Open GIS Consortium Inc.

Date: 2003-10-16

Reference number of this OpenGIS® document: OGC 03-088r1

Version: 0.1.0

Category: OpenGIS® Discussion Paper

Editor: Arliss Whiteside

OGC Web Services Common Implementation Specification

Copyright notice

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:	OpenGIS® Discussion Paper
Document subtype:	Normatively referenced by Implementation Specifications
Document stage:	Approved
Document language:	English

File name: 03-088r1.doc

Contents

i. Preface

This document specifies many of the aspects that are, or should be, common to all or multiple OGC Web Service (OWS) interface Implementation Specifications. These common aspects are primarily some of the parameters and data structures used in operation requests and responses. Of course, each such Implementation Specification must specify the additional aspects of that interface, including specifying all additional parameters and data structures needed in all operation requests and responses.

One expected use of this document is as a normative reference from all future versions of OWS interface Implementation Specifications. Rather than continuing to repeat this material in each such Implementation Specification, usually with small changes, each specification should normatively reference each relevant part of this document. Such normative references can take the form of stating "This TBD shall include TBD as specified in Subclause TBD of OGC document TBD." Such normative references are expected to:

a) Reduce the work needed to edit each such Implementation Specification

b) Reduce the length of each such Implementation Specification

c) Increase interoperability among such Implementation Specifications by increasing commonality and discouraging non-essential differences

d) Provide useful guidance to writers of new and revised Implementation Specifications

Each existing OGC-approved and draft OWS interface Implementation Specifications should consider this document to be a formal change request to modify that specification in its next revision to agree with all the relevant material specified herein. Each such specification is also requested to normatively reference each relevant part of this document (instead of repeating the same material).

This is the first version of this document. Most of the current contents were agreed on by an ad-hoc OGC working group on OWS (or W*S) harmonization, initiated at the June 2003 OGC meeting. The current editors of the various OWS interface Implementation Specifications have participated in this OWS harmonization group. For many of the open issues not yet decided by that working group, this document contains an EDITOR'S NOTE that mentions the currently open issue(s). This first version has been approved for public release as an OGC Discussion Paper.

Suggested additions, changes, and comments on this draft specification are welcome and encouraged. Such suggestions may be submitted to the editor by email message. Extensive and/or multiple changes can be suggested by making changes in an edited copy of this document. If you choose to submit suggested changes by editing this document, please make your suggested changes with change tracking on.

ii. Document contributor contact points

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

Contact	Company	Address	Phone	Email
Arliss Whiteside	BAE SYSTEMS Mission Solutions	10920 Technology Dr. San Diego, CA 92127-1608 USA	+1 858-592-1608	Arliss.Whiteside@baesystems.com
Jeff de La Beaujardière	NASA Goddard Space Flight Center	Code 933 Greenbelt MD 20771 USA	+1 301-286-1569	delabeau@iniki.gsfc.nasa.gov
Peter Vretanos	CubeWerx, Inc.		+1 416-701-1985	pvretano@cubewerx.com
John Evans	NASA / GST, Inc.		+1 301-286-0803	jdevans@gst.com
Keith Pomakis	CubeWerx, Inc.	200 rue Montcalm, Gatineau, Québec J8Y 3B5 Canada	+1 819-771-8303 x204	pomakis@cubewerx.com
Jeff Simon	General Dynamics - AIS		+1 703-668-3696	jeff.simon@gd-ais.com

iii. Revision history

Date	Release	Editor	Primary clauses modified	Description
2003-10-06	0.0.0	Arliss Whiteside	All	Initial version
2003-10-16	0.1.0	Arliss Whiteside	7.2.2.3, 7.4.3, 8.1	First approved discussion paper, small additions based on discussions in Architecture WG

iv. Changes to the OpenGIS® Abstract Specification

The OpenGIS© Abstract Specification does not require changes to accommodate the technical contents of this document.

v. Future work

This document should be extended to include other aspects that should be common among multiple OGC Web Service Implementation Specifications, such as:

a) More of the contents of service metadata documents, such as for:
   1) Service metadata section
   2) Capability metadata section
   3) Filter metadata section

b) More parameters used in operation requests and responses, such as for:
   1) Output format
   2) Interpolation method
   3) Coordinate reference system
   4) Units

c) More common operations, such as for:
   1) Transaction
   2) Get(data)
   3) Describe(data)

d) More complete XML Schemas, ready to use

e) Example XML documents based on included XML Schemas

f) UML models of the OWS interface aspects specified herein

g) More guidance for editing OGC Web Service Implementation Specifications

Introduction

This document specifies many of the aspects that are common to all or multiple OGC Web Service (OWS) interface Implementation Specifications. Those specifications currently include the Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). These common aspects include: operation request and response contents; parameters included in operation requests and responses; and encoding of operation requests and responses.

One expected use of this document is as a normative reference from future versions of OWS interface Implementation Specifications. Rather than continuing to repeat this material in each such Implementation Specification, each specification should normatively reference each relevant part of this document.

OGC Web Services Common Implementation Specification

1 Scope

This document specifies many of the aspects that are, or should be, common to all or multiple OGC Web Service (OWS) interface Implementation Specifications. The common Implementation Specification aspects specified by this document currently include:

a) Operation request and response contents, most partial

b) Parameters included in operation requests and responses

c) XML and KVP encoding of operation requests and responses

One use of this document is as a normative reference from future versions of OWS interface Implementation Specifications. Those specifications currently include the Web Map Service (WMS), Web Feature Service (WFS), and Web Coverage Service (WCS). Rather than continuing to repeat this material in each such Implementation Specification, each specification should normatively reference each relevant part of this document.

2 Conformance

Conformance with this specification shall be checked using all the relevant tests specified in each separate specification that normatively references this specification, and specifically references the applicable parts of this specification.

3 Normative references

The following normative documents contain provisions which, 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.

ISO 8601:1988, TBD

ISO 19115:2003, Geographic information — Metadata

ISO 19119:TBD, Geographic information — Services

W3C Recommendation 6 October 2000, Extensible Markup Language (XML) 1.0 (Second Edition), http://www.w3.org/TR/REC-xml

W3C Recommendation 2 May 2001: XML Schema Part 0: Primer, http://www.w3.org/TR/2001/REC-xmlschema-0-20010502/

W3C Recommendation 2 May 2001: XML Schema Part 1: Structures, http://www.w3.org/TR/2001/REC-xmlschema-1-20010502/

W3C Recommendation 2 May 2001: XML Schema Part 2: Datatypes, http://www.w3.org/TR/2001/REC-xmlschema-2-20010502/

TBD

4 Terms and definitions

For the purposes of this document, the following terms and definitions apply.

4.1
client
software component that can invoke an **operation** from a **server**

4.2
interface
named set of operations that characterize the behaviour of an entity [ISO 19119]

4.3
operation
specification of a transformation or query that an object may be called to execute [ISO 19119]

4.4
parameter
variable whose name and value are included in an operation **request** or **response**

4.5
request
invocation of an **operation** by a **client**

4.6
response
result of an **operation,** returned from a **server** to a **client**

4.7
server
a particular instance of a **service**

4.8
service
distinct functionality that is provided by an entity through **interfaces** [ISO 19119 edited]

4.9
service metadata
metadata describing the **operations** and **geographic information** available at a **server**

4.10
version
version of an Implementation Specification (document) to which the requested operation conforms

NOTE An OWS Implementation Specification version may specify XML Schemas against which an XML encoded operation request or response must conform and should be validated.

5 Conventions

5.1 Symbols (and abbreviated terms)

KVP	Key Value Pair
OGC	Open GIS Consortium
OWS	OGC Web Services
TBD	To Be Determined
TBR	To Be Reviewed
UML	Unified Modeling Language
WMS	Web Map Service
XML	Extensible Markup Language

5.2 UML notation

Some of the diagrams in this document are presented using the Unified Modeling Language (UML) static structure diagram. The UML notations used in this document are described in the diagram below.

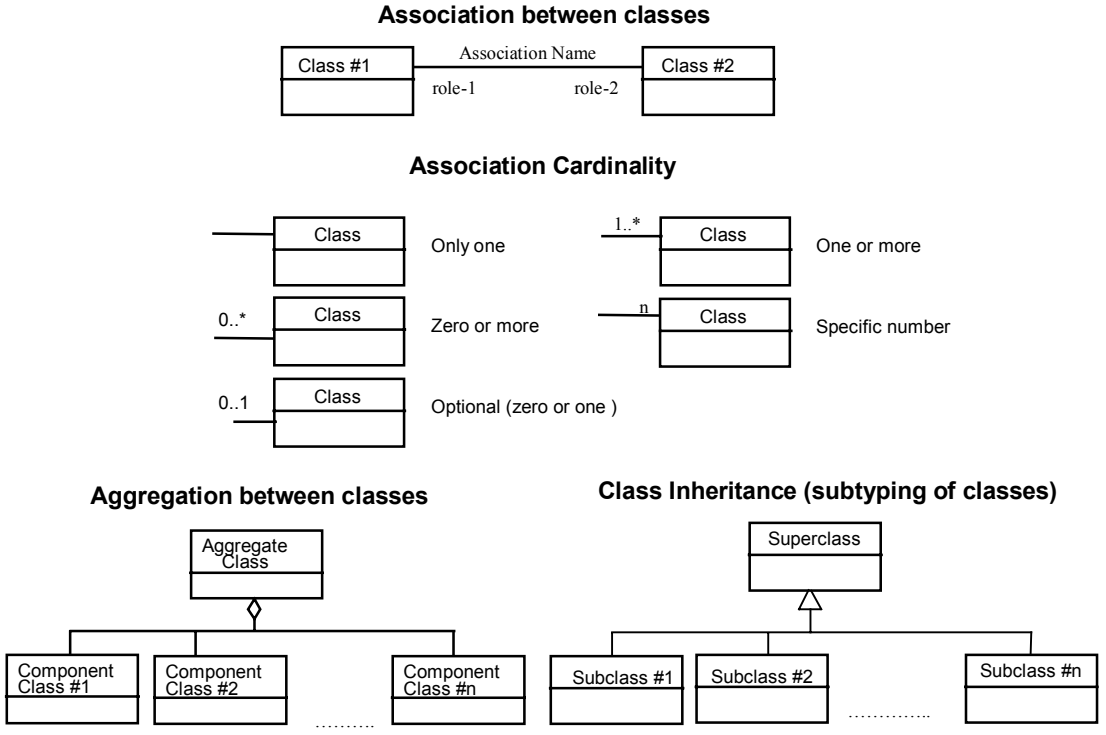

Figure 1 — UML notation

In these UML class diagrams, the class boxes with a light background are the primary classes being shown in this diagram, often the classes from one UML package. The class boxes with a gray background are other classes used by these primary classes, usually classes from other packages.

In this diagram, the following TBD stereotypes of UML classes are used:

a) <<DataType>> A descriptor of a set of values that lack identity (independent existence and the possibility of side effects). A DataType is a class with no operations, whose primary purpose is to hold the information.

b) <<Enumeration>> A data type whose instances form a list of alternative literal values. Enumeration means a short list of well-understood potential values within a class.

c) <<CodeList>> A flexible enumeration for expressing a long list of potential alternative values. If the list alternatives are completely known, an enumeration shall be used; if the only likely alternatives are known, a code list shall be used.

d) <<Interface>> A definition of a set of operations that is supported by objects having this interface. An Interface class cannot contain any attributes.

e) <<Type>> A stereotyped class used for specification of a domain of instances (objects), together with the operations applicable to the objects. A Type class may have attributes and associations.

f) <<Union>> A list of alternate attributes where only one of those attributes can be present at any time.

NOTE All the stereotypes listed above are adapted from Subclause 6.8 of ISO 19103.

In this document, the following standard data types are used:

a) CharacterString – A sequence of characters

b) Boolean – A value specifying TRUE or FALSE

c) URI – An identifier of a resource that provides more information about data

d) URL – An identifier of an on-line resource that can be electronically accessed

e) Integer – An integer number

f) Double – A double precision floating point number

5.3 Document terms and definitions

The following specification terms and definitions are used in this document:

a) shall – verb form used to indicate a requirement to be strictly followed to conform to this specification, from which no deviation is permitted

b) should – verb form used to indicate desirable ability or use, without mentioning or excluding other possibilities

c) may – verb form used to indicate an action permissible within the limits of this specification

d) can – verb form used for statements of possibility

e) informative – a part of a document that is provided for explanation, but is not required

f) normative – a part of a standards document that is required

g) annex – an auxiliary part of a document, called an “appendix” in United States English

h) clause – a major part of a document, called a “section” or “paragraph” in United States English

i) subclause – a secondary part of a clause or annex, called a “subsection” in United States English

6 Document overview

This document is organized into clauses that discuss the:

a) Operations that are used by multiple OGC Web Services (OWSs), currently including the operations:

   1) GetCapabilities

2) All operations except GetCapabilities, minimum parameters
3) Exception responses to all operation requests

b) Parameters (or variables) that are used by multiple operation request and response messages, currently including the parameters:
1) "service"
2) "request"
3) "version"
4) "acceptVersions"
5) "section"
6) "updateSequence"
7) "exceptionText"
8) "exceptionCode"
9) "locator"
10) "OnlineResource"
11) "language"

c) KVP encoding of operation requests

d) XML encoding of operation requests and responses

The annexes to this document provide related informative information on:

a) Reasons for including parameters (informative)

b) UML model of aspects specified herein (informative)

c) More complete XML Schemas, ready to use (normative)

d) TBD

7 Commonly used operations

7.1 Introduction

This clause specifies the common aspects of operations which are used by multiple OGC Web Services (OWSs), currently including the operations:

a) GetCapabilities

b) All operations except GetCapabilities, minimum parameters

c) Exception responses to all operation requests (part of all operations)

NOTE The reasons for including the various parameters in operation requests and responses are briefly listed in Annex A (informative).

7.2 GetCapabilities operation

7.2.1 Introduction

This subclause partially specifies the GetCapabilities operation that shall be provided by each OGC Web Service. The GetCapabilities operation allows any client to retrieve metadata about the services available from any server that implements an OGC Web Service interface Implementation Specification. The normal response to the GetCapabilities operation is a service metadata document that is returned to the requesting client. This service metadata document contains general metadata about the server, plus metadata about its specific abilities (such as about the specific data available from that server).

7.2.2 GetCapabilities request

7.2.2.1 Request parameters

A request to perform the GetCapabilities operation shall include the parameters listed in Table 1, as each listed parameter is specified in Clause 8. Table 1 also specifies the optionality of each listed parameter, and the meaning to servers when each optional parameter is not included in the operation request.

Table 1 — Parameters used by GetCapabilities operation request

Name	Definition	Optionality
service	Service type identifier	Mandatory
request	Operation name	Mandatory unless operation name is XML element name
version	Specification version for operation	Not included (Deprecated: Optional, see version negotiation)
acceptVersions	Prioritized sequence of specification versions accepted by client, with preferred versions listed first	Mandatory (see version negotiation)
section	Name of requested section of complete service metadata document	Optional Return complete service metadata document when omitted
updateSequence	Service metadata document version, value is "increased" whenever any change is made in complete service metadata document	Optional Return latest service metadata document when omitted or not supported by server

NOTE The name capitalization rules being used here are specified in Subclause 10.2.

7.2.2.2 KVP encoding example

An Implementation Specification fragment for the GetCapabilities operation request, with specific values appropriate for WCS 1.0.0, is shown in Table 2.

Table 2 — GetCapabilities operation request URL parameters

Request parameter (Note 1)	**Optionality**	**Definition**
service=WCS	Mandatory	Service type identifier
request=GetCapabilities	Mandatory	Operation name
acceptVersions=1.0.0,0.8.3	Optional When omitted, return latest supported version	Prioritized sequence of specification versions accepted by client, with preferred versions listed first
section=All *or* =Service *or* =Capability *or* =ContentMetadata	Optional When omitted, return complete service metadata document	Section of service metadata document to be returned
updateSequence=XXX (where XXX is character string previously provides by server)	Optional When omitted, return latest service metadata document version	Service metadata document version, value is "increased" whenever any change is made in complete service metadata document
NOTE 1 All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 9.2.		

This table shall be followed by a specification of the meanings of all the allowed "section" values for that OGC Web Service. The allowed "section" values should be specified in a table such as Table 3.

Table 3 — Meanings of "section" parameter values

"section" value	**Meaning**
All	Return complete service metadata document
Service	Return Service metadata element in service metadata document
Capability	Return Capability metadata element in service metadata document
Contents	Return Contents metadata element in service metadata document
Filter	Return Filter metadata element in service metadata document

NOTE The "section" parameter specifies which XML element within a service metadata document shall be returned, instead of the complete element. To obtain two separate parts, the GetCapabilities operation can be invoked twice.

EDITOR'S NOTE Most of the "section" parameter values are expected to be common for all OGC Web Services, but the common section XML element names have not yet been specified.

A corresponding example of a GetCapabilities request message encoded using KVP is:

TBD

7.2.2.3 XML encoding

All GetCapabilities operation request parameters shall be encoded as XML attributes (not encoded as an XML element), and the parameter name capitalization listed in Table 1 shall be used for these attributes. An XML Schema fragment for a generic GetCapabilities operation request is:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/ows"
xmlns:ows="http://www.opengis.net/ows"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
version="0.0.1" xml:lang="en">
   <annotation>
      <appinfo>owsGetCapabilities.xsd 2003/10/06</appinfo>
      <documentation>
         <scope>OGC Web Service. </scope>
         <description>This XML Schema encodes the OGC Web Service (OWS)
GetCapabilities operation request. This XML Schema must be edited by
each OWS, to specify a specific value for the "service" attribute and
sometimes additional values for the "section" attribute. Primary
editor: Arliss Whiteside. </description>
         <copyright>Copyright (c) 2003 OpenGIS, All Rights
Reserved.</copyright>
      </documentation>
   </annotation>
   <!-- ========================================================
      elements and types
   ========================================================= -->
   <element name="GetCapabilities" type="ows:GetCapabilitiesType"/>
   <!-- ========================================================= -->
   <complexType name="GetCapabilitiesType">
      <annotation>
         <documentation>XML encoded GetCapabilities operation request.
This operation allows clients to retrieve metadata about a specific
service instance. In this XML encoding, no "request" parameter is
included, since the element name specifies the specific operation.
</documentation>
      </annotation>
      <sequence/>
      <attribute ref="ows:service" use="required"/>
      <attribute ref="ows:acceptVersions" use="optional"/>
      <attribute ref="ows:section" use="optional" default="All"/>
      <attribute ref="ows:updateSequence" use="optional"/>
   </complexType>
   <!-- ========================================================= -->
   <attribute name="service" type="string">
      <annotation>
         <documentation>The type of OGC Web Service that should handle
this operation request. The service must be identified in an operation
request because a single endpoint may implement more than one service,
```



```
and the same operation name may be used by multiple service types.
</documentation>
      </annotation>
   </attribute>
   <!-- ========================================================= -->
   <attribute name="acceptVersions" type="string">
      <annotation>
         <documentation>Prioritized sequence of specification versions
accepted by client, separated by commas and with preferred versions
listed first. Each version is of the Implementation Specification
(document) to which the requested operations will conform. Each version
number shall contain three non-negative integers separated by decimal
points, in the form "x.y.z". The integers y and z shall not exceed 99.
An Implementation Specification version normally specifies XML Schemas
against which an XML encoded operation response and request must
conform and should be validated. See Version negotiation text for more
information. </documentation>
      </annotation>
   </attribute>
   <!-- ========================================================= -->
   <attribute name="updateSequence" type="string">
      <annotation>
         <documentation>Service metadata document version, value is
"increased" whenever any change is made in service metadata document.
Values are selected by each server, and are always opaque to clients.
See updateSequence parameter use text for more information.
</documentation>
      </annotation>
   </attribute>
   <!-- ========================================================= -->
   <attribute name="section">
      <annotation>
         <documentation>Name of desired XML element to be returned from
a complete service metadata document. The specific values specified
here may be incomplete, and need to be extended or edited for each
specific OGC Web Service. </documentation>
      </annotation>
      <simpleType>
         <restriction base="string">
            <enumeration value="All">
               <annotation>
                  <documentation>Identifies the complete service
metadata document. </documentation>
               </annotation>
            </enumeration>
            <enumeration value="Service">
               <annotation>
                  <documentation>Identifies the Service XML element in
the service metadata document. </documentation>
               </annotation>
            </enumeration>
            <enumeration value="Capability">
               <annotation>
                  <documentation>Identifies the Capability XML element
in the service metadata document. </documentation>
               </annotation>
            </enumeration>
```



```
            <enumeration value="Contents">
               <annotation>
                  <documentation>Identifies the Contents XML element in
the service metadata document. </documentation>
               </annotation>
            </enumeration>
         </restriction>
      </simpleType>
   </attribute>
</schema>
```

This XML Schema fragment contains documentation of the meaning of each element, attribute, type, and enumeration value, and this documentation shall be considered normative.

A corresponding example of a GetCapabilities request message encoded in XML is:

TBD

7.2.3 Version negotiation

Version negotiation is performed using the mandatory acceptVersions parameter in the GetCapabilities operation request. The value of this parameter is a sequence of protocol version numbers that the client supports, in order of client preference. The server, upon receiving a GetCapabilities request, shall scan through this list and find the first version number that it supports. It shall then return a service metadata document conforming to that version of the specification, and containing that value of the “version” parameter. If the list does not contain any version numbers that the server supports, the server shall return a ServiceException with exceptionCode="VersionNegotiationFailed".

To ensure backward compatibility, clients shall also be prepared to accept an unknown response and treat this situation as an indication that version negotiation has failed. Furthermore, if a server receives a GetCapabilities request without the acceptVersions parameter, it shall return a service metadata document that is compliant to the highest protocol version that the server supports. This makes it convenient for humans to make requests manually, and allows for forward compatibility with any possible other future incarnation of version negotiation.

This version negotiation process is compatible with the old-style version negotiation mechanism that was defined in earlier versions of the various OWS specifications, using the optional “version” parameter in a GetCapabilities request. The old-style version negotiation mechanism stated that if the “version” parameter is missing, then a service metadata document compliant to the highest-supported version shall be returned. Therefore, if a new client sends a GetCapabilities request containing an acceptVersions parameter to an old server that does not recognize it, the server will return a service metadata document compliant to the highest version that it supports. The client will either recognize this version, in which case version negotiation has been successful, or it does not. In the situation where the client sees a service metadata document for a version that

it does not recognize, the client may optionally revert back to the old-style version negotiation mechanism to complete the negotiation.

A server may also optionally implement the old-style version negotiation mechanism so that old clients that send GetCapabilities requests containing a "version" parameter can be served. If both a "version" and an acceptVersions parameter exist in a GetCapabilities request, the server shall ignore the "version" parameter.

The old-style version negotiation process is as follows. In response to a **GetCapabilities** request containing a version number, an OGC Web Service **shall** either respond with output that conforms to that version of the specification, **or** negotiate a mutually agreeable version if the requested version is not implemented on the server. If no version number is specified in the request, the server **shall** respond with the highest version it understands and label the response accordingly. Version number negotiation occurs as follows:

a) If the server implements the requested version number, the server shall send that version.

b) If a version unknown to the server is requested, the server shall send the highest version it knows that is less than the requested version.

c) If the client request is for a version lower than any of those known to the server, then the server shall send the lowest version it knows.

d) If the client does not understand the new version number sent by the server, it may either cease communicating with the server or send a new request with a new version number that the client does understand but which is less than that sent by the server (if the server had responded with a lower version).

e) If the server had responded with a higher version (because the request was for a version lower than any known to the server), and the client does not understand the proposed higher version, then the client may send a new request with a version number higher than that sent by the server.

The process is repeated until a mutually understood version is reached, or until the client determines that it will not or cannot communicate with that particular server.

7.2.4 updateSequence parameter use

The **optional** updateSequence parameter can be used for maintaining the consistency of a client cache of contents of a service metadata document. The parameter value can be an integer, a timestamp in [ISO 8601:1988] format, or any other number or string. The server **may** include an UpdateSequence value in its service metadata document. If present, this value **shall** be increased when any changes are made to the complete service metadata document (for example, when new coverages are added to the service). The server is the sole judge of lexical ordering sequence. The client **may** include this parameter in its GetCapabilities request. The response of the server based on the presence and relative value of UpdateSequence in the client request and the server metadata **shall** be according to Table 4.

Table 4 — Use of updateSequence parameter

Operation request updateSequence value	Server metadata updateSequence value	Server response
None	Any	most recent service metadata document
Any	None	most recent service metadata document
Equal	Equal	exception message with exceptionCode = CurrentUpdateSequence
Lower	Higher	most recent service metadata document
Higher	Lower	exception message with exceptionCode = InvalidUpdateSequence

7.2.5 GetCapabilities response

A service metadata document response from performing the GetCapabilities operation shall contain metadata appropriate to the specific OGC Web Service and the specific server. That service metadata document shall include, in addition to other data, the parameters listed in Table 5, as each listed parameter is specified in Clause 8. This table also specifies the optionality of each listed parameter, and the meaning to the client when each optional parameter is not included in the service metadata document.

Table 5 — Parameters included in service metadata document

Name	Definition	Optionality
version	Specification version for operation	Mandatory
updateSequence	Service metadata document version, value is "increased" whenever any change is made in complete service metadata document	Optional Omitted when parameter not supported by server

NOTE The term "Capabilities XML" document was previously usually used for what is here called "service metadata" document. The term "service metadata" is now used because it is more descriptive and is compliant with ISO 19119. This "service metadata" includes metadata for a specific server.

EDITOR'S NOTE One of the open issue is where these parameters should be included in the service metadata document, considering the various allowed values of the "section" parameter. Of course, a service metadata document must contain much more metadata, some of which is common but not yet specified in this document.

7.3 All operations except GetCapabilities, minimum parameters

7.3.1 Introduction

This subclause specifies minimum abilities of all operations except GetCapabilities that are implemented by any OGC Web Service.

7.3.2 Operation request

7.3.2.1 Request parameters

A request to perform any operation except GetCapabilities shall include, in addition to operation-specific parameters, the parameters listed in Table 6, as each listed parameter is specified in Clause 8. Table 6 also specifies the optionality of each listed parameter.

Table 6 — Parameters used by all operation requests except GetCapabilities

Name	Definition	Optionality
service	Service type identifier	Mandatory
request	Operation name	Mandatory unless operation name is XML element name
version	Specification version for operation	Mandatory

7.3.2.2 KVP encoding example

TBD

7.3.2.3 XML encoding

TBD

7.3.3 Operation response

A response from any operation except GetCapabilities shall include, in addition to operation-specific data, the parameters listed in Table 7, as each listed parameter is specified in Clause 8. Table 7 also specifies the optionality of each listed parameter, and the meaning to a client when each optional parameter is not included in the operation response. (TBR)

Table 7 — Parameters used by all operation responses except GetCapabilities

Name	Definition	Optionality
TBD		

7.4 Service exception responses to all operation requests

7.4.1 Introduction

Whenever a server detects an exception condition while responding to an operation request, and cannot produce a normal response to that operation, the server shall respond to the client using a Service Exception Message. This subclause specifies the service exception message response to all operation requests for all OGC Web Services.

7.4.2 Exception message contents

Each Service Exception Message shall be encoded in XML and shall contain zero (TBR) or more Service Exception elements. Each Service Exception element shall contain the parameters listed in Table 8, as each listed parameter is specified in Clause 8. Table 8 also specifies the optionality of each listed parameter, and the meaning to a client when each optional parameter is not included.

Table 8 — Parameters in Service Exception element

Name	Definition	Optionality
(exceptionText, not named)	Text describing the specific exception represented by the exceptionCode	Mandatory
exceptionCode	Code representing the type of this exception	Mandatory
locator	Indicator of the location in the client's operation request where this exception was encountered	Optional Omitted when no useful value available

EDITOR'S NOTE An open issue is the meaning if no ServiceException element is included in a Service Exception Message. Another open issue is the meaning when more than one ServiceException elements is included in a Service Exception Message.

The ServiceException elements (if any) shall be interpreted by clients as being independent of one another (not hierarchical, TBR).

In addition to the Service Exception elements, a Service Exception Message shall also contain the parameters listed in Table 9, as each listed parameter is specified in Clause 8. Table 9 also specifies the optionality of each listed parameter, and the meaning to a client when each optional parameter is not included.

Table 9 — Parameters in Service Exception Message

Name	Definition	Optionality
service	Service type identifier	Mandatory
OnlineResource	Reference to on-line resource used by this operation request	Mandatory
version	Specification version for operation request	Mandatory
language	Language used by exception text and exceptionCodes	Optional Can omit when English

EDITOR'S NOTE An open issue is whether the "service" and "OnlineResource" parameters are really useful in an OWS Service Exception Message. Another open issue is whether the "language" parameter should be in the Service Exception Message (as shown) or in each Service Exception element.

7.4.3 XML encoding

All Service Exception Message request parameters shall be encoded as XML attributes (not encoded as an XML element), except:

a) exceptionText, which is encoded as the string contents of an XML element named ServiceException

b) OnlineResource, which is encoded in an XML element

The parameter name capitalization listed in Tables 8 and 9 shall be used for these attributes and elements. An XML schema fragment for a generic Service Exception Message is:

```
<?xml version="1.0" encoding="UTF-8"?>
<schema targetNamespace="http://www.opengis.net/ows"
xmlns:ows="http://www.opengis.net/ows"
xmlns:xlink="http://www.w3.org/1999/xlink"
xmlns="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified"
version="0.0.1" xml:lang="en">
   <annotation>
      <appinfo>serviceException.xsd 2003/10/06</appinfo>
      <documentation>
         <scope>OGC Web Service. </scope>
         <description>This XML Schema encodes the Service Exception
Message response to an OGC Web Service. Primary editor: Arliss
Whiteside. </description>
         <copyright>Copyright (c) 2003 OpenGIS, All Rights
Reserved.</copyright>
      </documentation>
   </annotation>
   <!-- ============================================================
      includes and imports
   ============================================================= -->
   <import namespace="http://www.w3.org/1999/xlink"
schemaLocation="http://schemas.opengis.net/gml/3.0.0/xlink/xlinks.xsd"/
>
   <!-- ============================================================

      elements and types
   ============================================================= -->
   <element name="ServiceExceptionReport">
      <annotation>
         <documentation>The Service Exception Report contains zero or
more Service Exception elements that each describes a service
exception. This report is returned to the client that requested an
operation when the server detects an exception condition while
processing that operation request. </documentation>
      </annotation>
      <complexType>
         <sequence>
            <element name="ServiceException"
type="ows:ServiceExceptionType" minOccurs="0" maxOccurs="unbounded">
               <annotation>
```



```
                  <documentation>Unordered list of zero or more
ServiceException elements that each describes a service exception.
These ServiceException elements shall be interpreted by clients as
being independent of one another (not hierarchical, TBR).
</documentation>
               </annotation>
            </element>
            <element name="OnlineResource">
               <annotation>
                  <documentation>Reference to on-line resource used by
this operation request. </documentation>
               </annotation>
               <complexType>
                  <attributeGroup ref="xlink:simpleLink"/>
               </complexType>
            </element>
         </sequence>
         <attribute name="service" type="string">
            <annotation>
               <documentation>The type of OGC web service that handled
this operation request. </documentation>
            </annotation>
         </attribute>
         <attribute name="version" type="string" use="required">
            <annotation>
               <documentation>The version of the Implementation
Specification (document) to which the requested operation conformed. A
version number shall contain three non-negative integers separated by
decimal points, in the form "x.y.z". The integers y and z shall not
exceed 99. </documentation>
            </annotation>
         </attribute>
         <attribute name="language" type="string" use="optional"
default="eng">
            <annotation>
               <documentation>The language attribute identifies the
language used by the exception text and exceptionCodes. These language
code values shall use ISO (TBR) 639.2. </documentation>
            </annotation>
         </attribute>
      </complexType>
   </element>
   <!-- ======================================================== -->
   <complexType name="ServiceExceptionType">
      <annotation>
         <documentation>A ServiceException element describes an
exception which the service wishes to convey to the client application.
The base string contains the exception text describing the specific
exception represented by the exceptionCode. </documentation>
      </annotation>
      <simpleContent>
         <extension base="string">
            <attribute name="exceptionCode" type="string">
               <annotation>
                  <documentation>A code representing the type of this
exception, which shall be selected from a set of exceptionCode values
specified for the specific service. </documentation>
```

```
            </annotation>
         </attribute>
         <attribute name="locator" type="string" use="optional">
            <annotation>
               <documentation>Indicator of the location in the
client's request where this exception was encountered. If the request
included one or more "handle" attributes, a handle value may be used to
identify the offending part of the request. If a parameter value was
missing or invalid, the parameter name should be used to identify the
incorrect part of the request. Otherwise the service may try to use
other means to locate the exception, such as line numbers or bytes
offset from the beginning of the request. </documentation>
            </annotation>
         </attribute>
      </extension>
   </simpleContent>
</complexType>
</schema>
```

This XML Schema fragment contains documentation of the meaning of each element, attribute, and/or type, and this documentation shall be considered normative.

7.4.4 exceptionCode values

Each Implementation Specification shall specify a set of standard allowed values for the exceptionCode attribute, as needed for that OGC Web Service. The allowed standard exceptionCode values should be specified in a table such as Table 10.

Table 10 — Standard exception codes

exceptionCode value	Meaning
OperationNotSupported	Request is for an operation that is not supported by this server.
VersionNegotiationFailed	List of versions in "acceptVersions" parameter value did not include any version supported by this server.
CurrentUpdateSequence	Value of (optional) updateSequence parameter in GetCapabilities operation request is equal to current value of service metadata updateSequence number.
InvalidUpdateSequence	Value of (optional) updateSequence parameter in GetCapabilities operation request is greater than current value of service metadata updateSequence number.
MissingParameterValue	Request does not include a parameter value, and this server did not declare a default value for that parameter.
InvalidParameterValue	Request contains an invalid parameter value.
InvalidFormat	Request specifies a Format not offered by this server.

EDITOR'S NOTE An open issue is whether an exceptionCode value of "NoApplicableCode" should be specified here or elsewhere. Another open issue is whether a server implementation can specify and use exceptionCode values not specified in the Implementation Specification. Another open issue is whether all the exceptionCode values listed above are applicable to all OGC Web Services. Of course, most OGC Web Services will need to specify additional allowed exceptionCode values.

An example of a Service exception message encoded in XML is:

TBD

8 Commonly used parameters

8.1 Introduction

This clause more completely specifies many of the parameters used by the operation requests and responses specified in the preceding clause. The parameter names, meanings, and formats shall be as specified in Tables 11 and 12. Subsequent subclauses provide more information about some of these parameters.

Table 11 — Definitions of operation request and response parameters

Name	Definition	Data type and value
service	Service type identifier	Character String type, not empty Value is OWS type abbreviation (e.g., "WMS", "WFS")
request	Operation name	Character String type, not empty Value is operation name (e.g., "GetCapabilities")
version	Specification version for operation	Character String type, not empty Value format is x.y.z, where x, y, and z are non-negative integers separated by decimal points (e.g., "2.1.3") Value specified by each Implementation Specification and Schemas version
acceptVersions	Prioritized sequence of specification versions accepted by client, with preferred versions listed first	Character String type, not empty Value format is sequence of x.y.z "version" values, separated by commas (e.g., "2.1.3,2.1.0,1.7.1")
section	Name of requested section of complete service metadata document	Character String type, not empty Allowed values specified by each OWS specification (e.g., "service")
updateSequence	Service metadata document version, value is "increased" whenever any change is made in complete service metadata document	Character String type, not empty Values are selected by each server, and are always opaque to clients

Table 12 — Definitions of service exception parameters

Name	Definition	Data type and value
exceptionText	Text describing the specific exception represented by the exceptionCode	Character String type, not empty
exceptionCode	Code representing the type of this exception	Character String type, not empty Value selected from a set of values specified for each specific service
locator	Indicator of the location in the client's operation request where this exception was encountered	Character String type, not empty Value: see Note 1
OnlineResource	Reference to on-line resource used by this operation request	URL type Value is URL that received this operation request
language	Language used by exception text and exceptionCodes	Character String type, not empty Values specified by ISO (TBR) 639.2 (e.g., “eng” for English)
NOTE 1 If the operation request included one or more "handle" attributes, a handle value may be used to identify the offending part of the request. If a parameter value was missing or invalid, the parameter name should be used to identify the incorrect part of the request. Otherwise the service may try to use other means to locate the exception, such as line numbers or bytes offset from the beginning of the request		

NOTE The name capitalization rules being used here are specified in Subclause 10.2.

EDITOR’S NOTE The data type of many of these parameters is specified as “Character String type, not empty”. In the XML Schema fragments suggested herein, these parameters are encoded with the xsd:string type, which does NOT require that these strings not be empty. It is possible to develop an XML Schema encoding that would require these Character Strings to not be empty.

8.2 Version parameter

Each OWS Implementation Specification revision shall specify a version number, which enables interacting clients and servers to agree on which version of the specification they are conforming to. A version number shall contain three non-negative integers separated by decimal points, in the form "x.y.z". The integers y and z shall not exceed 99.

Through the evolution of specifications, each service will have a number of versions defined for it, each with a different version number. Each OWS shall have its own sequence of version numbers; the version numbers of different services are independent and therefore may overlap. When the protocol version number changes, it shall increase monotonically and shall comprise no more than three integers separated by decimal points, with the first integer being the most significant. There may be gaps in the numerical sequence, and some numbers may denote draft versions. Servers and their clients need not support all defined versions, but are encouraged to support multiple versions.

9 KVP encoding

9.1 Introduction

This clause specifies the Key Value Pair (KVP) encoding of the operation request parameters specified in Clauses 7 and 8. TBD

9.2 Capitalization

The capitalization of parameter names when KVP encoded shall be case insensitive, meaning that parameter names may have mixed case or not.

EXAMPLES The "request" parameter name could be REQUEST, request, Request, or ReQuEsT.

NOTE The XML capitalization is uniformly used in Clauses 7 and 8 plus Annex A of this document.

The capitalization of parameter values when encoded using Key Value Pairs shall be as used in Clause 7 of this document. More generally, all value strings shall have the first word and any subsequent words in the name capitalized. All other letters will be lower case.

EXAMPLE One possible "request" parameter value shall be GetCapabilities.

10 XML encoding

10.1 Introduction

This clause specifies the XML encoding of the operation request and response parameters specified in Clauses 7 and 8. TBD

10.2 Capitalization

The capitalization of parameter and operation names when encoded as XML elements and attributes shall be as used in Clauses 7 and 8 plus Annex A of this document. More generally, these name capitalization rules shall be used:

a) All names of XML elements shall have the first word and any subsequent words in the name capitalized. All other letters will be lower case.

b) All names of XML attributes shall have the first word in lower case and any subsequent word in the name capitalized. All other letters shall be lower case.

EXAMPLES The GetCapabilities operation request element name shall be GetCapabilities. The acceptVersions parameter attribute name shall be acceptVersions.

The capitalization of parameter values when encoded as XML stings shall be as used in Clause 7 of this document. More generally, all XML string values shall have the first word and any subsequent words in the name capitalized. All other letters will be lower case.

EXAMPLE One possible “request” parameter value shall be GetCapabilities.

10.3 XML Schema extension and restriction

TBD

Annex A
(informative)

Reasons for parameters

A.1 Introduction

This annex briefly states the reasons for deciding to include the parameters in the various operation request and response messages specified in this document.

A.2 Reasons for GetCapabilities parameters

The reasons for deciding to include the parameters listed in the GetCapabilities operation request and response messages, in Subclause 7.2, are briefly stated in the Table A.1.

Table A.1 — Reasons for GetCapabilities parameters

Name	Reason
service	The service must be identified in all operation requests because a single endpoint may implement more than one service, and the same operation name may be used by multiple service types.
request	The requested operation must be identified in all operation requests because a single endpoint may implement more than one operation.
version	The specification version must be identified in all GetCapabilities operation responses because a single server may support more than one version of a specification, and the client needs to know the specific version of the operations supported (see Subclause 7.2.3). (Deprecated: The specification version was optional in the GetCapabilities operation request for all OGC web services to support client-server version negotiation.)
acceptVersions	The acceptVersions is mandatory in the GetCapabilities operation request for all OGC Web Services to support efficient client-server version negotiation as specified in Subclause 7.2.3.
section	The section should be optional in the GetCapabilities operation request for all OGC Web Services to allow clients to request, and servers to respond, with only the needed part of the complete service metadata document. Inclusion of more than one section value is not supported since a client can request another GetCapabilities operation to retrieve another document section.
updateSequence	The updateSequence should be optional the GetCapabilities operation request for all OGC Web Services to allow clients to request, and servers to respond, with a service metadata document only when it has been updated since the last version returned (see Subclause 7.2.4).

A.2 Reasons for all operations except GetCapabilities minimum parameters

The reasons for deciding to include the minimum parameters listed for all operation request and response messages except GetCapabilities, in Subclause 7.3, are briefly stated in the Table A.2.

Table A.2 — Reasons for all operations except GetCapabilities minimum parameters

Name	Reason
service	The service must be identified in all operation requests because a single endpoint may implement more than one service, and the same operation name may be used by multiple service types.
request	The requested operation must be identified in all operation requests because a single endpoint may implement more than one operation.
version	The specification version must be identified in all operation requests except GetCapabilities because a single server may support more than one version of a specification, and thus needs to know the specific version of the operation being requested.

A.3 Reasons for Exception Message parameters

The reasons for deciding to include the parameters listed for Exception Messages, in Subclause 7.4, are briefly stated in the Table A.3.

Table A.3 — Reasons for Exception Message parameters

Name	Reason
exceptionText	TBD
exceptionCode	TBD
locator	TBD
service	TBD
OnlineResource	TBD
version	TBD
language	TBD

Bibliography

[1] ISO/TS 19103, Geographic information — Conceptual schema language

[2] OGC 00-014r1, Guidelines for Successful OGC Interface Specifications,

[3] OGC 01-068r3, Web Map Service Implementation Specification, v1.1.1

[4] OGC 02-058, Web Feature Service Implementation Specification, v1.0.0

[5] OGC 02-61r2, Web Coordinate Transformation Service Implementation Specification, draft

[6] OGC 03-065r6, Web Coverage Service Implementation Specification, v1.0.0

[7] OGC 03-086, Geographic information — Web Map Service interface, draft