either path position or in the query part following GET/KVP notation, but not in both simultaneously.

Note 1 A path component consists of the complete string between separating “/” characters. For example, a path component may contain subsetting in one axis while a query component can do subsetting on another axis, such as:
http://acme.com/coverage/C0001/
 subset=lon(-71:47)/
 subset=lat(-66:51)?
 subset=t("2009-11-06T23:20:52Z")

Note 2 This merges GET/KVP with REST, effectively allowing coverageId and subset as per as well as any other path component defined by some WCS extension to appear in both path and query position.

Example In this mixed REST / KVP notation allowed by this standard, the following requests are equivalent:
http://www.acme.com/coverage/C0001
http://www.acme.com/coverage?coverageid=C0001

1. A *coverage* resource request **shall** result ineither a coverage response obtained by evaluating all URL Components in the exact sequence as provided in the request path (evaluated left to right), or in an UnsupportedOperationSequence exception.

Note 1 This requirement is particularly relevant in presence of URL components defined in WCS Extensions, such as range subsetting, reprojection, etc.

Note 2 No sequencing rule is defined here for query elements in a request. WCS extensions may establish sequencing and compatibility rules among these query components, or may leave it open to nondeterministic choice.

### 6.5.2 Output encoding

Requesting an output format encoding is done RESTfully via the Accept Header as defined in IETF RFC 4229 [5]. If none is indicated, the coverage will be returned in its Native Format (see below).

1. A *coverage* resource request **may** contain an HTTP Accept Header [5] with a value consisting of one or more MIME type identifiers, optionally using wildcards, with a syntax and semantics as specified in RFC2616 [4] Clause 14 .

Example 1 The following request header indicates that a representation of coverage resource C0001 in the JPEG2000 format is to be delivered by the server:
Accept: image/jp2

Example 2 The following request header indicates that one of JPEG2000 and PNG is acceptable, and the server can pick one response format at its discretion:
Accept: image/jp2, image/png

Example 3 The following request header indicates that the client prefers to receive TIFF, but any image format is acceptable if it is the best available after an 80% mark-down in quality:
Accept: image/\*; q=0.2, image/tiff

Note The Accept Header corresponds to the format parameter in the Get/KVP request encoding.

1. If a *coverage* resource request contains an HTTP Accept Header then its value **shall** be one of the supported formats listed in the Capabilities document, and shall be suitable to encode the complete or subcoverage representation requested.

Example An x/y slice taken from an x/y/t image timeseries cube can be encoded in image/tiff; an x/y/t subcube cannot be encoded in image/tiff, but in application/x-netcdf.

1. If a *coverage* resource request contains an HTTP Accept Header with MIME type identifier M then the coverage response **shall** be encoded in the format identified by M, otherwise the coverage response **shall** be encoded in its Native Format as stored in the server

Note This is coherent with RFC2616 [4]: “If no Accept header field is present, then it is assumed that the client accepts all media types.”

1. If a *coverage* resource request contains an HTTP Accept Header where a media type X is suffixed with a parameter:
 mediaType=multipart/related
then the coverage response, when encoded in X, **shall** be consist of a GMLCOV multipart/related message with the range set encoded in X.
**Dependency:** WCS Core [OGC 09-110] and GMLCOV [OGC 09-146r2]

Example The following request header indicates that the client prefers to receive JPEG2000, with a GML metadata header, both combined into a multipart/related message as defined in GMLCOV:
Accept: image/jp2; mediaType=multipart/related

Note This indicates that the expected server response consists of a multipart/related message where the first part contains coverage information encoded in GML, but the range set replaced by an xlink:href reference to a further message part containing the coverage range set. At the time of this writing, WCS defines only one such range set part in the multipart message, but future versions may allow more than one such part.

# Exception Codes

1. When a WCS server encounters an error described in column “meaning of exception code” in Table 3 then it **shall** return the corresponding exception report message with the contents of the locator parameter value as specified in the right column of Table 3.

Note WCS Core and extensions specify further exception situations and codes, and additionally a server may add its own exceptions.

Table 4 — Exception codes for rest

|  |  |  |  |
| --- | --- | --- | --- |
| **exceptionCode value** | **HTTP code** | **Meaning of exception code** | **locator value** |
| InvalidEncodingSyntax | 400 | Document received does not conform with protocol syntax | First offending URL component |
| UnsupportedOperationSequence | 400 | Sequence of operations as specified in the URL component sequence is not supported by service addressed | First offending URL component |

# Bibliography

1. Fielding, Roy Thomas, Architectural Styles and the Design of Network-based Software Architectures, Doctoral dissertation, University of California, Irvine
2. OGC 08-094r1, OGC® SWE Common Data Model Encoding Standard, version 2.0
3. IETF RFC 3986, Uniform Resource Identifiers (URI): Generic Syntax. IETF, 2005
4. IETF RFC 2616, Hypertext Transfer Protocol -- HTTP/1.1. IETF, 1999
5. IETF RFC 4229, HTTP Header Field Registrations. IETF, 2005

# Annex A(normative)Abstract test suite

A REST Protocol Binding Extension implementation must satisfy the following system characteristics, plus those of the WCS core [OGC 09-110r3], to be conformant with this specification.

Test identifiers below are relative to [http://www.opengis.net/spec/WCS/2.0/WCS\_protocol-binding\_rest/1.0/conf](http://www.opengis.net/spec/WCS/2.0/WCS_protocol-binding_rest-interface/1.0/conf).

* 1. Conformance Test Class: *rest*

The OGC URI identifier of this conformance class is:
<http://www.opengis.net/spec/WCS/2.0/WCS_protocol-binding_rest/1.0/conf/rest/>.

Record-subsetting extension identifier

|  |  |
| --- | --- |
| Test id: Test Purpose: | **extension-identifier**A WCS service implementing the conformance class *rest* of this Extension **shall** include the following URI in the Profile element of the ServiceIdentification in a *GetCapabilities* response:http://www.opengis.net/spec/WCS\_protocol-binding\_rest/1.0/conf/rest |
| Test method: | Send a *GetCapabilities* request to server under test, verify that the response contains a Profile element with said URI.Test passes if all individual tests pass. |

Encode special characters

|  |  |
| --- | --- |
| Test id:Test Purpose: | **url-encoding**Operation responses **shall** URL-encode special characters as defined in [2]. |
| Test method: | For each request type, send a request to the service under test which con- tains special characters and send a request such that the response contains special characters. Check correct handling of the special characters.Test passes if all individual tests pass. |

Proper case handling

|  |  |
| --- | --- |
| Test id: | **case-sensitivity** |
| Test Purpose: | URL Components **shall** be case sensitive. |
| Test method: | For each request type: * Send requests to the server under test containing lower, mixed, and upper case keys. Check proper response.
* Send requests to the server under test with different case in values. Check thatthe server differentiates in its response.

Test passes if all individual tests pass. |

Capabilities Response Structure

|  |  |
| --- | --- |
| Test id: | **service-path** |
| Test Purpose: | **Error! Reference source not found.** |
| Test method: | Send a valid resource request without specifying the service path URL Components. Pass test if the resource requested is not returned.Test passes if all individual tests pass. |

Capabilities response structure

|  |  |
| --- | --- |
| Test id: | **capabilities-response-structure** |
| Test Purpose: | The response to a *capabilities* resource request **shall** be a valid XML document of type wcs:CapabilitiesType. |
| Test method: | Send a valid capabilities resource request. Pass test if an XML validator reports validity of the response document against its schema definition.Test passes if all individual tests pass. |

Coverage description request syntax

|  |  |
| --- | --- |
| Test id: | **coverage-description-request-syntax** |
| Test Purpose: | The *coverage description* URL Components in a request URL **shall** adhere to the syntax defined in **Table 1** |
| Test method: | Request a coverage description resource, testing server response on the cases distinguished in said reference. Check proper response.Test passes if all individual tests pass. |

Coverage description response structure

|  |  |
| --- | --- |
| Test id: | **capabilities-response-structure** |
| Test Purpose: | The response to a successful *coverage-description* resource request **shall** be a valid XML document of type wcs:DescribeCoverageType. |
| Test method: | Send a valid coverage description resource request. Pass test if an XML validator reports validity of the response document against its schema definition.Test passes if all individual tests pass. |

Coverage request syntax

|  |  |
| --- | --- |
| Test id: | **coverage-request-syntax** |
| Test Purpose: | A *coverage* resource request **shall** contain the URL components defined in Table 2­andRequirement10**.** |
| Test method: | Request a coverage resource, testing server response on the cases distinguished in said reference. Check proper response.Test passes if all individual tests pass. |

Coverage request format

|  |  |
| --- | --- |
| Test id: | **coverage-request-format** |
| Test Purpose: | 1. A *coverage* resource request **may** contain an HTTP Accept Header [5] with a value consisting of one or more MIME type identifiers, optionally using wildcards, with a syntax and semantics as specified in RFC2616 [4] Clause 14 .

Example 1 The following request header indicates that a representation of coverage resource C0001 in the JPEG2000 format is to be delivered by the server:Accept: image/jp2Example 2 The following request header indicates that one of JPEG2000 and PNG is acceptable, and the server can pick one response format at its discretion:Accept: image/jp2, image/pngExample 3 The following request header indicates that the client prefers to receive TIFF, but any image format is acceptable if it is the best available after an 80% mark-down in quality:Accept: image/\*; q=0.2, image/tiffNote The Accept Header corresponds to the format parameter in the Get/KVP request encoding.If a *coverage* resource request contains an HTTP Accept Header then its value **shall** be one of the supported formats listed in the Capabilities document |
| Test method: | Request a coverage resource in a format present in the capabilities document. Check if the response coverage is in the selected format.Test passes if all individual tests pass. |

Coverage request subset component specification

|  |  |
| --- | --- |
| Test id: | **subset-spec-syntax** |
| Test Purpose: | Syntax rules are as follows: underlined tokens represent literals which appear as is (“terminal symbols”), other tokens represent sub-expressions to be substituted (“non- terminals”). A vertical bar (“|”) denotes alternatives, items in brackets (“[ ]”) are optional. Non-terminals NCName, number, and token follow the respective XML definitions.Each SubsetSpec **shall** adhere to this EBNF syntax: SubsetSpec: subset ( axisName ( intervalOrPoint) ) axisName: NCName intervalOrPoint: interval | point interval: low : high low: point | \* high: point | \* point: number | "token" //" = ASCII code 0x42 |
| Test method: | Request coverage resources from the service under test, evaluate whether responses are adequate (based on knowledge about a pre-existing coverages with at least 3 dimensions). Exercise tests for the following situations: * No subsetting parameter
* Zero/one/two trimmings, no slicing
* Zero/one/two slicings, no trimming
* trim operations with trim coordinates and with “\*” for low and high
* bound (independently)

Each test in the above set shall be performed * Once for each dimensionality supported by the server
* Without CRS parameter provided in the request, and with a valid CRS parameter provided (if the server supports a CRS extension).

Pass if coverage responses indicate (by range set inspection) that the operation has been recognized and executed properly. |

Exceptions

|  |  |
| --- | --- |
| Test id: | **exceptions** |
| Test Purpose: | When a WCS server encounters an error described in column “meaning of exception code” in Table 3 then it **shall** return the corresponding exception report message with the contents of the locator parameter value as specified in the right column of Table 3. |
| Test method: | Request resources of all types supported to the server under test. Each request shall include all (mandatory and) optional parameters and shall be valid except for one parameter which shall contain an encoding error described in the exception specification. Test passes if exception is reported according to requirement.Test passes if all individual tests pass. |

-- end of ATS –