

Open Geospatial Consortium Inc.

Date: 2009-01-15

Reference number of this document: OGC 07-068r4

Version: 1.1.4

Category: OpenGIS[®] Implementation Standard

Editor: Arliss Whiteside

Web Coverage Service (WCS) — Transaction operation extension

Copyright © 2009 Open Geospatial Consortium, Inc. All Rights Reserved.
To obtain additional rights of use, visit <http://www.opengeospatial.org/legal/>.

Document type:	OpenGIS [®] Implementation Standard
Document subtype:	Extension
Document stage:	Approved version 1.1.4
Document language:	English

Contents	Page
i. Preface.....	v
iii. Document contributor contact points.....	vi
iv. Revision history	vi
v. Changes to the OGC Abstract Specification.....	vi
vi. Future work.....	vi
Foreword.....	vii
Introduction.....	viii
1 Scope.....	1
2 Compliance	1
3 Normative references	2
4 Terms and definitions	2
5 Conventions	3
5.1 Abbreviated terms	3
5.2 UML notation	3
5.3 Used parts of other documents	3
5.4 Platform-neutral and platform-specific specifications	3
6 WCS Transaction extension implementation model.....	3
6.1 Introduction	3
6.2 Transaction operation	5
6.3 Transaction operation request	5
6.3.1 Transaction request contents.....	5
6.3.2 Transaction operation actions	8
6.3.3 Input coverages metadata.....	9
6.3.4 Use of OtherSource parameters	11
6.3.5 Implementation requirements	12
6.4 Transaction operation response	12
6.4.1 Transaction execution modes.....	12
6.4.2 Acknowledgement message.....	13
6.4.3 Transaction response contents	13
6.4.4 Implementation requirements	14
6.5 Complete UML model.....	14
7 Transaction data encoding using XML and SOAP.....	16
7.1 Introduction	16
7.2 Transaction request encoding.....	16
7.3 Acknowledgement message encoding.....	17
7.4 Transaction response XML encoding.....	17

7.5	Transaction operation exceptions.....	18
7.6	Transaction operation metadata.....	19
7.7	Transaction extension identifier URNs	20
7.8	XML schema document	21
Annex A (normative) Abstract test suite		22
Annex B (normative) WCS Transaction operation conformance classes.....		23
B.1	Introduction	23
B.2	Transaction extension core conformance class	23
B.3	Transaction extension other conformance classes.....	25
B.3.1	Transaction “action” extension conformance classes	25
B.3.2	Transaction input extension conformance classes	25
B.3.3	Other Transaction extension conformance classes	25
B.4	Transaction extensions of base WCS conformance classes	25
B.4.1	Introduction.....	25
B.4.2	Coverage range conformance classes	25
B.4.3	Coverage domain conformance classes	26
B.4.4	Other WCS conformance classes.....	26

Figures	Page
Figure 1 — WCS interface UML diagram	4
Figure 2 — Transaction operation request UML class diagram.....	6
Figure 3 — Acknowledgement UML diagram.....	13
Figure 4 — TransactionResponse UML class diagram	14
Figure 5 — WCS Transaction package class diagram.....	15

Tables	Page
Table 1 — Parameters in Transaction operation request	7
Table 2 — Coverage data structure	8
Table 3 — Transaction operation “actions” and input data.....	9
Table 4 — URNs for “role” values.....	11
Table 5 — Contents of Acknowledgement data structure	13
Table 6 — Parts of TransactionResponse data structure	14
Table 7 — Exception codes for Transaction operation	19
Table 8 — Constraints on Transaction operation	19

i. Preface

This Standard is an extension of the Web Coverage Service (WCS) 1.1 corrigendum 2 (version 1.1.2) Implementation Standard [OGC 07-067r5]. With small changes, this extension is expected to also apply to WCS 1.2.0. Unlike previous versions, WCS 1.1 and 1.2 are divided into a base standard plus multiple extensions (formerly called application profiles). This extension is based on change request [OGC-06-043r4].

EDITOR'S NOTE 1 This document specifies asynchronous execution of the Transaction operation using the form of asynchronous execution specified for the GetCoverage operation in WCS 1.1.2. That form is the same as now specified for CSW, in Clause 10 of Catalogue 2.0.2 [OGC 07-006r1]. The OGC is now considering alternatives to this form of asynchronous execution. However, OGC selection and specification of a better form of asynchronous execution is not imminent. When that better form has been selected and specified, it can be considered for WCS 1.2.

EDITOR'S NOTE 2 This document specifies the approach to providing one or more InputCoverages to a Transaction operation that is the same as the WCS 1.1 approach for providing one or more output coverages from the GetCoverage operation.

EDITOR'S NOTE 3 This document specifies only implementation of XML encoding with SOAP packaging of Transaction operation requests and responses, although WCS 1.1.2 also specifies XML encoding without SOAP packaging. The WCS 1.2 SWG is now considering deprecating XML encoding without SOAP packaging.

This Standard separates the complete UML model (here called the Implementation Model), from specifications of the XML and other encoding needed in a complete Implementation Standard. This Implementation Model might alternatively be moved into a separate document. This Implementation Model is included as the platform-independent specification required by Subclause 10.3 of OGC Abstract Specification Topic 12: OpenGIS Service Architecture (which contains a copy of ISO 19119). The majority of the implementation requirements are stated in this Implementation Model, with the remaining implementation requirements being stated in the specifications of the XML and other encoding.

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

ii. Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 06-121r3], 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.

iii. Document contributor contact points

All questions regarding this document should be directed to the editor or the contributors:

Name	Organization
Arliss Whiteside	BAE Systems E&IS

iv. Revision history

Date	Release	Editor	Primary clauses modified	Description
2007-06-27	1.1.1 07-068	Arliss Whiteside	All	First draft, majority copied from OGC 06-043r4
2007-11-12	1.1.2 07-068r1	Arliss Whiteside	i, Foreword, 3, 6, 7.2.2, 7.3.5, 7.3.6, 7.4	General improvements
2008-05-12	1.1.3 07-068r2	Arliss Whiteside	i, Foreword, 6, 7, B, C, D	Re-structure Transaction requests and responses, general improvements
2008-05-14	1.1.4 07-068r3	Arliss Whiteside	6.3.1, 6.3.2, 6.3.3, 6.4.1, 6.4.2, 6.5, 7.7	Miscellaneous small corrections and improvements

v. Changes to the OGC Abstract Specification

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

vi. Future work

Improvements in this document are desirable to:

- a) Support transactions involving multiple servers
- b) Include WSDL definition of SOAP encoding of Transaction operation request and response
- c) Include more example XML documents

Foreword

This document is an extension of the Web Coverage Service (WCS) 1.1 corrigendum 2 (version 1.1.2) Implementation Standard [OGC 07-067r5]. With small changes, this extension is expected to also apply to WCS version 1.2. Unlike previous versions, WCS 1.1 and 1.2 are divided into a basic standard plus multiple extensions. This extension is based on change request [OGC-06-043r4], and supersedes that Discussion Paper. This document does not supersede any other previously approved OGC document

This document includes two annexes; annexes A and B are both normative.

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 Inc. 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 Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space-varying phenomena. A WCS provides client access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi-valued coverages, and input into scientific models and other clients. The WCS is currently limited to quadrilateral grid coverages, providing information at the grid points, usually with interpolation between these grid points.

This extension of the WCS standard specifies an additional Transaction operation that may optionally be implemented by WCS servers. This Transaction operation allows clients to add, modify, and delete grid coverages that are available from a WCS server. The Transaction operation request references or includes the new or modified coverage data, including all needed coverage metadata.

Web Coverage Service (WCS) — Transaction operation extension

1 Scope

This extension of the WCS standard specifies an additional Transaction operation that may optionally be implemented by WCS servers. This Transaction operation allows clients to add, modify, and delete grid coverages that are available from a WCS server. The Transaction operation request references the new or modified coverage data, including all needed coverage description metadata.

The Web Coverage Service (WCS) supports electronic retrieval of geospatial data as "coverages" – that is, digital geospatial information representing space-varying phenomena. A WCS provides client access to potentially detailed and rich sets of geospatial information, in forms that are useful for client-side rendering, multi-valued coverages, and input into scientific models and other clients. The WCS is currently limited to quadrilateral grid coverages, providing information at the grid points, usually with interpolation between these grid points.

This document is an extension of the Web Coverage Service (WCS) 1.1 Corrigendum 2 (version 1.1.2) Implementation Standard [OGC 07-067r5]. With small changes, this extension is expected to also apply to WCS 1.2. Unlike previous versions, WCS 1.1 and 1.2 are divided into a base standard plus multiple extensions.

2 Compliance

Compliance with this extension Standard shall be checked for WCS servers claiming to implement this Standard, using all the relevant tests specified in Annex A (normative). This Standard defines six conformance classes:

- a) Transaction extension core conformance class: Tested product correctly implements the requirements that shall always be implemented when this Transaction extension is implemented, and excludes testing all the optional requirements included in the further extensions defined below. Briefly, the Transaction extension core conformance class includes testing of the Transaction operation request and response for adding one coverage (using the Add action). These conformance tests are specified in the Abstract Test Suite in Clause A.2.
- b) Transaction multiple action extension conformance class: Tested product correctly implements the additional requirements to allow performing more than one action in response to one Transaction operation request. These conformance tests are specified in the Abstract Test Suite in Clause A.3
- c) Transaction UpdateMetadata extension conformance class: Tested product correctly implements the additional requirements for the UpdateMetadata action, which

replaces all the metadata for one or more offered coverages. These conformance tests are specified in the Abstract Test Suite in Clause A.4.

- d) Transaction Delete extension conformance class: Tested product correctly implements the additional requirements for the Delete action, which deletes the data and metadata for one or more offered coverages. These conformance tests are specified in the Abstract Test Suite in Clause A.5.
- e) Transaction UpdateAll extension conformance class: Tested product correctly implements the additional requirements for the UpdateAll action, which replaces the data and metadata one or more offered coverages. These conformance tests are specified in the Abstract Test Suite in Clause A.6.
- f) Transaction UpdateDataPart extension conformance class: Tested product correctly implements the additional requirements for the UpdateDataPart action, which replaces part of the range (or pixel) data for one or more offered coverages. These conformance tests are specified in the Abstract Test Suite in Clause A.7.

The last four extensions conformance classes listed above are independently implementable, in any combination. However, these four extensions conformance classes shall all require implementation of the second class: Transaction multiple action extension conformance class.

3 Normative references

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

OGC 07-067r5, *OpenGIS[®] Web Coverage Service Implementation Standard, version 1.1.2*

NOTE This WCS Standard contains a list of normative references that are also applicable to this extension Standard.

OGC 07-092r1, *Definition identifier URNs in OGC namespace, version 1.1.2*

OGC 06-121r3, *OpenGIS[®] Web Services Common Specification, version 1.1.0*

NOTE This OWS Common Specification contains a list of normative references that are also applicable to this Implementation Standard.

In addition to this document, this part of this standard includes the normative XML Schema Document file referenced in Subclause 7.8.

4 Terms and definitions

For this extension standard, the definitions specified in Clause 4 of the base WCS 1.1.2 Implementation Standard [OGC 07-067r5] shall apply.

5 Conventions

5.1 Abbreviated terms

Most of the abbreviated terms listed in Subclause 5.1 of the base WCS 1.1.2 Implementation Standard [OGC 07-067r5] apply to this extension standard.

5.2 UML notation

The diagrams that appear in this part of this standard are presented using the Unified Modeling Language (UML) static structure diagram, as described in Subclause 5.2 of [OGC 06-121r3].

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 06-121r3]. The contents of these data dictionary tables are normative, including any table footnotes.

5.3 Used parts of other documents

This document uses parts of document [OGC 06-121r3]. To reduce the need to refer to that document, this document copies some of those parts with small modifications. To indicate those parts to readers of this document, the largely copied parts are shown with a light grey background (15%).

5.4 Platform-neutral and platform-specific specifications

As specified in Subclause 5.4 of OWS Common [OGC 06-121r3], this document includes both platform-neutral and platform-specific specifications.

EXAMPLES 1 Platform-neutral specifications are contained in Clause 6.

EXAMPLES 2 Platform-specific specifications for XML encoding are contained in Clause 7.

6 WCS Transaction extension implementation model

6.1 Introduction

This extension Standard specifies an additional Transaction operation that may be requested by a WCS client and performed by a WCS server. This extension Standard shall be optional implementation by servers. When implemented, the specified Transaction operation provides an open standard, web accessible way to add, modify, and/or delete coverages in the set of grid coverages offered by a WCS server

The Transaction operation allows a client to request that a WCS server input new coverage(s) into that server, update existing coverages, and/or delete existing coverages. This operation references the coverages to be inserted, updated, or deleted in that WCS server, and it is the job of the server to resolve the reference(s), fetch that data, and store this data for future access.

Figure 1 is a simple UML diagram summarizing the extended WCS interface. This class diagram shows that the WebCoverageServer interface class inherits the getCapabilities operation from the OGCWebService interface class, and adds the getCoverage and

describeCoverage operations. This diagram also shows the new WebCoverageServiceTransactional interface class, added by this extension, inherits the getCapabilities, getCoverage, and describeCoverage operations from the WebCoverageServer interface class, and adds the “transaction” operation. (This capitalization of names uses the OGC/ISO profile of UML.) A more complete UML model of the Transaction operation is provided in Subclause 6.5.

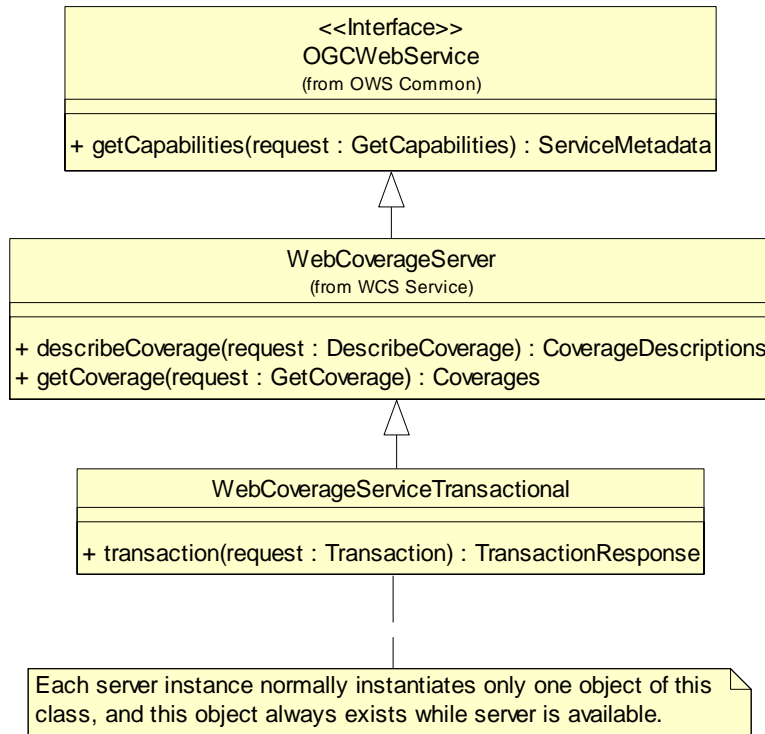


Figure 1 — WCS interface UML diagram

The encoding of Transaction operation requests shall use HTTP POST with XML encoding, as specified in Clause 11 of [OGC 06-121r3], together with SOAP packaging, as specified in Annex F of [OGC 07-067r5].

NOTE No KVP encoding of the WCS Transaction operation request is now specified, since KVP encoding appears impractical without significantly restricting Transaction requests.

This Clause 6 provides a UML model of this Transaction operation extension, which specifies the interface in a platform-neutral manner as is required by Subclause 10.3 of OGC Abstract Specification Topic 12: OpenGIS Service Architecture (which contains a copy of ISO 19119). The one platform-specific specification of this extension is subsequently provided in Clause 7, which specifies the XML and SOAP encoding used by this specific platform. The majority of the implementation requirements are stated in this Implementation Model, with the remaining implementation requirements being stated in the specifications of the XML and other encoding.

6.2 Transaction operation

The Transaction operation allows WCS clients to request that a WCS server import one or more new coverages into that server, update existing coverages, and/or delete existing coverages. The operation request references (externally and attached) the data to be added or updated in that WCS server. It is the job of the server to resolve these references, fetch that data, and store this data for future access. This operation is required implementation by WCS servers that implement this extension Standard.

NOTE This Transaction operation is similar to the Transaction operation in the CSW protocol binding, specified in Subclause 10.11 of the Catalogue Services Specification version 2.0.2 [OGC 07-006r1]. This WCS operation makes use of the Manifest data structure that is specified in Subclause 13.2 of [OGC 06-121r3], to support adding or modifying multiple coverages, and to include all needed metadata associated with each coverage.

The Transaction operation has two modes of operation, controlled by a parameter in the operation request. The first mode of operation is synchronous, in which the server receives a Transaction request from the client, executes it immediately, and sends the results to the client while the client waits. The second mode of operation is asynchronous, in which the server receives a Transaction request from the client, and sends the client an immediate acknowledgement that the request has been successfully received. The server can then execute the Transaction operation when it has time, taking as much time as required, and then send the execution results to a URI specified in the original Transaction request. This latter mode of operation is included to support Transaction operation execution that could run longer than most HTTP timeouts allow.

6.3 Transaction operation request

6.3.1 Transaction request contents

A request to perform the Transaction operation shall include the parameters and data structure shown graphically in the UML diagram in Figure 3, and specified in Table 1 through Table 4.

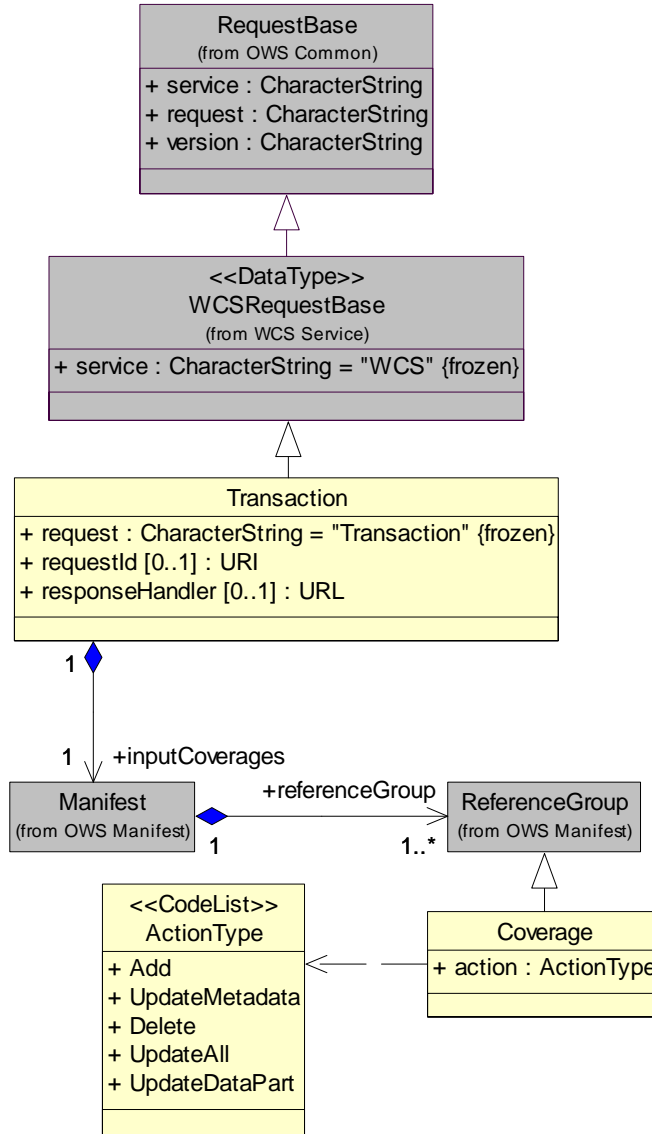


Figure 2 — Transaction operation request UML class diagram

NOTE 1 To reduce the need for readers to refer to other documents, the first three parameters listed below are largely copied from Table 26 in Subclause 9.2.1 of [OGC 06-121r3].

Table 1 — Parameters in Transaction operation request

Names ^a	Definition	Data type and value	Multiplicity and use
service service	Service type identifier	Character String type, not empty Value is OWS type abbreviation, namely “WCS”	One (mandatory)
request request	Operation name	Character String type, not empty Value is operation name, namely “Transaction”	One (mandatory)
version version	Specification version for operation	Character String type, not empty Value is specified by each Specification version	One (mandatory)
input Coverages Input Coverages	List of one or more coverages to be added, modified, or deleted	Manifest data structure, see Subclause 13.3 of OGC 06-121r3 Contains one or more Coverage data structures, see Table 2 and Subclauses 6.3.2 and 6.3.3	One (mandatory)
requestId RequestId	Unique identifier of this operation request	URI	Zero or one (optional) Include if assigned by client
response Handler Response Handler	Address to respond to when server has completed processing this operation request	URL	Zero or one (optional) Include to process operation request asynchronously ^b
<p>a Although some values listed in the “Names” column appear to contain spaces, they shall not contain spaces. See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.</p> <p>b When this parameter is omitted in the operation request, this operation shall be executed synchronously and immediately by the server, with the operation response returned to the client following operation execution. When this parameter is included, this operation shall be executed asynchronously by the server. In that case, the server shall return an operation acknowledgement to the client immediately following operation acceptance. When operation execution is later completed, the (normal or exception) operation response shall be sent by the server to the response handler address provided by this parameter value.</p>			

The ReferenceGroup data structure used by the Manifest data structure (see Subclause 13.3 of OGC 06-121r3) is extended by the new Coverage data structure as indicated in Figure 3, and specified in Table 2. This Coverage data structure shall be used in the InputCoverages data structure (instead of the ReferenceGroup data structure).

NOTE 2 To reduce the need for readers to refer to other documents, the first six parameters listed below are largely copied from the ReferenceGroup data structure specified in Table 46 of [OGC 06-121r3].

Table 2 — Coverage data structure

Names	Definition	Data type	Multiplicity and use
identifier Identifier	An unambiguous identifier of this Coverage, normally used by software	ows:CodeType, an adaptation of MD_Identifier class in ISO 19115 ^a	Zero or one (optional) Include when needed
title ^c Title	Title of this Coverage, normally used for display to a human	LanguageString data structure, see Figure 15 in OGC 06-121r3	Zero or more (optional) Include when available and useful ^b Include one for each language represented
abstract ^c Abstract	Brief narrative description of this Coverage, normally available for display to a human	LanguageString data structure, see Figure 15 in OGC 06-121r3	Zero or more (optional) Include when available and useful Include one for each language represented
keywords ^c Keywords	Unordered list of one or more commonly used or formalised word(s) or phrase(s) used to describe this Coverage	MD_Keywords class in ISO 19115	Zero or more (optional) One for each keyword authority used
metadata Metadata	Additional metadata about this Coverage ^c	reference to metadata or metadata contents, see gml:metaDataProperty ^d	Zero or more (optional) One for each useful metadata object
reference Reference	Reference to one document or resource	Reference data structure, see Table 48 in OGC 06-121r3	One or more (mandatory) One for each reference in this group
action Action	Identifier if action to be performed on this coverage	CodeList data structure See Table 4	One (mandatory)

a The optional codeSpace attribute in the ows:CodeType is expected to rarely be used in the short term. Wherever a specific OWS specification expects this codeSpace attribute to be used, that specification shall specify how it should be used there, including its values and meanings.

b Software may display the “Identifier” value when the “Title” is absent.

c This Metadata should be used primarily by specific servers, not by specific OWSs. The specification editors for each specific OWS should decide what additional metadata to require or encourage be included. That specific OWS should then specify additional parameters and groups as needed to contain this additional metadata, with specific names and meanings.

d Use or adaptation of the gml:metaDataProperty data type is specified in order to include references to the type of metadata that is pointed to (or included), and to what aspect of the data(set) this metadata applies to.

e The multilingual scoping rules in Subclause 10.7.3 of [OGC 06-121r3] shall apply.

6.3.2 Transaction operation actions

The InputCoverages data structure in a Transaction operation request shall include one Coverage data structure for each offered coverage to be added, modified, or deleted by this operation. Each Coverage in an InputCoverages specifies an “action” (or “sub-

operation") of the Transaction operation. All actions included in one Transaction request shall either succeed or fail as a group.

In each Coverage data structure, the Action parameter shall specify whether this Coverage is being added, modified, or deleted, as specified in Table 4. As needed, the Identifier in each ReferenceGroup data structure shall identify the existing coverage to be modified or deleted, as specified in the third column of Table 4. The InputCoverages information for each action shall include the coverage (pixel) data and/or metadata as specified in the right column of Table 4.

Table 3 — Transaction operation “actions” and input data

Action value	Transaction action	Identifier value	Coverage includes
Add	Add this coverage to set of coverages being offered	Client-suggested identifier for new coverage, if any ^a	All coverage data and metadata
UpdateMetadata	Replace all the metadata for this coverage currently being offered	Identifier of coverage whose metadata is to be replaced	All coverage metadata
Delete	Delete all the data and metadata for this coverage currently being offered	Identifier of coverage to be deleted	(none)
UpdateAll	Replace all the data and metadata for this coverage currently being offered	Identifier of coverage to be completely updated	All coverage data and metadata
UpdateDataPart	Replace range (or pixel) values for part of the domain data of this coverage currently being offered	Identifier of coverage whose data is to be partially updated	Coverage range data for part of domain being updated ^b
<p>a If a client does not suggest a coverage Identifier, it may provide an empty value. If a client suggests an identifier value that is already in use by the server, or does not follow the format required by that server, the server shall change that Identifier as may be appropriate.</p> <p>b The input data for each UpdateDataPart action shall be a rectangular coverage in the Domain GridCRS or ImageCRS, plus any temporal domain, of this offered coverage. This (currently simple) input range data shall cover exactly the rectangular part of the offered coverage’s domain to be replaced, and shall have the same coverage range structure (not values) as this offered coverage.</p>			

NOTE 1 The coverage Identifiers are not normally human readable, and are separate from the normally human readable “title” values. Multiple “title” values can be used, possibly providing a human-readable title in different languages.

NOTE 2 A separate action is not specified for updating all the data, since the metadata should also be updated in this case. The UpdateAll action should thus be used in this case. Alternately, the UpdateDataPart action may be used, providing new range values for the entire coverage domain.

Each input coverage data and/or its metadata shall be referenced by one Coverage data structure within the InputCoverages data structure. The coverage pixel data will often be referenced remotely, but can be attached to the Transaction request message. Similarly, the metadata can be attached to the Transaction request message or referenced remotely.

6.3.3 Input coverages metadata

The coverage metadata referenced in the right column of Table 4 may include several separate metadata files. One of these metadata files shall be the CoverageDescription

specified in Subclause 9.3.1 of [OGC 07-067r5]. This CoverageDescription shall contain all the description information needed for that coverage which is not specific to a WCS server.

If a WCS server that implements this Transaction operation implements CoverageSummary elements in the contents section of a Capabilities document, one or more of these metadata files shall be a CoverageSummary as specified in Subclause 8.3.4.2 of [OGC 07-067r5]. Each included CoverageSummary shall contain all the summary information needed for that coverage which is not specific to a WCS server.

NOTE 1 The number of CoverageSummary objects provided will depend on whether and how the CoverageSummary object hierarchy is being used for or by each coverage.

NOTE 2 The CoverageDescription and CoverageSummary files are required to be included in the metadata provided in order to not require the server to extract that information from other coverage metadata that may be provided. In addition, the other coverage metadata may not include all the information required.

In addition, each new or modified coverage (or image) that is unrectified but georeferenced should include metadata file(s) containing the georeferencing coordinate Transformation(s). At least one such transformation should be included when the server is expected to be able to georectify an unrectified coverage, since the server will use these transformation(s) to perform georectification when requested by a client. All this coverage metadata shall be encoded in XML as specified in [OGC 07-067r5] or in GML 3.1.1.

The CoverageDescription and CoverageSummary metadata omitted as being specific to a specific server should be only the SupportedCRS and SupportedFormat values other than the CRS and format used by the offered coverage being added or updated.

NOTE 3 In general, the client of the Transaction operation will not know all the SupportedCRS and SupportedFormat values that a WCS server chooses to support.

In all cases, the Identifier shall be included in the CoverageDescription, and when needed in a CoverageSummary. For a coverage that is being added, the client-suggested Identifier shall be changed by the server if it is not in the server-supported format or conflicts with an existing Identifier.

A set of CoverageSummaries may use inheritance of summary parameters as desired, using inheritance as specified in Subclause 8.3.3.4 of [OGC 07-067r5]. In each CoverageSummary, any subsidiary CoverageSummaries shall be correctly referenced. The client shall provide a consistent set of CoverageSummaries; the server shall not have to adjust the consistency of the input set of CoverageSummaries.

In each CoverageDescription, the InterpolationMethods object shall be included, to indicate the desired default and other interpolation methods. If an interpolation method is listed that is not supported by this server (for this coverage), the server shall detect an error and return an InvalidParameterValue error message.

NOTE 4 In general, suitable interpolation methods are logically associated with each coverage, or with each field in a coverage range. The Transaction operation client is thus allowed to specify the desired default and other interpolation methods for each field. However, a server implementation is not required to implement all (frequently-used) interpolation methods.

Each file referenced by one Coverage shall include a "role" parameter value that indicates the type of data referenced. These "role" values shall use URNs in the "ogc" URN namespace as specified in Table 5. The right column in Table 5 indicates the allowed multiplicity and use of "role" parameters in separate Reference elements in one Coverage element, for the Add, UpdateMetadata, and UpdateAll actions. Notice that the URNs in this table use the "objectType" value of "role" that is not (yet) specified in Table 2 of [OGC 07-092r1].

Table 4 — URNs for "role" values

URN for "role"parameter	Meaning	Multiplicity and use ^a
urn:ogc:def:role:WCS:1.1:Pixels	File containing coverage range values	One or more (mandatory) Include for each separate file ^b
urn:ogc:def:role:WCS:1.1:CoverageDescription	File containing CoverageDescription object, as specified in Subclause 9.3 of [OGC 07-067r5]	One (mandatory)
urn:ogc:def:role:WCS:1.1:CoverageSummary	File containing CoverageSummary object, as specified in Subclause 8.3.3 of [OGC 07-067r5]	Zero or more (optional) Include for each needed CoverageSummary
urn:ogc:def:role:WCS:1.1:GeoreferencingTransformation	File containing unrectified image georeferencing coordinate Transformation	Zero or more (optional) Include for each needed and available Transformation ^c
urn:ogc:def:role:WCS:1.1:OtherSource	Reference to catalogue server from which coverage metadata is available, as specified in Subclause 8.3.3.5 of [OGC 07-067r5]	Zero or more (optional) Include for each needed OtherSource
<p>^a This Multiplicity and use column applies to each coverage for the Add, UpdateMetadata, and UpdateAll actions (not for the Delete and UpdateDataPart actions). None of these file types and role parameters are used for a Delete action. Only the Pixels and CoverageDescription file types and role parameters are used for an UpdateDataPart action, with the CoverageDescription describing the coverage part being updated.</p> <p>^b Include for each separate file containing pixel (coverage range) value data, as required or allowed by a specified coverage encoding extension. (Multiple files are possible, but are expected to be rare.)</p> <p>^c More than one unrectified image georeferencing coordinate Transformation may be included, when useful as may be specified by a future extension. At least one such transformation shall be included when the server is expected to be able to georectify an unrectified coverage, and the server shall use these transformation(s) to perform georectification when requested by a client. Any image georeferencing coordinate Transformation optionally included in CoverageSummary shall not be used by the server to perform georectification.</p>		

NOTE 5 The above listed URNs for "role" values should be re-used when applicable by specified coverage encoding extensions. (WCS 1.1 does not specify any URNs for "role" values, but only includes some examples, which are superseded by this extension.)

6.3.4 Use of OtherSource parameters

The preceding discussion largely assumes that all coverages have at least one CoverageSummary, instead of using only OtherSource reference(s) to catalogue(s) of coverages. The use(s) of OtherSource references should be specified in a WCS and a CSW extension. When coverage catalogue(s) are being used, there are at least two alternative methods for a catalogue to get new and modified coverage metadata.

One method is for the client(s) that execute WCS Transaction operations to also execute the corresponding Catalogue Transaction operations. In this case, CoverageSummaries may be omitted in WCS Transaction requests. The metadata sent to the WCS for each new or modified coverage shall also include OtherSource reference(s) to the appropriate catalogue(s). The server shall then add each of these OtherSource(s) to the existing list, if it is not already in that list.

Another method is for coverage catalogue(s) to obtain new and modified coverage metadata from WCS servers. Again, CoverageSummaries may be omitted in WCS Transaction requests. In this case, the CoverageDescriptions shall include all the coverage metadata that might be needed by a Catalogue. Just how a Catalogue knows when and which coverages have been added, updated, or deleted is not specified here (but may be specified in an extension).

6.3.5 Implementation requirements

The “Multiplicity and use” column in Table 1 specifies the optionality of each listed parameter and data structure in the Transaction operation request. All the “mandatory” parameters and data structures shall be implemented by all WCS clients that use this operation, using a specified value(s). Similarly, all the “mandatory” parameters and data structures shall be implemented by all WCS servers that implement this operation, checking that each request parameter or data structure is received with an allowed value(s).

The “optional” RequestId and ResponseHandler parameters, in the Transaction operation request, shall be implemented by all WCS servers that implement this Transaction operation. The responseHandler parameter shall be implemented by all WCS clients that can execute this operation asynchronously. The RequestId parameter shall be implemented by all WCS clients that use this parameter.

The Add Transaction action listed in Table 4 shall be implemented by all servers that implement this Transaction operation extension. The other four Transaction actions listed in Table 4 shall each be optional implementation by servers that implement this Transaction operation extension, as indicated by the conformance classes defined in Clause 2.

6.4 Transaction operation response

6.4.1 Transaction execution modes

The Transaction operation shall be executed in one of two modes, depending on the presence or absence of the ResponseHandler parameter in the operation request. If the ResponseHandler parameter is not included, then the WCS server shall execute the Transaction operation immediately (synchronously) and then respond to the waiting client with a TransactionResponse message. When a WCS server encounters an error while immediately performing a Transaction operation, it shall return an exception report message.

If the ResponseHandler parameter is included in the Transaction operation request, then the WCS server should verify the request syntax and immediately respond to the client

with an Acknowledgement message. When a WCS server encounters an error while immediately checking a Transaction operation request, it shall immediately return an exception report message.

Later, after the server has executed the Transaction operation, it shall generate a TransactionResponse message and send it to the URL specified by the ResponseHandler parameter, using the protocol encoded therein. Common protocols are *ftp* for sending the response to a ftp server and *mailto* which may be used to send the response to an email address. When a WCS server encounters an error after sending an Acknowledgement message, it shall return an exception report message to the ResponseHandler URL.

6.4.2 Acknowledgement message

When the Transaction operation is performed asynchronously, the immediate response to a valid operation request shall be an Acknowledgement message that uses the data structure shown in the UML diagram in Figure 5. The attributes and associations of the new class shall include the parameters and data structures listed and defined in Table 7.

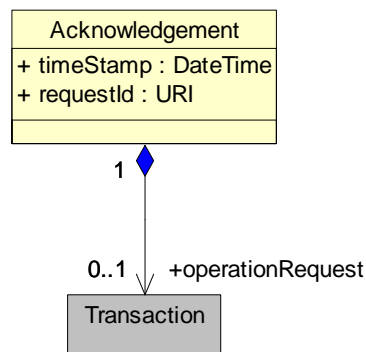


Figure 3 — Acknowledgement UML diagram

Table 5 — Contents of Acknowledgement data structure

Names	Definition	Data type and values	Multiplicity and use
timeStamp TimeStamp	Data and time this Acknowledgement message generated	DateTime type	One (mandatory)
requestId RequestId	Unique identifier of this operation request	URI Value from operation request or assigned by server	One (mandatory)
operationRequest OperationRequest	Copy of Transaction operation request being acknowledged	Transaction request data structure, see Figure 3 and Table 1	Zero or one (optional) Include when expected to be useful to clients

6.4.3 Transaction response contents

The normal response to a valid Transaction operation request shall list each coverage modified (successfully). More precisely, the response from a Transaction operation shall use the TransactionResponse data structure shown in the UML diagram in Figure 7, and detailed in Table 9.

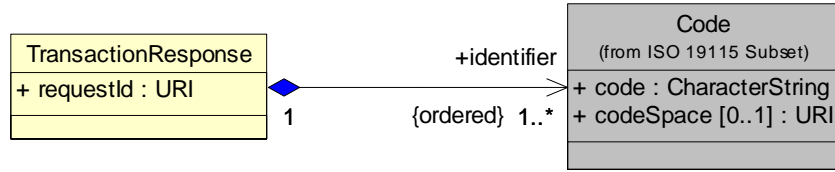


Figure 4 — TransactionResponse UML class diagram

Table 6 — Parts of TransactionResponse data structure

Names ^a	Definition	Data type	Multiplicity
requestId RequestId	Unique identifier of this operation request	URI Value from operation request or assigned by server	One (mandatory)
identifier Identifier	Unique identifier or name of a coverage added, modified, or deleted	ows:CodeType, as adaptation of MD_Identifier class in ISO 19115	One or more (mandatory) Include one for each coverage added, modified, or deleted ^b
<p>a See Table 1 of [OGC 06-121r3] for UML and XML naming conventions.</p> <p>b These Identifiers shall be listed in the same order as the corresponding Coverage data structures in the Transaction request.</p>			

6.4.4 Implementation requirements

The “Multiplicity and use” column in Table 9 specifies the optionality of each listed parameter and data structure in the TransactionResponse. The “mandatory” parameters and data structure shall be implemented by all WCS servers that implement this operation. Similarly, the “mandatory” parameters and data structures shall be implemented by all WCS clients that use this operation, using a specified value(s).

6.5 Complete UML model

The complete WCS Transaction package is shown in the class diagram in Figure 9. This diagram also shows several used classes from other packages. The six classes introduced by this package are further defined by Table 1 through Table 9 in this document.

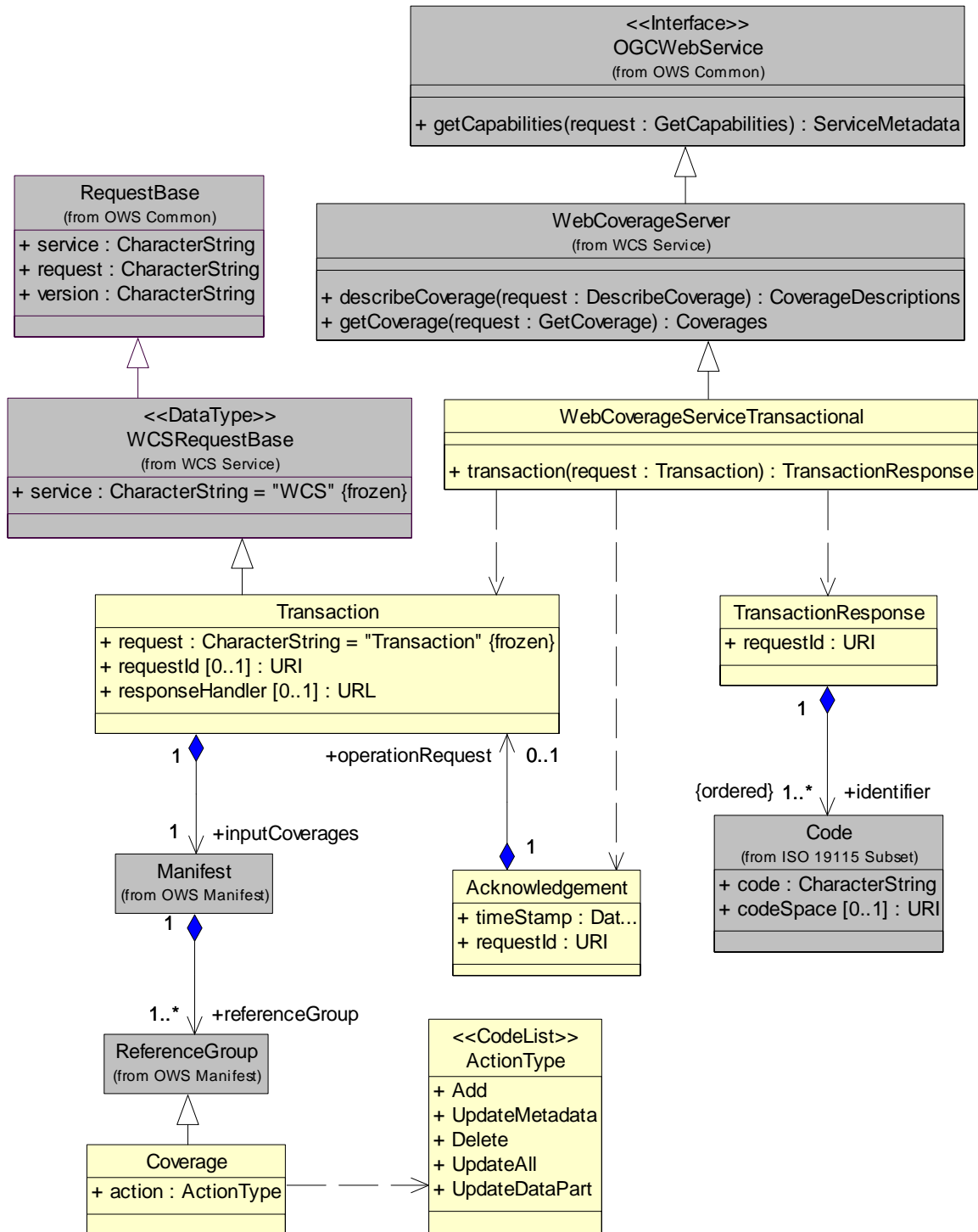


Figure 5 — WCS Transaction package class diagram

7 Transaction data encoding using XML and SOAP

7.1 Introduction

This clause specifies how to encode the Transaction operation request and response, specified in Clause 6, in the currently specified platform-specific manner, using XML encoding with SOAP transfer of Transaction operation requests and responses.

WCS servers that implement the Transaction operation shall implement SOAP version 1.2 transfer of the Transaction operation request and all responses as specified in Annex F of [OGC 07-067r5], using the XML encoding specified below.

7.2 Transaction request encoding

WCS servers that implement the Transaction operation shall implement XML encoding of the Transaction operation request. The XML Schema fragment for encoding the Transaction operation request shall be as specified by the Transaction element in the attached `wcstTransaction.xsd` file.

EXAMPLE 1 An example Transaction operation request for adding one coverage is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transaction xmlns="http://www.opengis.net/wcs/1.1/wcst"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/1.1/wcst ../wcstTransaction.xsd"
  service="WCS" version="1.1">
  <InputCoverages>
    <Coverage>
      <ows:Reference xlink:href="http://foo.bar.com/Images/abcd1234Pixels"
xlink:role="urn:ogc:def:role:WCS:1.1:Pixels"/>
      <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd1234Metadata2"
xlink:role="urn:ogc:def:role:WCS:1.1:CoverageDescription"/>
      <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd1234Metadata3"
xlink:role="urn:ogc:def:role:WCS:1.1:GeoreferencingTransformation"/>
      <Action>Add</Action>
    </Coverage>
  </InputCoverages>
</Transaction>
```

EXAMPLE 2 An example Transaction operation request for different “actions” on four coverages is:

```
<?xml version="1.0" encoding="UTF-8"?>
<Transaction xmlns="http://www.opengis.net/wcs/1.1/wcst"
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/1.1/wcst
../wcstTransaction.xsd"
  service="WCS" version="1.1">
  <InputCoverages>
    <Coverage>
      <ows:Identifier
codeSpace="TBD">CoverageIdentifierTBD1</ows:Identifier>
      <ows:Reference xlink:href="http://foo.bar.com/Images/abcd1234Pixels"
xlink:role="urn:ogc:def:role:WCS:1.1:Pixels"/>
```



```

        <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd1234Metadata2"
xlink:role="urn:ogc:def:role:WCS:1.1:CoverageDescription"/>
        <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd1234Metadata3"
xlink:role="urn:ogc:def:role:WCS:1.1:GeoreferencingTransformation"/>
        <Action>Add</Action>
    </Coverage>
</Coverage>
    <ows:Identifier
codeSpace="TBD">CoverageIdentifierTBD2</ows:Identifier>
    <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd3456Metadata2"
xlink:role="urn:ogc:def:role:WCS:1.1:CoverageDescription"/>
    <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd3456Metadata3"
xlink:role="urn:ogc:def:role:WCS:1.1:GeoreferencingTransformation"/>
    <Action>UpdateMetadata</Action>
</Coverage>
</Coverage>
    <ows:Identifier
codeSpace="TBD">CoverageIdentifierTBD3</ows:Identifier>
    <ows:Reference xlink:href="http://foo.bar.com/Images/abcd1234Pixels"
xlink:role="urn:ogc:def:role:WCS:1.1:Pixels"/>
    <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd5678Metadata2"
xlink:role="urn:ogc:def:role:WCS:1.1:CoverageDescription"/>
    <ows:Reference
xlink:href="http://foo.bar.com/Images/abcd5678Metadata3"
xlink:role="urn:ogc:def:role:WCS:1.1:GeoreferencingTransformation"/>
    <Action>UpdateAll</Action>
</Coverage>
</Coverage>
    <ows:Identifier
codeSpace="TBD">CoverageIdentifierTBD4</ows:Identifier>
    <ows:Reference xlink:href=""/>
    <Action>Delete</Action>
</Coverage>
</InputCoverages>
</Transaction>

```

7.3 Acknowledgement message encoding

WCS servers that implement the Transaction operation shall implement XML encoding of the Transaction operation Acknowledgement response. The XML Schema fragment for the Acknowledgement message shall be as specified by the Acknowledgement element in the attached wcsTransaction.xsd file.

7.4 Transaction response XML encoding

WCS servers that implement the Transaction operation shall implement XML encoding of the Transaction operation response. The XML Schema fragment for encoding the (complete) Transaction operation response shall be as specified by the TransactionResponse element in the attached wcstTransaction.xsd file.

EXAMPLE 1 An example Transaction operation response from adding one coverage is:

```

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

```

```

    xsi:schemaLocation="http://www.opengis.net/wcs/1.1/wcst
../wcstTransaction.xsd">
    <RequestId>RequestTBD</RequestId>
    <ows:Identifier>CoverageTBD</ows:Identifier>
</TransactionResponse>

```

EXAMPLE 2 An example Transaction operation response from different “actions” on five coverages is:

```

<?xml version="1.0" encoding="UTF-8"?>
<TransactionResponse xmlns="http://www.opengis.net/wcs/1.1/wcst "
  xmlns:ows="http://www.opengis.net/ows/1.1"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wcs/1.1/wcst
../wcstTransaction.xsd">
  <RequestId>RequestIdentifierTBD</RequestId>
  <ows:Identifier>CoverageTBD1</ows:Identifier>
  <ows:Identifier>CoverageTBD2</ows:Identifier>
  <ows:Identifier>CoverageTBD3</ows:Identifier>
  <ows:Identifier>CoverageTBD4</ows:Identifier>
  <ows:Identifier>CoverageTBD5</ows:Identifier>
</TransactionResponse>

```

7.5 Transaction operation exceptions

When a WCS server encounters an error while performing a Transaction operation, it shall return an exception report message as specified in Subclause 7.4 of [OGC 06-121r3]. The allowed standard exception codes shall include at least those listed in Table 11. For each listed exceptionCode, the contents of the “locator” parameter value shall be as specified in the right column of Table 11.

NOTE To reduce the need for readers to refer to other documents, the first four values listed below are copied from Table 25 in Subclause 8.3 of [OGC 06-121r3].

Table 7 — Exception codes for Transaction operation

exceptionCode value	Meaning of code	“locator” value
OperationNotSupported	Request is for an operation that is not supported by this server	Name of operation not supported
MissingParameterValue	Operation request does not include a parameter value, and this server did not declare a default value for that parameter	Name of missing parameter
InvalidParameterValue	Operation request contains an invalid parameter value	Name of parameter with invalid value
NoApplicableCode	No other exceptionCode specified by this service and server applies to this exception	None, omit “locator” parameter
InvalidURI	One or more of the URI values of an xlink:href parameter was not a valid URL from which data could be accessed by this server ^a	First invalid xlink:href value
ActionFailed	An action specified in the Transaction request could not be completed successfully ^b	Requested action and coverage Identifier
<p>a For this Transaction operation, these xlink:href parameters are Reference XML elements within the InputCoverages element in the Transaction operation request.</p> <p>b When the Transaction operation request specifies adding, updating, and/or deleting multiple coverages, operation execution should be terminated when the first specified action cannot be performed. In this case, the exceptionCode value should be for one of the specific errors detected, if any. If no specific error is detected, the ActionFailed code should be used.</p>		

7.6 Transaction operation metadata

When this Transaction operation is implemented by a WCS server, the metadata for that operation shall also include the three Constraints specified in Table 13, in the Operation data structure in the OperationsMetadata section of the Capabilities (service metadata) document.

Table 8 — Constraints on Transaction operation

Constraint name	Constraint meaning	Data type	AllowedValues
InputFormat	Identifier of accepted format of coverage data inputs to this server	MIME type, see Subclause 10.5 of [OGC 06-121r3]	Unordered list of all format identifiers accepted
InputCRS	Reference to accepted Coordinate Reference System (CRS) for coverage data inputs to this server	URI	Unordered list of all CRSs accepted
Action	Accepted actions (in Title parameter) by Transaction operation of this server	CodeList, see “First Title value” column in Table 4	Unordered list of all actions implemented

EXAMPLE An example of OperationsMetadata for the Transaction operation is

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

```

xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:schemaLocation="http://www.opengis.net/ows/1.1
../../../../ows/1.1.0/owsOperationsMetadata.xsd" name="Transaction">
  <DCP>
    <HTTP>
      <Post xlink:href="TBD"/>
    </HTTP>
  </DCP>
  <Constraint name="InputFormat">
    <AllowedValues>
      <Value>netCDF</Value>
    </AllowedValues>
  </Constraint>
  <Constraint name="InputCRS">
    <AllowedValues>
      <Value>TBD</Value>
      <Value>TBD</Value>
    </AllowedValues>
  </Constraint>
  <Constraint name="Action">
    <AllowedValues>
      <Value>Add</Value>
      <Value>Delete</Value>
    </AllowedValues>
  </Constraint>
</Operation>

```

7.7 Transaction extension identifier URNs

When this WCS Transaction extension is implemented by a WCS server, this extension shall be identified by values of the Profile parameter, in the ServiceIdentification section of the Capabilities XML document. The URNs in the “ogc” namespace used to identify this WCS Transaction extension and each action implemented shall be:

- a) urn:ogc:extension:WCS:1.1:TransactionAdd, for servers that satisfy the Transaction extension core conformance class, which are able to add one coverage (using the Add action)
- b) urn:ogc:extension:WCS:1.1:TransactionMultipleActions, for servers that satisfy the Transaction multiple action extension conformance class, which are able to perform more than one action in response to one Transaction operation request
- c) urn:ogc:extension:WCS:1.1:TransactionUpdateMetadata, for servers that satisfy the Transaction UpdateMetadata extension conformance class, which are able to replace all the metadata for one or more offered coverages
- d) urn:ogc:extension:WCS:1.1:TransactionDelete, for servers that satisfy the Transaction Delete extension conformance class, which are able to delete all the data and metadata for one or more offered coverages
- e) urn:ogc:extension:WCS:1.1:TransactionUpdateAll, for servers that satisfy the Transaction UpdateAll extension conformance class, which are able to replaces all the data and metadata one or more offered coverages

- f) urn:ogc:extension:WCS:1.1:TransactionUpdateDataPart, for servers that satisfy the Transaction UpdateDataPart extension conformance class replace part of the range (or pixel) data for one or more offered coverages

Notice that the URNs listed above use the “objectType” value of “extension” that is not (yet) specified in Table 2 of [OGC 07-092r1].

7.8 XML schema document

The WCS Transaction operation extension Standard specified in this document includes one normative XML Schema Document, which is bundled in the zip file with this document. After OGC acceptance of this standard, this XML Schema Document will also be posted online at the URL <http://schemas.opengespatial.net/wcs/1.1>. In the event of a discrepancy between the bundled and online versions of the XML Schema Documents, the online files shall be considered authoritative. This XML Schema Document roughly matches the one UML package described in Subclause 6.5, and is named:

- a) wcstTransaction.xsd

This XML Schema Document imports and builds upon parts of OWS Common 1.1.0 [OGC 06-121r3] which is also bundled in the zip file with this document.

All these XML Schema Documents contain documentation of the meaning of each element and attribute, and this documentation shall be considered normative as specified in Subclause 11.6.3 of [OGC 06-121r3].

Annex A
(normative)

Abstract test suite

An abstract test suite is not provided in this version of this Implementation Standard, but should be included in version 1.2.0.

Annex B

(normative)

WCS Transaction operation conformance classes

EDITOR'S NOTE This annex will probably be deleted in a future version of this draft document, with some of the information here being included in Annex A.

B.1 Introduction

This annex specifies a set of implementation conformance classes for this WCS Transaction operation extension. These conformance classes include one Transaction core class that shall always be implemented, plus multiple extension conformance classes which may also be implemented. These extension conformance classes are based on the optional abilities specified and implied in this document.

NOTE 1 Most of the possible extension conformance classes listed below are intended to be independently selectable server and client abilities, although interface and implementation details will often overlap. However, at least one of the extensions in some categories must be implemented. Extensions that extend other extensions can be defined, but few of the possible extensions listed below extend other listed extensions.

NOTE 2 One could define extensions that are mutually exclusive extensions, but none of the extensions listed below are (intended to be) mutually exclusive.

B.2 Transaction extension core conformance class

The Transaction extension core conformance class includes the abilities that shall always be implemented when this extension is implemented, and excludes all the optional abilities included in all the further extensions defined below. Briefly, the core conformance class includes implementation and testing of the Transaction operation request and response for adding one coverage, including:

- a) Transaction operation request parameters:
 - 1) service
 - 2) version
 - 3) InputCoverages, referencing one Coverage to be added with all required CoverageDescription and CoverageSummary metadata
 - 4) RequestId, optional
 - 5) ResponseHandler (optional), controlling asynchronous or synchronous execution of the Transaction operation
- b) The Add "action" first Title value, to add this coverage to the set of coverages offered

- c) CoverageDescription (in Transaction operation request) with the parameters:
 - 1) Identifier
 - 2) Title (allowed)
 - 3) Abstract (allowed)
 - 4) Keywords (allowed)
 - 5) Metadata (zero or more)
 - 6) SupportedCRS (one)
 - 7) SupportedFormat (one)
 - 8) Domain, including SpatialDomain but no TemporalDomain
 - 9) Range, with (only) one Field, no Axes, and only “nearest” InterpolationMethod
- d) Acknowledgement message, when Transaction operation executed asynchronously, containing:
 - 1) TimeStamp, giving current date and time
 - 2) RequestId, same as in operation request or generated by server
 - 3) OperationRequest copy without referenced data (optional)
- e) Transaction operation normal response, using the TransactionResponse data structure and including the identifier of the one Coverage that was added.
- f) SOAP packaging of XML encoding (in HTTP POST transfer) of all Transaction operation requests and responses
- g) Constraints on Transaction Operation element in OperationsMetadata section of Capabilities document, including Constraints named:
 - 1) InputFormat, containing list of all format identifiers accepted
 - 2) InputCRS, containing list of all CRSs accepted
 - 3) Action, containing list of all actions implemented
- h) When a WCS server encounters an error while performing a Transaction operation, it shall return an exception report message using the specified format and containing an appropriate exceptionCode value with corresponding “locator” value. The allowed standard exception codes shall include:
 - 1) MissingParameterValue
 - 2) InvalidParameterValue
 - 3) InvalidURI
 - 4) NoApplicableCode

B.3 Transaction extension other conformance classes

B.3.1 Transaction “action” extension conformance classes

Some Transaction extensions support other Transaction “action” options, to handle modifying and/or deleting coverages. These “action” extension conformance classes include (optional) implementation and testing of zero or more of the following actions:

- a) UpdateMetadata: Replace all the metadata for this offered coverage
- b) UpdateDataPart: Replace part of the range (or pixel) data for this offered coverage
- c) UpdateAll: Replace all the data and metadata for this offered coverage
- d) Delete: Delete all the data and metadata for this offered coverage

B.3.2 Transaction input extension conformance classes

Some Transaction extensions support coverage input options. These “input” extension conformance classes include (optional) implementation and testing of ONE or more of EACH of the following:

- a) Each specific data format supported for InputCoverages (multiple profiles)
- b) Each specific CRS supported for InputCoverages (multiple profiles)
- c) Each specific interpolation method supported for InputCoverages (multiple profiles)

B.3.3 Other Transaction extension conformance classes

Some Transaction extensions support other options. These “other” extension conformance classes include (optional) implementation and testing of zero or more of the following:

- a) Support optional inclusion of multiple coverages with same “action” in one Transaction operation request
- b) Support optional inclusion of multiple coverages with different “actions” in one Transaction operation request

B.4 Transaction extensions of base WCS conformance classes

B.4.1 Introduction

Since the Transaction operation uses the CoverageDescription and CoverageSummary XML-encoded data structures, the WCS 1.2 draft conformance classes that apply to these data structures also apply to the instances of these data structures in Transaction operation (extension) requests and responses. Those base WCS conformance classes are summarized below.

B.4.2 Coverage range conformance classes

Some WCS extensions support coverage range options, to handle more complex ranges than one scalar field in a CoverageDescription. These “range” conformance classes include (optional) implementation and testing of zero or more of the following:

- a) Support more than one range field
- b) Support vector range field(s), with one axis in a range field
- c) Support more than one axis in range field(s) (requires above conformance class)
- d) Support RangeSubset in GetCoverage operation request (requires one or more above conformance classes)

B.4.3 Coverage domain conformance classes

Some WCS extensions support coverage domain options, to handle different numbers and types of domain dimensions. These “domain” conformance classes include (optional) implementation and testing of ONE or more of the following:

- a) Support coverages with two spatial domain dimensions that are both horizontal
- b) Support coverages with three spatial domain dimensions
- c) Support coverages with two spatial domain dimensions that are not both horizontal (one is vertical and one is horizontal)
- d) Support coverages with only one spatial domain dimension which is vertical
- e) Support coverages with one only spatial domain dimension which is horizontal
- f) Support coverages with one temporal domain dimension

B.4.4 Other WCS conformance classes

Some WCS extensions support other options. These “other” conformance classes include (optional) implementation and testing of zero or more of the following:

- a) Support optional hierarchical CoverageSummaries, in Contents Section of Capabilities document
- b) Allow specifying desired extent using specific CRS different from (grid base) CRSs of offered and output coverages (multiple modules, using different CRSs)