

Open Geospatial Consortium Inc.

Date: 2007-05-02

Reference number of this OGC® document: **OGC 07-004**

Version: 0.0.1

Category: OpenGIS® Discussion Paper

Editor: Panagiotis (Peter) A. Vretanos

OWS-4 GeoDDS Mass Market (formerly GeoRSS) Interoperability Program Report

Copyright notice

Copyright © 2007 Open Geospatial Consortium. All Rights Reserved

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

Warning

This document is not an OGC Standard. This document is an OGC Discussion Paper and is therefore not an official position of the OGC membership. It is distributed for review and comment. It is subject to change without notice and may not be referred to as an OGC Standard. Further, an OGC Discussion Paper should not be referenced as required or mandatory technology in procurements.

Document type: OpenGIS® Discussion Paper
Document subtype: Engineering Specification
Document stage: Draft
Document language: English

Contents

1 SCOPE	1
2 CONFORMANCE	1
3 NORMATIVE REFERENCES	1
4 TERMS AND DEFINITIONS	3
5 CONVENTIONS	4
5.1 Symbols (and abbreviated terms).....	4
5.2 Document terms and definitions.....	5
6 BASIC SERVICE ELEMENTS	5
6.1 Version numbering and negotiation.....	5
6.2 General request encoding rules.....	5
6.2.1 <i>HTTP method bindings</i>	5
6.2.2 <i>HTTP GET request encoding</i>	5
6.2.3 <i>HTTPS</i>	6
6.2.4 <i>General HTTP response rules</i>	6
6.2.5 <i>Exception reporting</i>	7
7 OGC PUBLISH-SUBSCRIBE BASIC SERVICE (OPS-B)	8
7.1 Introduction.....	8
7.2 Simplifying assumptions.....	8
7.2.1 <i>Standard message output</i>	8
7.2.2 <i>User identity</i>	8
7.2.3 <i>Requirements</i>	9
7.3 OPS-B operations.....	9
7.3.1 <i>Introduction</i>	9
7.3.2 <i>CreateSubscription operation</i>	9
7.3.2.1 Request encoding.....	9
7.3.2.2 Response encoding.....	10
7.3.3 <i>Remove subscription</i>	12
7.3.3.1 Request encoding.....	12
7.3.3.2 Response encoding.....	13
7.3.4 <i>GetSubscriptions operation</i>	13
7.3.4.1 Request encoding.....	13
7.3.4.2 Response encoding.....	14
7.3.5 <i>Subscribe operation</i>	16
7.3.5.1 Request encoding.....	16
7.3.5.2 Response encoding.....	16
7.3.6 <i>GetUserSubscriptions</i>	18
7.3.6.1 Request encoding.....	18
7.3.6.2 Response encoding.....	19
7.3.7 <i>PauseSubscription operation</i>	22
7.3.7.1 Request encoding.....	22
7.3.7.2 Response encoding.....	23
7.3.8 <i>ResumeSubscription operation</i>	25
7.3.8.1 Request encoding.....	25
7.3.8.2 Response encoding.....	26
7.3.9 <i>Cancel Subscription operation</i>	27
7.3.9.1 Request encoding.....	27

7.3.10 Response encoding.....	28
8 WEB FEATURE SERVICE - SIMPLE.....	28
8.1 Introduction	28
8.2 Simplifying assumptions.....	28
8.3 Common service elements.....	29
8.4 Common request parameters.....	29
8.4.1.1 Service parameter.....	29
8.4.1.2 Version parameter.....	29
8.5 GetCapabilities operation (mandatory).....	29
8.5.1 Introduction.....	29
8.5.2 Request encoding.....	30
8.5.3 Response encoding.....	30
8.5.4 Example.....	31
8.6 DescribeFeatureType operation (optional).....	32
8.6.1 Introduction.....	32
8.6.2 Request encoding.....	32
8.6.3 Response encoding.....	33
8.6.3.1 Introduction.....	33
8.6.3.2 BXFS data description	33
8.6.3.3 Example.....	36
8.6.3.4 Plain text data description.....	36
8.6.3.5 Geography markup language(GML).....	37
8.7 GetFeature operation (mandatory).....	37
8.7.1 Introduction.....	37
8.7.2 Request encoding.....	37
8.7.2.1 Parameter discussion.....	38
8.7.3 Response encoding.....	42
8.7.3.1 WFS-S feature collection using BXFS elements.....	42
8.7.3.1.1 Example.....	43
8.7.3.2 Atom+GeoRSS response.....	44
A.1 Introduction.....	46
A.1 Introduction.....	46
A.1 Introduction.....	46
A.1 Introduction.....	46
A.1 Introduction.....	46
A.2 WFS-S response schema.....	46
A.2 WFS-S response schema.....	46
A.2 WFS-S response schema.....	46
A.2 WFS-S response schema.....	46
A.2 WFS-S response schema.....	46
B.1 XML schema for BXFS.....	50
B.1 XML schema for BXFS.....	50
B.1 XML schema for BXFS.....	50
B.1 XML schema for BXFS.....	50
B.1 XML schema for BXFS.....	50

Tables

TABLE 1 – A GENERAL WEB SERVICE REQUEST.....	6
TABLE 2 – CREATESUBSCRIPTION KVPREQUEST ENCODING.....	9
TABLE 3 – REMOVESUBSCRIPTION KVPREQUEST ENCODING.....	13
TABLE 4 – GETSUBSCRIPTIONS KVP REQUEST ENCODING.....	13
TABLE 5 – SUBSCRIBE KVP REQUEST ENCODING.....	16
TABLE 6 – GETUSERSUBSCRIPTIONS KVP REQUEST ENCODING.....	18
TABLE 7 – PAUSESUBSCRIPTION KVPREQUEST ENCODING.....	22
TABLE 8 – RESUMESUBSCRIPTION KVP REQUEST ENCODING.....	25
TABLE 9 – CANCELSUBSCRIPTION KVPREQUEST ENCODING.....	27
TABLE 10 – COMMON REQUEST PARAMETER.....	29
TABLE 11 – GETCAPABILITIES REQUEST ENCODING.....	30
TABLE 12 – DESCRIBEFEATURETYPE REQUESTENCODING.....	33
TABLE 13 – CORRELATING DATATYPE WITH MAXLENGTH, PRECISION AND SCALE ATTRIBUTES.....	35
TABLE 14 – GETFEATURE REQUEST ENCODING.....	37
TABLE 15 – STANDARD OUTUTFORMAT VALUES.....	40

i.Preface

This Interoperability Program Report provides an overview of the OGC Publish/Subscribe Basic (OPS-B) and the Web Feature Service Simple (WFS-S) services as they were implemented for the OWS-4 test bed. These services are simplifications and generalizations of existing OGC services. OPS-B is a simplification and generalization of the notions developed for the OGC Publish Subscribe Service for SWE. WFS-S is a simplification and generalization of the Web Feature Service Implementation Specification (04-094).

The common thread is the definition of simplified OGC web service API's and data structures in order to reduce implementation barriers and promote their use in mainstream consumer, media, and business IT infrastructures.

The service endpoints implemented for OWS-4 are:

<http://eo1.geoblivi.org/ops>

<http://eo1.geoblivi.org/wfs>

Additional endpoints implementing parts of this specification, but not specifically participating in this thread in OWS-4, can be found at:

<http://www.cubewerx.com/ows4/wfss/wfss.cgi>

<http://wiki.hsr.ch/Prog1Java/wiki.cgi?WFSSimple4GeoRSS>

(soon to become: <http://geometa.info/demo/wfsimple?REQUEST=GetFeature>)

ii. Submitting organizations

The following organizations submitted this document to the Open Geospatial Consortium Inc.

Cubewerx Inc.

Vightel Corporation

Innovative Solutions Inc

iii. Document contributor contact points

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

Contact	Company	Email
Cappelaere, Pat	Vightel Corp.	pat[at]cappelaere.com
Derezinski, Linda	Innovative Solutions	linda[at]i-sol.biz
Singh, Raj	OGC	rsingh[at]opengeospatial.org
Vretanos, Panagiotis(Peter) A. (ed.)	CubeWerx Inc.	pvretano[at]cubewerx.com

iv.Revision history

Date	Release	Editor	Primary clauses modified	Description
2007-01-22	0.0.1	P.A.V.	All	Initial version.
2007-05-02	0.0.2	Carl Reed	Various	Prepare document for posting as a DP

v.Changes to the OpenGIS® Abstract Specification

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

vi.Future work

Foreword

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 specification set forth in this document, and to provide supporting documentation.

This is the first version of this document.

The services described in this document are related to services and components describe in the “Web Feature Service Implementation Specification” (OGC 04-094), the “Filter Encoding Implementation Specification” (OGC 04-095), “An Introduction to GeoRSS: A Standard Based Approach for Geo-enabling RSS feeds” (OGC 06-050r3), the “Web Notification Service” (OGC 06-095) and notions developed for the OGC Publish/Subservice Service for SWE.

This document contains 2 annexes of which Annex A, B are normative.

OGC 07-004 consists of the following parts:

–Part 1: OGC Publish Subscribe Service – Basic (OPS-B)

–Part 2: Web Feature Service - Simple (WFS-S)

Introduction

The OGC Web Services, Phase 4 (OWS-4) Testbed was an initiative of OGC's Interoperability Program to collaboratively extend and demonstrate OGC's baseline for geospatial interoperability. The main development of OWS-4 was conducted from June to December 2006 with the following outcomes:

59 Components were implemented and deployed in interoperability testing. Components were developed in 7 threads:

- Sensor Web Enablement (SWE)
- Geo Processing Workflow (GPW)
- Geo-Decision Support (GeoDSS)
- Geo-Digital Rights Management (GeoDRM)
- CAD / GIS / BIM (CGB)
- OGC Location Services (OpenLS)
- Compliance Testing (CITE)

This document reports on one component of the work completed in the Geo-Decision Support (GeoDSS) Thread. Geo-Decision Support Services (GeoDSS) provide interoperable access to distributed geospatial web services to aid decision makers in forming, analyzing, and selecting alternatives. GeoDSS utilizes workflow management to produce context-specific results from information and knowledge from multiple communities. The GeoDSS subtask extended the Decision Support and the Information Interoperability work done in OWS-3 to include multilingual interoperability and compressed GML data.

OWS-4 GeoDDS Mass Market (formerly GeoRSS) Interoperability Program Report

1 Scope

This OpenGIS® document describes the API for two web services capable of generating several simplified data formats including GeoRSS and the Basic XML Feature Schema (BXFS). This first service, called OGC Publish Subscribe Service – Basic (OPS-B), employs a publish/subscribe model to “push” Atom+GeoRSS feeds to clients who have subscribed to receive specific notifications. The second service, called Web Feature Service – Simple (WFS-S), uses a request-response model to allow clients to query the contents of the service and receive the response in a variety of service-advertised data formats. These output formats can include plain text, BXFS and Atom+GeoRSS feeds. WFS-S is a simplified version of the WFS (OGC 04-095)

2 Conformance

This Conformance clause is required in an OGC Implementation Specification, but is not required for an IP IPR, DIPR, or Discussion Paper.

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.

IETF RFC 2045 (November 1996), *Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies*, <http://www.ietf.org/rfc/rfc2045.txt>

IETF RFC 2046 (November 1996), *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types*, <http://www.ietf.org/rfc/rfc2046.txt>

IETF RFC 2141 (May 1997), *URN Syntax*, R. Moats,
<http://www.ietf.org/rfc/rfc2141.txt>

IETF RFC 2396 (August 1998), *Uniform Resource Identifiers (URN): Generic Syntax*,
<http://www.ietf.org/rfc/rfc2396.txt>

OGC 07-004

IETF RFC 2616 (June 1999), *Hypertext Transfer Protocol – HTTP/1.1*,
<http://www.ietf.org/rfc/rfc2616.txt>

IETF RFC 4287 (December 2005), *The Atom Syndication Format*,
<http://www.ietf.org/rfc/rfc4287>

IANA, Internet Assigned Numbers Authority, *MIME Media Types*, available at
<http://www.iana.org/assignments/media-types>

ISO 8601:2004, *Data elements and interchange formats – Information interchange – Representation of dates and times*

ISO 19013: 2005, *Geographic Information – Conceptual schema language*

ISO 19105: 2005, *Geographic Information – Conformance and Testing*

ISO 19109: 2005, *Geographic Information – Rules for application schema*

ISO 19119: 2005, *Geographic Information – Services*

ISO 19136: --¹, *Geographic Information - Geography Markup Language (GML)*

ISO 19143: --¹, *Geographic Information - Filter Encoding*

NCSA, *The Common Gateway Interface*,
<http://hoohoo.ncsa.uiuc.edu/cgi/>

OGC 05-008c1, *OGC Web Services Common Specification*

OGG 05-010, *URNs of definitions in ogc namespace*

OGC 06-050r3, *GeoRSS White Paper*

RSS Advisor Board, *RSS 2.0 Specification*,
<http://www.rssboard.org/rss-2-0>

W3C Recommendation (): *Namespaces in XML*,
<http://www.w3.org/TR/1999/REC-xml-names-19990114>

W3C Recommendation (): *XML Information Set*,
<http://www.w3.org/TR/xml-infoset/>

W3C Recommendation (), *Extensible Markup Language (XML) 1.0 (Third Edition)*,
<http://www.w3c.org/TR/REC-xml>

W3C Recommendation (): *XML Schema Part 1: Structures*,
<http://www.w3.org/TR/2004/REC-xmlschema-1-20041028>

W3C Recommendation (): *XML Schema Part 2: Datatypes*,
<http://www.w3.org/TR/2004/REC-xmlschema-2-20041028>

¹ To be published.

W3C Recommendation (): *XML Linking Language (XLink) Version 1.0*,
<http://www.w3.org/TR/2001/REC-xlink-20010627/>

W3C Recommendation (24 June 2003) SOAP Version 1.2 Part 1: Messaging Framework,
<http://www.w3.org/TR/SOAP/>

4 Terms and definitions

For the purposes of this specification, the definitions given in OWS Common Implementation Specification [OGC05-008c1] and in OGC Abstract Specification Topic 2: Spatial referencing by coordinates [OGC 05-103] shall apply as do the following terms:

4.1

aggregator

Two main types of aggregators: web-based aggregators and desktop/software aggregators. Web-based aggregators allow individuals to subscribe to feeds online and read feeds in a web browser. Desktop aggregators are software programs installed locally that update when feeds are updated. The aggregator shows new information and allows for users to read feeds.

4.2

atom

An XML-based file format. Although Atom is not the same as RSS, it has similar functions. FeedForAll will convert Atom feeds to RSS v2 feeds.

4.3

feed

An XML file containing headlines and descriptions also called news feed, content feed, XML feed or web feed.

4.4

feed validator

Tests the formatting of feeds to ensure that they are correctly formed.

4.5

RSS

RSS or Really Simple Syndication is a defined standard with the specific purpose of delivering updates to web-based content. Using this standard, webmasters provide headlines and fresh content in a succinct manner. Meanwhile, consumers use RSS readers and news aggregators to collect and monitor their favourite feeds in one centralized program or location. Content viewed in the RSS reader or news aggregator is known as an RSS feed or just a feed.

4.6

GeoRSS

GeoRSS are Geographically Encoded Objects for RSS Feeds. It is a simple method for encoding “where” the feed element is located. See www.georss.org for further information.

5Conventions

5.1Symbols (and abbreviated terms)

CRS	Coordinate Reference System
DCP	Distributed Computing Platform
EPSG	European Petroleum Survey Group
GML	Geography Markup Language
GeoRSS	Geographically Encoded Objects for RSS
HTTP	Hypertext Transfer Protocol
ISO	International Organization for Standardization
KVP	Keyword Value Pair
MIME	Multipurpose Internet Mail Extensions
OGC	Open Geospatial Consortium
OPS-B	OpenGIS Publish/Subscribe Service - Basic
OWS	OGC Web Service, or Open Web Service
RSS	Really Simple Syndication
TBD	To Be Determined
TBR	To Be Reviewed
UML	Unified Modeling Language
URI	Universal Resource Identifier
URL	Uniform Resource Locator
URN	Universal Resource Name
WCS	Web Coverage Service
WFS	Web Feature Service
WFS-S	Web Feature Service - Simple
WMS	Web Map Service
XML	Extensible Markup Language
1D	One Dimensional
2D	Two Dimensional
3D	Three Dimensional
4D	Four Dimensional

5.2 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 Basic service elements

6.1 Version numbering and negotiation

The OWS-4 work described in this document did not indicate that a version parameter was part of any request and so version numbering and negotiation is not discussed.

Editor's Note: As per 05-008, all OGC service should include a VERSION parameter in their encoding to discriminate one version of an API with the next. Also, including a SERVICE parameter is a good idea to support endpoints that implement more than one service.

6.2 General request encoding rules

6.2.1 HTTP method bindings

The HTTP/1.1 specification [RFC 2616] defines eight methods for manipulating and retrieving representations of resources. Only the GET method is used in this specification and all of the service operations defined in this application profile shall be bound to the HTTP GET method.

6.2.2 HTTP GET request encoding

An Online Resource URL intended for HTTP GET requests, is, in fact, only a URL prefix to which additional parameters shall be appended in order to construct a valid Operation request. A URL prefix is defined as an opaque string including the protocol, hostname,

OGC 07-004

optional port number, path, a question mark '?', and, optionally, one or more server-specific parameters ending in an ampersand '&'. The prefix uniquely identifies the particular service instance. A client appends the necessary request parameters as name/value pairs in the form "name=value&". The resulting URL shall be valid according to the HTTPCommon Gateway Interface (CGI) standard, which mandates the presence of '?' before the sequence of query parameters and the '&' between each parameter. As with all CGI applications, the query URL is encoded as described in [3] to protect special characters. Operation parameter names shall be case insensitive. Operation parameters names shall be case sensitive.

The URL prefix shall end in either a '?' (in the absence of additional server-specific parameters) or a '&'. In practice, however, Clients should be prepared to add a necessary trailing '?' or '&' before appending the Operation parameters defined in this specification in order to construct a valid request URL.

Table 1 summarizes the components of an operation request URL.

Table 1 – A general web service request

URL Component	Description
http://host[:port]/path?{name[=value]&}	URL prefix of service operation. [] denotes 0 or 1 occurrence of an optional part; {} denotes 0 or more occurrences. The prefix is entirely at the discretion of the service provider.
name=value&	One or more standard request parameter name/value pairs defined by a web feature service. The actual list of required and optional parameters is mandated for each operation by the appropriate OWS specification.

6.2.3 HTTPS

In addition to or instead of offering web feature services using the HTTP protocol, a service provider may offer web feature services using HTTPS. HTTPS is HTTP over a secure communication channel that allows encrypted information to be transferred between machines over the World Wide Web.

The use of HTTPS does not affect the description of the requests and responses described in this specification but may require additional actions to be taken on both the client and the service in order to initiate the secure communication. The description of those actions is beyond the scope of this specification.

6.2.4 General HTTP response rules

Upon receiving a valid request, the service shall send a response corresponding exactly to the request as detailed in this specification.

Upon receiving an invalid request, the service shall issue a Service Exception as described in sub clause 6.2.5.

NOTE: As a practical matter, in the WWW environment a client should be prepared to receive either a valid result, or nothing, or any other result. This is because the client may itself have formed a non-conforming request that inadvertently triggered a reply by something other than a web feature service, because the Service itself may be non-conforming.

Response objects shall be accompanied by the appropriate Multipurpose Internet Mail Extensions (MIME) type for that object.

Response objects should be accompanied by other HTTP entity headers as appropriate and to the extent possible. In particular, the Expires and Last-Modified headers provide important information for caching; Content-Length may be used by clients to know when data transmission is complete and to efficiently allocate space for results, and Content-Encoding or Content-Transfer-Encoding may be necessary for proper interpretation of the results.

6.2.5 Exception reporting

In the event that an OPS-B or WFS-S service encounters an error while processing a request or receives an unrecognised request, it shall generate an XML document indicating that an error has occurred. The format of the XML error response is specified by and shall validate against, the exception response schema defined in clause 8 of the OWS Common Implementation Specification [OGC05-008].

An **<ExceptionReport>** element may contain one or more processing exceptions specified using the **<Exception>** element. The mandatory **version** attribute is used to indicate the version of the service exception report schema. For this version of the specification, this value is fixed at 1.1.0. The optional **language** attribute may be used to indicate the language used. The code list for the language parameter is defined in IETF RFC 1766.

Individual exception messages are contained within the **<ExceptionText>** element. The mandatory **code** attribute shall be used to associate an exception code with the accompanying message. The optional **locator** attribute may be used to indicate where an exception was encountered within the request that generated the error. A number of elements defined in this document include a **handle** attribute that may be used to associate a mnemonic name with the element. If such a **handle** exists, its value may be reported using the **locator** attribute of the **<ExceptionText>** element. If the **handle** attribute is not specified, then a web feature server implementation may attempt to locate the error using other means such as line numbers.

Example

The following is an example of an exception report. This exception indicates that the first insert statement failed because of a missing closing XML tag in the request.

```
<?xml version="1.0" ?>
<ExceptionReport
  version="1.1.0"
  xmlns="http://www.opengis.net/ogc"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="owsExceptionReport.xsd">
  <Exception code="999" locator="INSERT STMT 01">
    <ExceptionText>parse error: missing closing tag for element
wkbGeom</ExceptionText>
  </Exception>
</ExceptionReport>
```

7.1 Introduction

This clause describes the request and responses for a basic OGC Publish/Subscribe service (OPS-B). The purpose of the service is to provide publish/subscribe services across OGC web services. Many of these services, such as the WFS, SOS or SPS, generate data in an asynchronous manner. Users may be interested in some of the data as it happens. The idea is to create a subscription mechanism to support the delivery of the data (or notifications/alerts) to many subscribed users using user profile information for the delivery mechanism (IM, email, feed, sms...).

7.2 Simplifying assumptions

7.2.1 Standard message output

The proposed message output format for an OPS-B service is Atom+GeoRSS 1.0. This format has been popularized by Google News and extended to include spatial information (www.georss.org).

Subscriptions need to define a filter (OGC filter). It is important that the filter applies in a similar manner to incoming data. Data must include meta-data tag information as well as geo-location and time information to be effectively filtered. <http://www.georss.org/>

7.2.2 User identity

The concept of user identity is becoming critical to the OGC web services for interoperability in a distributed environment. This service assumes that user identity and profile information is available to the service:

1. Two mechanisms for providing user identity and profile information where considered. OpenId 2.0 (<http://openid.net>)
2. SAML/XACML (http://www.oasisopen.org/committees/tc_home.php?wg_abbrev=xacml)

No consensus was achieved on this issue. However, the implementation of OPS-B and WFS-S created for OWS-4 made use of the OpenId 2.0 standard.

Readers are also encouraged to review the following work performed during the OWS-4 testbed:

1. [OWSGeoDRM](#)
2. [Ows4GeoDrmRequirements](#)
3. [Ows4GeoDrmUseCases](#)

4. [Ows4GeoDrmEngView](#)

5. [Ows4GeoDrmTrustIpr](#)

7.2.3 Requirements

The following are identified as required for the OPS-B web service:

1. Capability to aggregate data/feeds from all OGC services and notify users
2. Capability to offer user-based subscriptions to those feeds given some user defined criteria
3. Service must be compatible with many information delivery system including email, instant messaging, short message service ... allowing users to select how they want to receive the information.

7.3 OPS-B operations

7.3.1 Introduction

This clause describes the operations that shall be supported by an OPS-B service. All operations shall be implemented. Since this API is intended to be a simplification of the complete [OPS](#) only keyword-value pair (KVP) encoding of service operations is supported.

The basic information unit that the OPS-B service operates upon is a *subscription*. A subscription is a GeoRSS feed that is optionally filtered based on some user-defined spatial and/or non-spatial constraints.

The OPS-B service supports operations to create, and remove a subscription as well as operations so that clients may discover the list of available subscriptions, subscribe to a subscription, get the list of subscriptions to which the user subscribes, pause, resume or cancel a subscription.

7.3.2 CreateSubscription operation

7.3.2.1 Request encoding

The **CreateSubscription** operation allows a client to create a subscription by optionally specifying spatial and/or non-spatial constraints to be applied to an Atom+GeoRSS feed. The new subscription shall be given a name and may also include an optional description which is a human readable narrative describing the subscription.

Table 2 defines the parameters for the **CreateSubscription** operation.

Table 2 – CreateSubscription KVP request encoding

Parameter Name	Description	Value	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of	Y

OGC 07-004

Parameter Name	Description	Value	Mandatory
		<i>CreateSubscription.</i>	
NAME	Name of new subscription.		Y
BBOX	Bounding box indicating area of interest.	Comma separated list of decimal degree values. Specified as lower left, upper right. Fixed CRS of EPSG"4326.	N
FILTER	Regular expression used against GeoRSS content elements.		N
DESCRIPTION	Human readable description.		N

Example: The following is an example of a minimal **CreateSubscription** request.

<http://eo1.geobliki.com/ops?REQUEST=CreateSubscription&NAME=everything>

Example: The following is an example of creating a subscription within a specific area of interest.

[http://eo1.geobliki.com/ops?REQUEST=CreateSubscription&NAME="Essex,MD"&BBOX=76.4840,39.3092,-76.3771,39.2351](http://eo1.geobliki.com/ops?REQUEST=CreateSubscription&NAME=)

7.3.2.2 Response encoding

In response to a **CreateSubscription** request the service shall create an Atom+GeoRSS feed that satisfies the spatial (i.e. using the BBOX parameter) and/or non-spatial (i.e. using the FILTER parameter) constraints specified in the request.

In the response, the server shall include an id for the newly created subscription. The subscription id may be used in subsequent requests to the server to reference the subscription.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

Editor's Note: In general it is not sufficient to say that the response shall be Atom+GeoRSS. Atom+GeoRSS is such an open standard that anything can be put into the response. The specification should be more precise about the contents of each Atom+GeoRSS element that should appear in the response.

Example: The following is an example of the Atom+GeoRSS response document that an OPS-B service might generate in response to a **CreateSubscription** request:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:georss="http://www.georss.org/georss"
  xmlns:geonames="http://www.geonames.org"
```

```

xml:lang="en-US">
<title>EO1</title>
<subtitle type="html">Sensor Web Enabled (SWE) Data Node</subtitle>
<id>urn:uuid:3382EF0D-7D5E-43FB-B15C-B339DEFB4476</id>
<generator version="0.10Beta"
  uri="http://www.geobliki.org">GeoBliki</generator>
<link href="http://eol.geobliki.com/ops"
  rel="self"
  type="application/atom+xml"/>
<link href="http://eol.geobliki.com/"
  rel="alternate"
  type="text/html"/>
<updated>2006-12-28T18:51:29Z</updated>
<entry>
  <author>
    <name>linda</name>
    <email>linda@innovatesolutions.com</email>
  </author>
  <id>urn:uuid:A5175225-743E-45A0-BE10-518930A6F4EB</id>
  <published>2006-12-28T18:51:29Z</published>
  <updated>2006-12-28T18:51:29Z</updated>
  <title>Essex, MD</title>
  <link href="http://eol.geobliki.com/articles/guid/A5175225-743E-45A0-
BE10-518930A6F4EB"
    rel="self" type="text/html"/>
  <category term="ops"
    scheme="http://eol.geobliki.com/ops/articles/tag/ops"/>
  <content type="html">EO1 OPS Subscription - Created
    &lt;br/&gt; A new Subscription has been created SUBSCRIPTIONID=20
&lt;br/&gt;
    &lt;table&gt;
      &lt;tr&gt;
        &lt;td&gt;Description:&lt;/td&gt;
        &lt;td&gt;missing&lt;/td&gt;
      &lt;/tr&gt;
      &lt;tr&gt;
        &lt;td&gt;BBOX:&lt;/td&gt;
        &lt;td&gt;76.4840,39.3092,-
76.3771,39.2351&lt;/td&gt;
      &lt;/tr&gt;
      &lt;tr&gt;
        &lt;td&gt;Filter:&lt;/td&gt;
        &lt;td&gt;*&lt;/td&gt;
      &lt;/tr&gt;
    &lt;/table&gt;
    &lt;br/&gt;If you would like to subscribe to this subscription,
select the subscribe link(s) from the table below:&lt;br/&gt;
    &lt;table&gt;
      &lt;tr&gt;
        &lt;td&gt;Delivery&lt;/td&gt;
        &lt;td&gt;Content Type&lt;/td&gt;
      &lt;/tr&gt;
      &lt;tr&gt;
        &lt;td&gt;email&lt;/td&gt;
        &lt;td&gt;html&lt;/td&gt;
      &lt;/tr&gt;
    &lt;/table&gt;
    &lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&amp;SUBSCRIPTIONID=20&amp;D
ELIVERY=SMTP&amp;OUTPUTFORMAT='text/html'"&gt;Subscribe&lt;/a&gt;
      &lt;/td&gt;
    &lt;/tr&gt;
      &lt;tr&gt;
        &lt;td&gt;email&lt;/td&gt;
        &lt;td&gt;GeoRSS&lt;/td&gt;
      &lt;/tr&gt;

```


Table 3 – RemoveSubscription KVP request encoding

Parameter Name	Description	Value(s)	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of <i>RemoveSubscription</i> .	Y
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the GetSubscriptions operation (subclause 7.2.4).	Y
DELIVERY	Specifies delivery method to be removed.	One of: <i>SMTP, XMPP, HTTP</i> The default value is <i>SMTP</i> .	N
OUTPUTFORMAT	Specified output format to be removed.	One of: <i>text/xml;subtype=georss/1,</i> <i>text/html.</i> The default value is <i>text/html</i> .	N

Example: The following is an example of a minimal **RemoveSubscription** request.

<http://eo1.geobliki.com/ops?REQUEST=RemoveSubscription&SUBSCRIPTIONID=20>

7.3.3.2 Response encoding

In response to a **RemoveSubscription** request a service shall generate a plain text response (i.e. text/plain) with the single word SUCCESS.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

7.3.4 GetSubscriptions operation

7.3.4.1 Request encoding

The **GetSubscriptions** operation allows a client to discover the available subscriptions to which the client may subscribe using the **Subscribe** operation.

Table 4 defines the parameters for the **GetSubscriptions** operation.

Table 4 – GetSubscriptions KVP request encoding

Parameter Name	Description	Value	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of <i>GetSubscriptions</i> .	Y

Example: The following is an example of the **GetSubscriptions** request.

<http://eo1.geobliki.com/ops?REQUEST=GetSubscriptions>

OGC 07-004

7.3.4.2 Response encoding

In response to a **GetSubscriptions** request the service shall generate an Atom+GeoRSS document listing the available subscriptions and their ids.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

Editor's Note: See note in clause 7.2.3.2.

Example: The following XML fragment is a sample response to the **GetSubscriptions** operation.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:geonames="http://www.geonames.org"
  xmlns:georss="http://www.georss.org/georss"
  xml:lang="en-US">
  <title>EO1</title>
  <subtitle type="html">Sensor Web Enabled (SWE) Data Node</subtitle>
  <id>urn:uuid:21827143-58EA-472D-BCA1-FA505A0DD458</id>
  <generator version="0.10Beta"
    uri="http://www.geobliki.org">GeoBliki</generator>
  <link href="http://eol.geobliki.com/ops"
    rel="self" type="application/atom+xml"/>
  <link href="http://eol.geobliki.com/"
    rel="alternate" type="text/html"/>
  <updated>2006-12-28T18:51:29Z</updated>
  <entry>
    <author>
      <name>linda</name>
      <email>linda@innovatesolutions.com</email>
    </author>
    <id>urn:uuid:A5175225-743E-45A0-BE10-518930A6F4EB</id>
    <published>2006-12-28T18:51:29Z</published>
    <updated>2006-12-28T18:51:29Z</updated>
    <title>Essex, MD</title>
    <link href="http://eol.geobliki.com/articles/guid/A5175225-743E-45A0-
    BE10-518930A6F4EB"
      rel="self" type="text/html"/>
    <category term="ops"
      scheme="http://eol.geobliki.com/ops/articles/tag/ops"/>
    <content type="html">EO1 OPS Subscription - Created
      &lt;br/&gt;A new Subscription has been created SUBSCRIPTIONID=20
    &lt;br/&gt;
      &lt;table&gt;
        &lt;tr&gt;
          &lt;td&gt;Description:&lt;/td&gt;
          &lt;td&gt;missing&lt;/td&gt;
        &lt;/tr&gt;
        &lt;tr&gt;
          &lt;td&gt;BBOX:&lt;/td&gt;
          &lt;td&gt;-76.4840,39.3092,76.3771,39.2351&lt;/td&gt;
        &lt;/tr&gt;
        &lt;tr&gt;
          &lt;td&gt;Filter:&lt;/td&gt;
          &lt;td&gt;*&lt;/td&gt;
        &lt;/tr&gt;
      &lt;/table&gt;
    </content>
  </entry>
</feed>
```

```

        <br/>If you would like to subscribe to this subscription,
select the subscribe link(s) from the table below:<br/>
<table>
<tr>
<td>Delivery</td>
<td>Content Type</td>
</tr>
<tr>
<td>email</td>
<td>html</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&DELIVERY=SMTP&OUTPUTFORMAT='text/html'">Subscribe</a>
</td>
</tr>
<tr>
<td>email</td>
<td>GeorSS</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&DELIVERY=SMTP&OUTPUTFORMAT='text/xml;subtype=georss/1'">Subscribe</a>
</td>
</tr>
<tr>
<td>chat</td>
<td>html</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&DELIVERY=XMPP&OUTPUTFORMAT='text/html'">Subscribe</a>
</td>
</tr>
<tr>
<td>chat</td>
<td>GeorSS</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&DELIVERY=XMPP&OUTPUTFORMAT='text/xml;subtype=georss/1'">Subscribe</a>
</td>
</tr>
<tr>
<td>email</td>
<td>GeorSS</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&DELIVERY=HTTP&OUTPUTFORMAT='text/xml;subtype=georss/1'">Subscribe</a>
</td>
</tr>
</table>
</content>
<georss:where>
  <gml:Envelope>
    <gml:lowerCorner>39.2351 -76.484</gml:lowerCorner>
    <gml:upperCorner>39.3092 -76.3771</gml:upperCorner>
  </gml:Envelope>
</georss:where>
</entry>
</feed>

```

7.3.5Subscribe operation

7.3.5.1Request encoding

The **Subscribe** operation allows a client to subscribe to an available subscription. The list of available subscriptions can be discovered using the **GetSubscriptions** operation (subclause 7.2.3). The intended subscription is identified, in the request, using its id. The request optionally allows the client to specify a delivery method and an output format.

Table 5 defines the parameters for the **Subscribe** operation.

Table 5 – Subscribe KVP request encoding

Parameter Name	Description	Value(s)	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of <i>Subscribe</i> .	Y
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the GetSubscriptions operation (subclause 7.2.4).	Y
DELIVERY	Handler to be used to deliver subscription.	One of: <i>SMTP, XMPP, HTTP</i> The default value is <i>SMTP</i> .	N
OUTPUTFORMAT	The format of the response message.	One of: <i>text/xml;subtype=georss/1,</i> <i>text/html.</i> The default value is <i>text/html</i> .	N

Example: The following is an example of a minimal **Subscribe** request.

<http://eo1.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20>

7.3.5.2Response encoding

In response to a **Subscribe** request the service shall respond with an Atom+GeoRSS document listing the available subscriptions.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

Editor's Note: See note in clause 7.2.3.2.

Example: The following XML fragment is a sample response to the Subscribe operation.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:geonames="http://www.geonames.org"
  xmlns:georss="http://www.georss.org/georss">
```

```

xml:lang="en-US">
<title>E01</title>
<subtitle type="html">Sensor Web Enabled (SWE) Data Node </subtitle>
<id>urn:uuid:FED71050-8BFA-4C21-A1C4-BEC758BE0FD6</id>
<generator version="0.10Beta"
  uri="http://www.geobliki.org">GeoBliki</generator>
<link href="http://eol.geobliki.com/ops"
  rel="self" type="application/atom+xml"/>
<link href="http://eol.geobliki.com/"
  rel="alternate" type="text/html"/>
<updated>2006-12-28T18:51:29Z</updated>
<entry>
  <author>
    <name>linda</name>
    <email>linda@innovatesolutions.com</email>
  </author>
  <id>urn:uuid:A5175225-743E-45A0-BE10-518930A6F4EB</id>
  <published>2006-12-28T18:51:29Z</published>
  <updated>2006-12-28T18:51:29Z</updated>
  <title>Essex, MD</title>
  <link href="http://eol.geobliki.com/articles/guid/A5175225-
743E-45A0-BE10-518930A6F4EB"
    rel="self" type="text/html"/>
  <category term="ops"
    scheme="http://eol.geobliki.com/ops/articles/tag/ops"/
>
  <content type="html">E01 OPS Subscription - Created
    &lt;br/&gt; A new Subscription has been created SUBSCRIPTIONID=20
&lt;br/&gt;
    &lt;table&gt;
    &lt;tr&gt;
    &lt;td&gt;Description:&lt;/td&gt;
    &lt;td&gt;missing&lt;/td&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
    &lt;td&gt;BBOX:&lt;/td&gt;
    &lt;td&gt;-76.4840,39.3092,-76.3771,39.2351&lt;/td&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
    &lt;td&gt;Filter:&lt;/td&gt;
    &lt;td&gt;*&lt;/td&gt;
    &lt;/tr&gt;
    &lt;/table&gt;
    &lt;br/&gt;If you would like to subscribe to this subscription, select
the subscribe link(s) from the table below:&lt;br/&gt;
    &lt;table&gt;
    &lt;tr&gt;
    &lt;td&gt;Delivery&lt;/td&gt;
    &lt;td&gt;Content Type&lt;/td&gt;
    &lt;td/&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
    &lt;td&gt;email&lt;/td&gt;
    &lt;td&gt;html&lt;/td&gt;
    &lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=SMTP&OUTPUTFORMAT='text/html'"&gt;Subscribe
&lt;/a&gt;
    &lt;/td&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
    &lt;td&gt;email&lt;/td&gt;
    &lt;td&gt;GeoRSS&lt;/td&gt;

```

OGC 07-004

```
<td><a href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=SMTP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</table>
<td>chat</td>
<td>html</td>
</tr>
</table>
<a href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=XMPP&OUTPUTFORMAT='text/html'">Subscribe
</a>
</td>
</tr>
</table>
<td>chat</td>
<td>GeorSS</td>
</tr>
</table>
<a href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=XMPP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</table>
<td>email</td>
<td>GeorSS</td>
</tr>
</table>
<a href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=HTTP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</table>
</content>
<georss:where>
  <gml:Envelope>
    <gml:lowerCorner>39.2351 -76.484</gml:lowerCorner>
    <gml:upperCorner>39.3092 -76.3771</gml:upperCorner>
  </gml:Envelope>
</georss:where>
</entry>
</feed>
```

7.3.6 GetUserSubscriptions

7.3.6.1 Request encoding

The **GetUserSubscriptions** request generates a list of subscriptions to which a client has subscribed.

Table 6 defines the parameters for the **GetUserSubscriptions** operation.

Table 6 – GetUserSubscriptions KVP request encoding

Parameter Name	Description	Value(s)	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of <i>GetUserSubscriptions</i> .	Y

Example: The following is an example of a **GetUserSubscriptions** request.

<http://eol.geobliki.com/ops?REQUEST=GetUserSubscriptions>

7.3.6.2 Response encoding

The response to a **GetUsersSubscriptions** request is a Atom+GeoRSS document which contains one entry for each subscription to which the user has subscribed.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

Example: The following XML fragment is an example response to a **GetUserSubscriptions** request.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:geonames="http://www.geonames.org"
  xmlns:georss="http://www.georss.org/georss"
  xml:lang="en-US">
  <title>E01</title>
  <subtitle type="html">Sensor Web Enabled (SWE) Data Node</subtitle>
  <id>urn:uuid:CD5890C8-B9C0-41B9-9DD2-922384FA0796</id>
  <generator version="0.10Beta"
    uri="http://www.geobliki.org">GeoBliki</generator>
  <link href="http://eol.geobliki.com/ops"
    rel="self"
    type="application/atom+xml"/>
  <link href="http://eol.geobliki.com/"
    rel="alternate"
    type="text/html"/>
  <updated>2006-12-28T18:51:29Z</updated>
  <entry>
    <author>
      <name>linda</name>
      <email>linda@innovatesolutions.com</email>
    </author>
    <id>urn:uuid:A5175225-743E-45A0-BE10-518930A6F4EB</id>
    <published>2006-12-28T18:51:29Z</published>
    <updated>2006-12-28T18:51:29Z</updated>
    <title>Essex, MD</title>
    <link href="http://eol.geobliki.com/articles/guid/A5175225-743E-45A0-BE10-518930A6F4EB"
      rel="self"
      type="text/html"/>
    <category term="ops"
      scheme="http://eol.geobliki.com/ops/articles/tag/ops"/>
    <content type="html">E01 OPS Subscription - Created
      &lt;br/&gt; A new Subscription has been created SUBSCRIPTIONID=20
&lt;br/&gt;
      &lt;table&gt;
      &lt;tr&gt;
      &lt;td&gt;Description:&lt;/td&gt;
      &lt;td&gt;missing&lt;/td&gt;
```

OGC 07-004

```
</tr>
</tr>
<td>BBOX:</td>
<td>-76.4840,39.3092,76.3771,39.2351</td>
</tr>
</tr>
<td>Filter:</td>
<td>*</td>
</tr>
</table>
<br>If you would like to subscribe to this subscription, select
the subscribe link(s) from the table below:<br>
<table>
<tr>
<td>Delivery</td>
<td>ContentType</td>
</tr>
</tr>
<tr>
<td>email</td>
<td>html</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=SMTP&OUTPUTFORMAT='text/html'">Subscribe
</a>
</td>
</tr>
</tr>
<tr>
<td>email</td>
<td>GeoRSS</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=SMTP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</tr>
<tr>
<td>chat</td>
<td>html</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=XMPP&OUTPUTFORMAT='text/html'">Subscribe
</a>
</td>
</tr>
</tr>
<tr>
<td>chat</td>
<td>GeoRSS</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=XMPP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</tr>
<tr>
<td>email</td>
<td>GeoRSS</td>
<td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=20&DELIVERY=HTTP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</tr>
```

```

    </table>
</content>
<georss:where>
  <gml:Envelope>
    <gml:lowerCorner>39.2351 -76.484</gml:lowerCorner>
    <gml:upperCorner>39.3092 -76.3771</gml:upperCorner>
  </gml:Envelope>
</georss:where>
</entry>
<entry>
  <author>
    <name>linda</name>
    <email>linda@innovatesolutions.com</email>
  </author>
  <id>urn:uuid:218F9CFC-FC36-41CB-837B-299D049AAFED</id>
  <published>2006-12-28T19:30:40Z</published>
  <updated>2006-12-28T19:30:40Z</updated>
  <title>Everything</title>
  <link href="http://eol.geobliki.com/articles/guid/218F9CFC-FC36-41CB-837B-299D049AAFED" rel="self" type="text/html"/>
  <category term="ops"
    scheme="http://eol.geobliki.com/ops/articles/tag/ops"/>
  <content type="html">E01 OPS Subscription - Created
    <br/> A new Subscription has been created SUBSCRIPTIONID=21
<br/>
    <table>
      <tr>
        <td>Description:</td>
        <td>missing</td>
      </tr>
      <tr>
        <td>BBOX:</td>
        <td>-180.0, 90.0, 180.0, -90.0</td>
      </tr>
      <tr>
        <td>Filter:</td>
        <td>*</td>
      </tr>
    </table>
    <br/>If you would like to subscribe to this subscription, select
the subscribe link(s) from the table below:<br/>
    <table>
      <tr>
        <td>Delivery</td>
        <td>Content Type</td>
      </tr>
      <tr>
        <td>email</td>
        <td>html</td>
      </tr>
      <tr>
        <td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=21&DELIVERY=SMTP&OUTPUTFORMAT='text/html'">Subscribe
</a>
        </td>
      </tr>
      <tr>
        <td>email</td>
        <td>GeorSS</td>
      </tr>
      <tr>
        <td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=21&DELIVERY=SMTP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
        </td>
      </tr>
    </table>

```

OGC 07-004

```
<td><tr>
<td><td>chat</td>
<td><td>html</td>
<td><td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=21&DELIVERY=XMPP&OUTPUTFORMAT='text/html'">Subscribe
</a>
</td>
</tr>
<tr>
<td><td>chat</td>
<td><td>GeoRSS</td>
<td><td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=21&DELIVERY=XMPP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
<tr>
<td><td>email</td>
<td><td>GeoRSS</td>
<td><td><a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTI
ONID=21&DELIVERY=HTTP&OUTPUTFORMAT='text/xml;subtype=geor
ss/1'">Subscribe</a>
</td>
</tr>
</table>
</content>
<georss:where>
<gml:Envelope>
<gml:lowerCorner>-180.0000 90.0000</gml:lowerCorner>
<gml:upperCorner>180.0000 -90.0000</gml:upperCorner>
</gml:Envelope>
</georss:where>
</entry>
</feed>
```

7.3.7PauseSubscription operation

7.3.7.1Request encoding

The **PauseSubscription** operation can be used to temporary suspend delivery of the specified subscription. The current list of user subscriptions can be discovered using the **GetUserSubscriptions** operation (subclause 7.2.5). The subscription to be pause is identified, in the request, using its id.

The request optionally allows the client to specify which combination of subscription delivery method and output format should be paused.

Table 7 defines the parameters for the **PauseSubscription** operation.

Table 7 – PauseSubscription KVP request encoding

Parameter Name	Description	Value(s)	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of <i>PauseSubscription</i> .	Y

Parameter Name	Description	Value(s)	Mandatory
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the GetSubscriptions operation (subclause 7.2.4).	Y
DELIVERY	Which deliver method should be paused.	One of: <i>SMTP, XMPP, HTTP</i> The default value is <i>SMTP</i> .	N
OUTPUTFORMAT	Which message output format should be paused.	One of: <i>text/xml;subtype=georss/1,</i> <i>text/html.</i> The default value is <i>text/html</i> .	N

Example: The following example pauses the the text/html format of a subscription delivered using SMTP.

<http://eol.geobliki.com/ops?REQUEST=PauseSubscription&SUBSCRIPTIONID=20>

7.3.7.2 Response encoding

In response to a **PauseSubscription** operation a service shall generate a GeoRSS document that contains ... T.B.D.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

Example: The following XML fragment is a sample response to the **PauseSubscription** request.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed
  xmlns="http://www.w3.org/2005/Atom"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
  xmlns:geonames="http://www.geonames.org"
  xmlns:georss="http://www.georss.org/georss"
  xml:lang="en-US">
  <title>E01</title>
  <subtitle type="html">Sensor Web Enabled (SWE) Data Node </subtitle>
  <id>urn:uuid:69893C02-5454-4DEA-BC24-362141FA15B5</id>
  <generator version="0.10Beta"
    uri="http://www.geobliki.org">GeoBliki</generator>
  <link href="http://eol.geobliki.com/ops"
    rel="self" type="application/atom+xml"/>
  <link href="http://eol.geobliki.com/"
    rel="alternate" type="text/html"/>
  <updated>2006-12-28T18:51:29Z</updated>
  <entry>
    <author>
      <name>linda</name>
      <email>linda@innovatesolutions.com</email>
    </author>
    <id>urn:uuid:A5175225-743E-45A0-BE10-518930A6F4EB</id>
    <published>2006-12-28T18:51:29Z</published>
    <updated>2006-12-28T18:51:29Z</updated>
  </entry>
</feed>
```

OGC 07-004

```
<title>Essex, MD</title>
<link href="http://eol.geobliki.com/articles/guid/A5175225-743E-45A0-
BE10-518930A6F4EB"
  rel="self" type="text/html"/>
<category term="ops"
  scheme="http://eol.geobliki.com/ops/articles/tag/ops"/>
<content type="html">E01 OPS Subscription - Created
  &lt;br/&gt; A new Subscription has been created SUBSCRIPTIONID=20
&lt;br/&gt;
  &lt;table&gt;
  &lt;tr&gt;
  &lt;td&gt;Description:&lt;/td&gt;
  &lt;td&gt;missing&lt;/td&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
  &lt;td&gt;BBOX:&lt;/td&gt;
  &lt;td&gt;-76.4840,39.3092,76.3771,39.2351&lt;/td&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
  &lt;td&gt;Filter:&lt;/td&gt;
  &lt;td&gt;*&lt;/td&gt;
  &lt;/tr&gt;
  &lt;/table&gt;
  &lt;br/&gt;If you would like to subscribe to this subscription, select
the subscribe link(s) from the table below:&lt;br/&gt;
  &lt;table&gt;
  &lt;tr&gt;
  &lt;td&gt;Delivery&lt;/td&gt;
  &lt;td&gt;Content Type&lt;/td&gt;
  &lt;td/&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
  &lt;td&gt;email&lt;/td&gt;
  &lt;td&gt;html&lt;/td&gt;
  &lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&amp;SUBSCRIPTI
ONID=20&amp;DELIVERY=SMTP&amp;OUTPUTFORMAT='text/html'"&gt;Subscribe
&lt;/a&gt;&lt;/td&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
  &lt;td&gt;email&lt;/td&gt;
  &lt;td&gt;GeoRSS&lt;/td&gt;
  &lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&amp;SUBSCRIPTI
ONID=20&amp;DELIVERY=SMTP&amp;OUTPUTFORMAT='text/xml;subtype=geor
ss/1'"&gt;Subscribe&lt;/a&gt;
  &lt;/td&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
  &lt;td&gt;chat&lt;/td&gt;
  &lt;td&gt;html&lt;/td&gt;
  &lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&amp;SUBSCRIPTI
ONID=20&amp;DELIVERY=XMPP&amp;OUTPUTFORMAT='text/html'"&gt;Subscribe
&lt;/a&gt;&lt;/td&gt;
  &lt;/tr&gt;
  &lt;tr&gt;
  &lt;td&gt;chat&lt;/td&gt;
  &lt;td&gt;GeoRSS&lt;/td&gt;
  &lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&amp;SUBSCRIPTI
ONID=20&amp;DELIVERY=XMPP&amp;OUTPUTFORMAT='text/xml;subtype=geor
ss/1'"&gt;Subscribe&lt;/a&gt;
  &lt;/td&gt;
  &lt;/tr&gt;
  &lt;/table&gt;
```

```

        </tr>
        <tr>
        <td>email</td>
        <td>GeorSS</td>
        </tr>
        </table>
        </td>
        </tr>
        </table>
        </content>
        <georss:where>
        <gml:Envelope>
        <gml:lowerCorner>39.2351 -76.484</gml:lowerCorner>
        <gml:upperCorner>39.3092 -76.3771</gml:upperCorner>
        </gml:Envelope>
        </georss:where>
        </entry>
    </feed>

```

7.3.8 ResumeSubscription operation

7.3.8.1 Request encoding

The **ResumeSubscription** operation can be used to resume delivery of a previously paused subscription. The request optionally allows the client to specify which combination of subscription delivery method and output format should be resumed.

Table 8 defines the parameters for the **ResumeSubscription** operation.

Table 8 – ResumeSubscription KVP request encoding

Parameter Name	Description	Value(s)	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of <i>ResumeSubscription</i> .	Y
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the GetSubscriptions operation (subclause 7.2.4).	Y
DELIVERY	Which deliver method should be resumed.	One of: <i>SMTP, XMPP, HTTP</i> The default value is <i>SMTP</i> .	N
OUTPUTFORMAT	Which message output format should be resumed.	One of: <i>text/xml;subtype=georss/1,</i> <i>text/html.</i> The default value is <i>text/html</i> .	N

Example: The following example pauses the the text/html format of a subscription delivered using SMTP.

```

http://eol.geobliki.com/ops?REQUEST=ResumeSubscription&SUBSCRIPTIONID=20

```

7.3.8.2 Response encoding

In response to a **ResumeSubscription** request, the service shall generate a Atom+GeoRSS document containing T.B.D.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

Example: Sample response.

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns:gml="http://www.opengis.net/gml"
xmlns:geo="http://www.w3.org/2003/01/geo/wgs84_pos#"
xmlns:geonames="http://www.geonames.org" xml:lang="en-US"
xmlns:georss="http://www.georss.org/georss" xmlns="http://www.w3.org/2005/Atom">
  <title>EO1</title>
  <subtitle type="html">Sensor Web Enabled (SWE) Data Node </subtitle>
  <id>urn:uuid:841DFBFC-D27B-41C3-ABAF-6B9E3EBE6EEB</id>
  <generator version="0.10Beta"
uri="http://www.geobliki.org">GeoBliki</generator>
  <link href="http://eol.geobliki.com/ops" rel="self"
type="application/atom+xml"/>
  <link href="http://eol.geobliki.com/" rel="alternate" type="text/html"/>

  <updated>2006-12-28T18:51:29Z</updated>
  <entry>
    <author>
      <name>linda</name>
      <email>linda@innovatesolutions.com</email>
    </author>
    <id>urn:uuid:A5175225-743E-45A0-BE10-518930A6F4EB</id>

    <published>2006-12-28T18:51:29Z</published>
    <updated>2006-12-28T18:51:29Z</updated>
    <title>Essex, MD</title>
    <link href="http://eol.geobliki.com/articles/guid/A5175225-743E-45A0-BE10-
518930A6F4EB" rel="self" type="text/html"/>
    <category term="ops" scheme="http://eol.geobliki.com/ops/articles/tag/ops"/>
```

```
<content type="html">E01 OPS Subscription - Created &lt;br/&gt; A new
Subscription has been created SUBSCRIPTIONID=20 &lt;br/&gt;
&lt;table&gt;&lt;tr&gt;&lt;td&gt;Description:&lt;/td&gt;&lt;td&gt;missing&lt;/t
d&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;BBOX:&lt;/td&gt;&lt;td&gt;-
76.4840,39.3092,-
76.3771,39.2351&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;Filter:&lt;/td&gt;&lt;
td&gt;*&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;&lt;br/&gt;If you would like to
subscribe to this subscription, select the subscribe link(s) from the table
below:&lt;br/&gt;&lt;table&gt;&lt;tr&gt;&lt;td&gt;Delivery&lt;/td&gt;&lt;td&gt;
Content
Type&lt;/td&gt;&lt;td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;email&lt;/td&gt;&lt;td
&gt;html&lt;/td&gt;&lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&D
ELIVERY=SMTP&OUTPUTFORMAT='text/html'"&gt;Subscribe&lt;/a&gt;&lt;/td&gt;&lt;
tr&gt;&lt;tr&gt;&lt;td&gt;email&lt;/td&gt;&lt;td&gt;GeoRSS&lt;/td&gt;&lt;td&gt;
t&lt;/td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&D
ELIVERY=SMTP&OUTPUTFORMAT='text/xml;subtype=georss/1'"&gt;Subscribe&lt;/a&gt;
t&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;chat&lt;/td&gt;&lt;td&gt;html&lt;/t
d&gt;&lt;td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&D
ELIVERY=XMPP&OUTPUTFORMAT='text/html'"&gt;Subscribe&lt;/a&gt;&lt;/td&gt;&lt;
tr&gt;&lt;tr&gt;&lt;td&gt;chat&lt;/td&gt;&lt;td&gt;GeoRSS&lt;/td&gt;&lt;td&gt;
&lt;/td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&D
ELIVERY=XMPP&OUTPUTFORMAT='text/xml;subtype=georss/1'"&gt;Subscribe&lt;/a&gt;
t&lt;/td&gt;&lt;/tr&gt;&lt;tr&gt;&lt;td&gt;email&lt;/td&gt;&lt;td&gt;GeoRSS&lt;
td&gt;&lt;/td&gt;&lt;a
href="http://eol.geobliki.com/ops?REQUEST=Subscribe&SUBSCRIPTIONID=20&D
ELIVERY=HTTP&OUTPUTFORMAT='text/xml;subtype=georss/1'"&gt;Subscribe&lt;/a&gt;
t&lt;/td&gt;&lt;/tr&gt;&lt;/table&gt;</content>
<georss:where>
  <gml:Envelope>
    <gml:lowerCorner>39.2351 -76.484</gml:lowerCorner>
    <gml:upperCorner>39.3092 -76.3771</gml:upperCorner>
  </gml:Envelope>
</georss:where>
</entry>
</feed>
```

7.3.9 Cancel Subscription operation

7.3.9.1 Request encoding

The **CancelSubscription** operation can be used to permanently suspend delivery of the specified subscription.

Since a client can subscribe to a feed multiple times using a different combination of deliver method and output format each time, the request optionally allows the client to specify which combination of subscription delivery method and output format shall be cancelled. By default the operation will cancel the feed with the output format *text/html* delivered using *SMTP*.

Table 9 defines the parameters for the **CancelSubscription** operation.

Table 9 – CancelSubscription KVP request encoding

Parameter Name	Description	Value(s)	Mandatory
REQUEST	Name of OPS-B operation.	Fixed value of	Y

OGC 07-004

Parameter Name	Description	Value(s)	Mandatory
		<i>CancelSubscription.</i>	
SUBSCRIPTIONID	The ID of an existing subscription	Possible values obtained using the GetSubscriptions operation (subclause 7.2.4).	Y
DELIVERY	Which deliver method should be cancelled.	One of: <i>SMTP, XMPP, HTTP</i> The default value is <i>SMTP</i> .	N
OUTPUTFORMAT	Which message output format should be cancelled.	One of: <i>text/xml;subtype=georss/1,</i> <i>text/html.</i> The default value is <i>text/html.</i>	N

Example: The following example cancels the the *text/html* format of a subscription delivered using *SMTP* with the id of 20.

<http://eo1.geobliki.com/ops?REQUEST=CancelSubscription&SUBSCRIPTIONID=20>

7.3.10 Response encoding

In response to a **CancelSubscription** request a service shall generate a plain text response (i.e. text/plain) with the single word SUCCESS.

In the event that an exception is encountered, the service shall respond with an exception report as specified in clause 6.2.5.

8 Web feature service - simple

8.1 Introduction

This clause defines the request and response message encoding for a simple Web Feature Service (WFS-S). WFS-S is an attempt to define a WFS conformance class that is simpler than the basic conformance class defined in 04-094 and has more mass-market appeal.

Both WFS-S and the basic WFS conformance class define the same set of operations: GetCapabilities, DescribeFeatureType and GetFeature. However, WFS-S makes a number of simplifying assumptions.

8.2 Simplifying assumptions

The primary differences between the OpenGIS Web Feature Service Implementation Specification and WFS-S are:

- a) GML is not a required output format

b) There is no XML encoding for request messages. Only keyword-value pair encoding (KVP) is supported.

c) Only one feature type is allowed per service instance (therefore eliminating the need for the typeName parameter).

d) Non-spatial predicates are encoded using regular expressions (regex) rather than using Filter (04-095).

8.3 Common service elements

8.4 Common request parameters

All WFS-S operations, with the exception of the **GetCapabilities** request, shall support the parameters described in Table 10.

Table 10 – Common request parameter

Parameter Name	Description	Value	Mandatory?	In WFS 1.1
SERVICE	Name of the service.	Fixed value of WFSS	Y	Y
VERSION	Interface version.	0.6.0	Y	Y

8.4.1.1 Service parameter

The **SERVICE** parameter is used to indicate which service shall be invoked at a particular endpoint. The **SERVICE** parameter shall be specified with each WFS-S request. Its value shall be WFS-S.

8.4.1.2 Version parameter

The **VERSION** parameter is used to indicate which version of the WFS-S API should be invoked. For this version of the specification, the value of the **VERSION** parameter shall be 0.6.0.

How changes are made to the version number and version number negotiation are described in sub-clause 7.3 of OGC 05-008c1.

8.5 GetCapabilities operation (mandatory)

8.5.1 Introduction

A WFS-S shall have the ability to describe its capabilities by returning service metadata, in the form of a XML document, in response to a **GetCapabilities** request. Specifically, every simple web feature service shall support the KVP encoded form of the **GetCapabilities** request over HTTP GET so that a client may always know how to obtain a capabilities document.

OGC 07-004

This section defines the KVP encoding for a **GetCapabilities** request and defines the service metadata generated in response as an XML document that a simple web feature service shall generate to describe its capabilities.

8.5.2 Request encoding

The normative request encoding for the GetCapabilities request is define in Table 1 of OGC 05-008c1. Table 1 from OGC 05-008c1 is reproduced here, as Table 11 for the sake of convenience.

Table 11 defines the encoding for a **GetCapabilities** request.

Table 11 – GetCapabilities request encoding

Parameter Name	Description	Value	Mandatory?	In WFS 1.1
SERVICE	Service type identifier	WFSS	Y	Y
REQUEST	Name of the operation to invoke.	Fixed value of GetCapabilities	Y	Y
ACCEPTVERSIONS	Prioritized sequence of one or more specification versions accepted by client, with preferred version listed first.		N	Y
SECTIONS	Unordered list of zero or more names of requested sections in complete service metadata document		N	Y
UPDATESEQUENCE	Service metadata document version, value is “increased” whenever any change is made in complete service metadata document		N	Y
ACCEPTFORMATS	Prioritized sequence of zero or more response formats desired by client, with preferred formats listed first.		N	Y

A complete description of the **GetCapabilities** request can be found in clause 7 of 05-008c1.

8.5.3 Response encoding

The following XML Schema fragment defines the response to a GetCapabilities request:

```
<xs:element name="WFS_Simple_Capabilities">
  <xs:complexType>
    <xs:complexContent>
      <xs:extension base="ows:CapabilitiesBaseType"/>
    </xs:complexContent>
  </xs:complexType>
</xs:element>
```

This conforms to the OGC common services specification for describing the capabilities of services (sub-clause 7.4 of OGC 05-008c1). It differs from WFS 1.1 in that no FeatureTypeList or FilterCapabilities section is specified.

8.5.4 Example

The following XML document is an example of a response to the GetCapabilities request:

<http://example.com/wfss?service=WFSS&version=0.6.0&REQUEST=GetCapabilities>

response:

```
<?xml version="1.0" encoding="UTF-8"?>
<wfss:WFS_Simple_Capabilities
  version="1.0.0"
  updateSequence="0"
  xmlns="http://www.opengis.net/ows"
  xmlns:wfss="http://www.opengis.net/wfss"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xsi:schemaLocation="http://www.opengis.net/wfss
http://schemas.opengis.net/wfss/0.0.6/wfss.xsd">

  <ServiceIdentification>
    <Title>USGS KY climate data generator</Title>
    <Abstract>WFS Simple capabilities document for USGS KY climate data
generator</Abstract>
    <Keywords>
      <Keyword>air</Keyword>
      <Keyword>water</Keyword>
      <Keyword>temperature</Keyword>
    </Keywords>
    <ServiceType>WFSS</ServiceType>
    <ServiceTypeVersion>0.0.6</ServiceTypeVersion>
    <Fees>none</Fees>
    <AccessConstraints>none</AccessConstraints>
  </ServiceIdentification>

  <ServiceProvider>
    <ProviderName>Acme, Inc.</ProviderName>
    <ServiceContact>
      <PositionName>Geo Sys Admin</PositionName>
      <ContactInfo>
        <Phone>
          <Voice>1.800.CALL.US</Voice>
        </Phone>
        <Address>
          <ElectronicMailAddress>geoadmin@example.com</ElectronicMailAddr
ess>
        </Address>
      </ContactInfo>
    </ServiceContact>
  </ServiceProvider>

  <OperationsMetadata>
    <Operation name="GetCapabilities">
      <DCP>
        <HTTP>
          <Get xlink:href="http://www.example.com/wfs"/>
        </HTTP>
      </DCP>
    </Operation>
  </OperationsMetadata>
</wfss:WFS_Simple_Capabilities>
```

OGC 07-004

```
</DCP>
</Operation>
<Operation name="DescribeFeatureType">
  <DCP>
    <HTTP>
      <Get xlink:href="http://www.example.com/wfs"/>
    </HTTP>
  </DCP>
  <Parameter name="outputFormat">
    <Value>text/xml; subtype=bxfs/0.0.3</Value>
    <Value>text/plain</Value>
  </Parameter>
</Operation>
<Operation name="GetFeature">
  <DCP>
    <HTTP>
      <Get xlink:href="http://www.example.com/wfs"/>
    </HTTP>
  </DCP>
  <Parameter name="outputFormat">
    <Value>application/bxfs+xml subtype=bxfs/0.0.3</Value>
    <Value>text/xml; subtype=atom+georss/2.0</Value>
  </Parameter>
</Operation>
</OperationsMetadata>
</wfs:WFS_Simple_Capabilities>
```

This sample document only lists a subset of the information a service provider can put in the `ServiceIdentification` and `ServiceProvider` sections of the capabilities document. The full set of elements is specified in sub-clause 7.4 of OGC 05-008c1.

8.6 DescribeFeatureType operation (optional)

8.6.1 Introduction

WFS Simple assumes that the primary way of understanding how to use the data will involve making sample requests and examining the responses or having a priori knowledge about the schema of the data being served. Based upon these assumptions, the optional **DescribeFeatureType** operation does not need to normatively describe the schema of the data being served.

Instead, we use **DescribeFeatureType** for three things:

1. specifying which variables can be queried (queryable attributes)
2. adding metadata about variables that can aid client developers to build better user interfaces (e.g. title)
3. adding metadata about variables that can aid data harvesting use cases (e.g. data type and other facets)

8.6.2 Request encoding

Table 12 defines the encoding for a **DescribeFeatureType** request.

Table 12 – DescribeFeatureType request encoding

Parameter Name	Description	Value	Mandatory	In WFS 1.1
REQUEST	Name of the operation to invoke.	Fixed value of DescribeFeatureType	Y	Y
OUTPUTFORMAT	The format of the response.	One of the values listed in the capabilities document, using the Parameter element, for the DescribeFeatureType operation.	N Default is: application/bxfs+xml; subtype=bxfs/0.0.3	Y

8.6.3 Response encoding

8.6.3.1 Introduction

The response to a **DescribeFeatureType** request shall describe the data's layout. As is the case with WFS, the format for doing this depends upon what is advertised in the service's capabilities, but whatever system is used, it should have three components:

- 1.a narrative description of the data that repeats and/or more fully explains the data's layout
- 2.a listing of properties (attributes) that can be queried
- 3.a more formal data description, preferably those offered by schema description languages like XML Schema, RELAX NG and DTD

None of these components are strictly required by WFS-S. They add useful information for clients to construct queries, but the basic functionality of WFS-S may still be fulfilled without a **DescribeFeatureType** operation.

This specification describes three preferred formats for describing the structure of the data.

- 1.XML (in the form of an XML document using BXFS elements)
- 2.Plain text
- 3.[Simple feature GML](#)

8.6.3.2 BXFS data description

The Basic XML Feature Schema (BXFS) is a fixed XML schema that may be used to encode feature data. BXFS creates a comprehensive baseline that all WFS Simple services and clients should be able to understand, and therefore promotes the highest level of interoperability across all WFS-S services. It is described in Annex B and includes a component for describing the properties of a feature type. This component may be used to create a response document to a **DescribeFeatureType** request.

OGC 07-004

WFS-S uses BXFS elements to define a response to the DescribeFeatureType request that fulfils the three goals listed in sub-clause 8.6.3.1. It deviates slightly from XML's data typing scheme, by defining a list of acceptable data types harmonized from XML, SQL, and GML.

The following XML schema fragment defines the response to a **DescribeFeatureType** request and makes use of BXFS elements:

```
<xs:element name="FeatureDescription">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="bxfs:Properties" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name"
      type="xs:string" use="optional"/>
    <!-- spec version number -->
    <xs:attribute name="version" type="xs:string"
      use="required" fixed="0.6"/>
  </xs:complexType>
</xs:element>
```

The BXFS element **bxfs:Properties** contains a list of property definitions encoded using the following XML Schema fragment:

```
<xs:element name="Property" type="wfss:PropertyType"
  substitutionGroup="bxfs:AbstractProperty"/>
<xs:complexType name="PropertyType">
  <xs:complexContent>
    <xs:extension base="bxfs:AbstractPropertyType">
      <xs:sequence>
        <xs:element ref="wfss:Property"
          minOccurs="0" maxOccurs="unbounded"/>
      </xs:sequence>
      <xs:attribute name="name"
        type="xs:string" use="required"/>
      <xs:attribute name="title"
        type="xs:string" use="optional"/>
      <xs:attribute name="queryable"
        type="xs:boolean" use="optional"/>
      <xs:attribute name="type"
        use="optional" default="xs:string">
        <xs:simpleType>
          <xs:restriction base="xs:QName">
            <xs:enumeration value="xs:string"/>
            <xs:enumeration value="xs:integer"/>
            <xs:enumeration value="xs:double"/>
            <xs:enumeration value="xs:boolean"/>
            <xs:enumeration value="xs:dateTime"/>
            <xs:enumeration value="xs:anyURI"/>
            <xs:enumeration value="xs:base64Binary"/>
            <xs:enumeration value="gml:Point"/>
            <xs:enumeration value="gml:Line"/>
            <xs:enumeration value="gml:Polygon"/>
            <xs:enumeration value="gml:Envelope"/>
          </xs:restriction>
        </xs:simpleType>
      </xs:attribute>
      <xs:attribute name="maxLength"
        type="xs:nonNegativeInteger"/>
      <xs:attribute name="precision"
        type="xs:positiveInteger"/>
      <xs:attribute name="scale" type="xs:integer"/>
      <xs:attribute name="valueDomain" type="xs:anyURI"/>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

```

    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

The **name** attribute shall be used to specify the name of property. This is the name that may be used in as a parameter in the **GetFeature** request to specify a query predicate.

The **title** attribute may be used to specify a human-readable title. Such a title might be used to label the property in a GUI, for example.

The **queryable** attribute is used to indicate whether the property may be used as a parameter in the **GetFeature** request to specify a constraint on the property.

The **type** attribute is used to specify the data type of the property. This shall only be specified for simple scalar properties. Aggregate or complex properties shall not specify the **type** attribute.

The **maxLength**, **precision** and **scale** attributes may be specified as modifiers for the **type** attribute. For example, if the **type** attribute is set to *xs:integer*, then the **precision** attribute may be used to specify the maximum number of digits that the integer may have. Table 13 specifies the correlation between the **type** of the property and the attributes **maxLength**, **precision** and **scale**.

Table 13 – Correlating datatype with maxLength, precision and scale attributes

Type value	maxLength	Precision	scale
string	Y	N	N
integer	N	Y	N
double	N	Y	Y
noolean	N	N	N
dateTime	N	N	N
anyURI	N	N	N
base64Binary	Y	N	N
Point	N	N	N
Line	N	N	N
Polygon	N	N	N
Envelope	N	N	N

The **valueDomain** attribute may be used to point to an authoritative list-of-values for the property. No specific format for the list-of-values is mandated by this specification, however the use of GML dictionaries is strongly recommended.

OGC 07-004

Servers that support this response to a **DescribeFeatureType** operation shall advertise this fact in their capability document, using the **Parameter** element within the **Operation** element, with the value *application/bxfs+xml; subtype=bxfs/0.0.3*.

8.6.3.3 Example

Consider the following database schema:

Field	Data type
name	string
income	integer
lat	Double
lon	double

For this schema, in response to a **DescribeFeatureType** request, where the value of the **outputFormat** parameter is set to *application/bxfs+xml; subtype=bxfs/0.0.3*, a WFS-S shall generate the following XML document:

```
<FeatureDescription
  xmlns="http://www.opengis.net/wfss"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfss
http://www.ogcnetwork.net/schemas/wfssimple/0.6/wfss.xsd"
  xmlns:gml="http://www.opengis.net/gml">
  <Description>
    <Title>My data service</Title>
    <Abstract>Income for Springfield residents</Abstract>
    <MetadataURL
      url="http://example.com/simpsons/secretdatabase/about.html"/>
  </Description>
  <gml:Envelope srsName="urn:x-ogc:def:crs:EPSG:6.3:4326">
    <gml:lowerCorner>43.2 -79.3</gml:lowerCorner>
    <gml:upperCorner>43.6 -79.4</gml:upperCorner>
  </gml:Envelope>
  <Properties>
    <Property name="name" title="Full Name"
      type="string" queryable="true" />
    <Property name="income" title="Income, 2006"
      type="integer" queryable="true" />
    <Property name="location" type="gml:Point"
      title="Location of primary residence"/>
  </Properties>
</FeatureDescription>
```

8.6.3.4 Plain text data description

It is important to note that a perfectly valid way of describing the information model is in a narrative format. This is particularly useful for legacy data formats like ASCII that may not look anything like a relational database or XML schema. The response would be a simple paragraph describing the data layout. This example illustrates the flexibility of the service API, but note that **this method is highly discouraged, as it hinders data interoperability and fits poorly into any concept of information flows in the World Wide Web.**

Servers that support a plain text response to the **DescribeFeatureType** operation shall advertise this fact, using the Parameter element within the Operation element, with the value *text/plain* optionally containing a *subtype* parameter to further qualify the format of the text response (e.g. *text/plain; subtype=rtf*).

8.6.3.5 Geography markup language (GML)

GML with XML Schema is, of course, a preferred data description format but that option is well described in the WFS specification (OGC 04-094). Of particular interest are levels 0 and 1 of the Simple Feature Profile of GML (OGC 06-049r1) which is well suited to the requirements of WFS-S.

8.7 GetFeature operation (mandatory)

8.7.1 Introduction

The primary operation of a WFS-S, and the means by which information is retrieved from the service, is the **GetFeature** operation.

8.7.2 Request encoding

Table 14 defined the encoding of the **GetFeature** operation.

Table 14 – GetFeature request encoding

Parameter Name	Description	Value	Mandatory	In WFS 1.1
REQUEST	Name of the operation to invoke.	Fixed value of GetFeature	Y	Y
BBOX	Constrain the results to features whose geometry spatially interacts with the specified BBOX.	Comma-separated list of real value defining the lower corner and upper corner of the BBOX. Optionally, the comma-separated list may include a crs URI as per subclause 10.2.3 of 05-008c1.	N	Y
TIME	Constraint the results to features whose timestamp lies within the specified time period.	An ISO8061 time period.	N	N
OUTPUTFORMAT	Encoding format of the response.	A string token identifying the response format (e.g. a MIME type such as <i>application/bxfs+xml; subtype=bxfs/0.0.3</i>)	N	Y
MAXFEATURES	The maximum number of features that the WFS-S shall present in the response ... even if more features are available in	A positive, non-zero integer.	N	Y

Parameter Name	Description	Value	Mandatory	In WFS 1.1
	the result set.			
SRSNAME	a WFS-supported SRS that should be used for returned feature geometries.	URN	N Default value is urn:x-ogc:def:crs:EPSG:6.3:4326	Y
PROPERTYNAME	A comma-separated list of property names (i.e. data fields)		N	Y
<i>PropertyNameQuery</i>	The name of a feature property that shall be constrained using a regular expression (i.e. regex)	String representing a regular expression (regex).	N	N

8.7.2.1 Parameter discussion

BBOX

The bounding box parameter, **BBOX**, is used to constraint the result set to those features that spatially interact with the specified rectangular geographic extent. The term “spatially interacts” means that any feature who's geometry touches, crosses the boundary is inside, overlaps, covers, is covered or is within a certain distance of the specified **BBOX** shall be present in the response of the **GetFeature** operation.

The format of the value of this parameter is specified in subclause 10.2.3 of OGC 05-008c1. The general form is:

BBOX=*lcc1,lcc2,ucc1,ucc2* [*crsURI*]

where *lcc* means Lower Corner Coordinate, *ucc* means Upper Corner Coordinate and *crsURI* means the URI reference to the coordinate system being used. If the *crsURI* is not specified then the 2D coordinates shall be specified in decimal degrees and WGS84 as described in 05-008c1.

If the **BBOX** parameter is not specified in the request then this shall be interpreted to mean that the extent of the query is the entire world.

TIME

The **TIME** parameter is used to constrain the result set to those features whose timestamp falls within the specified time period. The value of the **TIME** parameter is specified using the format *start_time/end_time*, where each time value is specified using the format *yyyy-mm-ddThh:mm:ss.sTZD*. The meaning of each component is the time value's format mask is:

YYYY = four-digit year
 MM = two-digit month (01=January, etc.)
 DD = two-digit day of month (01 through 31)
 hh = two digits of hour (00 through 23) (am/pm NOT allowed)
 mm = two digits of minute (00 through 59)
 ss = two digits of second (00 through 59)
 s = digit(s) representing a decimal fraction of a second
 TZD = time zone designator (Z or +hhmm or -hhmm)

Either the start value or the end value can be omitted to indicate no restriction on time in that direction.

example of start time only: 2006-10-23T04:05:06 -050

example of end time only: /2006-10-25T04:05:06 -0500

If no timezone is given, it shall be assumed that times are in UTC (normally designated with the 'Z').

If the concept of time does not apply to the data being served, and the **TIME** parameter is specified in a **GetFeature** request, the parameter shall be ignored.

If the **TIME** parameter is not specified in the request then this shall be interpreted to mean that no time limit is placed on the request.

OUTPUTFORMAT

The **OUTPUTFORMAT** parameter specifies the format of the response to a **GetFeature** request. Its value is typically a MIME type indicating the format of the output but any token may be used as long as the values supported by a specific service instance are advertised in the service's capability document. The following XML fragment shows how a WFS-S shall advertise the supported output format values:

```

<Operation name="GetFeature">
  <DCP>
    <HTTP>
      <Get xlink:href="http://www.example.com/wfs"/>
    </HTTP>
  </DCP>
  <Parameter name="outputFormat">
    <Value>application/bxfs+xml; subtype=bxfs/0.0.3</Value>
    <Value>application/atom+xml; subtype=atom+georss/2.0</Value>
    <Value>text/plain</Value>
    <Value>text/xml; gml/2.1.2</Value>
    <Value>text/xml; subtype=gml/3.1.1/profiles/gmlsf/1.0.0/0</Value>
  </Parameter>
</Operation>

```

In this example the server supports BXFS, GML 2.1.2, Atom+GeoRSS and plain text responses.

The following table lists standard **OUTPUTFORMAT** values to be used if the server supports the listed output format.

Table 15 – Standard OUTPUTFORMAT values

application/atom+xml; subtype=atom+georss/2.0	Atom+GeoRSS feed.
text/plain[; subtype=<value>]	Plain text response
text/xml; subtype=gml/2.1.2 ^a	GML 2.1.2
text/xml; subtype=gml/3.1.1 ^a	GML 3.1.1
text/xml; subtype=gml/3.1.1/profiles/gmlsf/1.0.0 ^a	Simple feature GML
^a These should probably be application/xml too but for compatibility with the WFS specification, text/xml is used.	

If the **OUTPUTFORMAT** parameter is not specified then the server shall generate the first output format advertised in the server's capabilities document for the **GetFeature** operation. For example, consider the following XML fragment from the capabilities document of a WFS-S:

```
<Operation name="GetFeature">
...
  <Parameter name="OUTPUTFORMAT">
    <Value>text/xml; subtype=atom+georss/2.0</Value>
    <Value>application/bxfs+xml; subtype=bxfs/0.0.3</Value>
    <Value>text/plain</Value>
    <Value>text/xml; gml/2.1.2</Value>
  </Parameter>
...
</Operation>
```

In this case, the default output format would be Atom+GeoRSS.

MAXFEATURES

The value of the **MAXFEATURES** parameter is used to indicate the maximum number of features that the server shall present in the response document. The format of the value of the **MAXFEATURES** parameter is $n, [n2]$.

The value n is a positive integer indicating the maximum number of feature instances that shall be presented in the response document.

The value $n2$ is a positive integer indicating at which feature number, in the current result set, the server shall begin presenting data. The default value is 0 or the first feature. Clients should consider the $n2$ value as a convenience for visual display as feature ordering is not guaranteed to stay the same over multiple requests.

If no value is specified for the **MAXFEATURES** parameter then all feature instances in the result set shall be presented in the response document.

SRSNAME

The **SRSNAME** parameter is used to specify a specific service-supported SRS that shall be used to represent the geometries of features in the response document. All servers shall

support the *urn:x-ogc:def:crs:EPSG:6.3:4326* value but may support additional values as long as they are advertised in the capability document of the service using the **Parameter** element inside the **Operation** element. The following XML fragment shows how **SRSNAME** values may be advertised in the capability document.

```
<Operation name="GetFeature">
...
  <Parameter name="SRSNAME">
    <Value>urn:x-ogc:def:crs:EPSG:6.3:4326</Value>
    <Value>urn:x-ogc:def:crs:EPSG:6.3:32000</Value>
  </Parameter>
...
</Operation>
```

In this example, the server supports two values for the **SRSNAME** parameter.

If the **SRSNAME** parameter is not specified in the request then the response shall be presented using the crs *urn:x-ogc:def:crs:EPSG:6.3:4326*.

PROPERTYNAME

The **PROPERTYNAME** parameter is a comma-separated list of property names (i.e. data fields). This limits the returned properties to a subset of the total. Clients may discover valid property names via the DescribeFeatureType operation. If a service does not advertise a DescribeFeatureType operation, clients should assume that this parameter is NOT supported.

PropertyName

PropertyNameQuery

A **GetFeature** request may optionally contain one or more request parameters of the form:

PropertyNameQuery=regex

where *PropertyNameQuery* is the name of a property of the features and *regex* is a regular expression that shall be applied to value instances of that property to identify which features belong in the result set of the request. Regular expression shall conform to the Unicode Technical Standard #18 – UNICODE REGULAR EXPRESSIONS.

If multiple *PropertyNameQuery* parameters are specified then these shall be logically AND'ed (along with the BBOX and TIME parameters) to form the query predicate for the request.

Only feature properties marked as *queryable* may be specified as parameters in a **GetFeature** request. If a service does not support regex filtering, it shall not advertise any feature properties are being *queryable*.

Editor's Note: I do not believe that this approach to specifying query predicates is complete. First, regex represents only a fraction of the filtering capabilities that one might expect in even the simplest data server; regex is more-or-less equivalent to the LIKE operation in SQL and can only be applied to string literals. What about numeric, date and other property types? Second, allowing the property name to be a request parameter is prone to all sorts of problem related to the lexical space of keywords in a GET request. While any "special" characters can be escaped, this is highly

OGC 07-004

inconvenient! I would, instead, propose that WFS-S include 2 parameters called CONSTRAINT and CONSTRAINTLANGUAGE that can be used to specify regex constraints are described in this subclause BUT can also supports other predicate languages (like CQL). I would also recommend that the specification strongly recommend that CQL (OGC's Common Query Language) be the preferred predicate language ... actually I would go as far as to say that CQL be the only supported predicate language. Regex is simply not sufficient in my opinion.

8.7.3 Response encoding

Currently, this document does not mandate that a WFS-S support any particular output format. The only requirement is that the output formats that a server supports are advertised in its capabilities document. Clients are expected to either know, through prior knowledge, which formats a particular WFS-S instance offers OR they are expected to negotiate with the server to arrive at a mutually agreeable output format. In the latter case the client must read the server's capability document to obtain the list of output formats that the server supports. If the client is able to handle one of the formats in the list then it can continue to interact with the server. Otherwise it can terminate the communication.

Editor's Note: Making OUTPUTFORMAT a list of values would allow the client to declare a list of formats it can deal with and thus the server can either respond with one of those formats or return an exception.

Although this specification does not mandate a particular output format, the preferred output formats are the WFS-S feature collection using BXFS elements and/or Atom+GeoRSS.

8.7.3.1 WFS-S feature collection using BXFS elements

BXFS, normatively defined in Annex B, creates a comprehensive baseline that all WFS Simple services and clients should be able to understand, and therefore promotes the highest level of interoperability across all WFS Simple services. Servers that support BXFS in response to a **GetFeature** operation shall advertise this fact in their capability document using the value *application/bxfs+xml; subtype=bxfs/0.0.3* as described in the table 15.

The following XML Schema fragments defines, using BXFS elements, the response to a **GetFeature** request where the value of the **OUTPUTFORMAT** parameter is set to *application/bxfs+xml; subtype=bxfs/0.0.3*:

```
<xs:element name="FeatureCollection">
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="gml:Envelope" minOccurs="0"/>
      <xs:element ref="bxfs:Properties" minOccurs="0"/>
      <xs:element ref="bxfs:Feature"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="version" type="xs:string"
      use="required" fixed="0.6"/>
    <xs:attribute name="name"
      type="xs:string" use="optional"/>
    <xs:attribute name="srsName"
      type="xs:anyURI" use="optional"/>
    <xs:attribute name="featureCount"
      type="xs:int" use="optional"/>
  </xs:complexType>
</xs:element>
```

The **gml:Envelope** element is used to express the bounding box of all the feature instances contained in the collection.

The **bxfs:Properties** element is used to express the name and types of the properties of the feature. This element is encoded in the same manner as the **bxfs:Properties** element described in subclause 8.6.3.2.

The property values of each feature instance in the collection are encoded within a **bxfs:Feature** element using the **bxfs:Value** element (see Annex B) and are presented in the same order in which they were defined using the **bxfs:Properties** element.

The optional **name** attribute on the **wfss:FeatureCollection** element can be used to assign a name to the collection. The **srsName** attribute is used to encoding the default SRS for the collection. Any geometry in the collection not otherwise labelled shall be assumed to be in this default SRS.

The **featureCount** attribute is used to encode a count of the number of feature instances in the collection.

8.7.3.1.1 Example

Consider the schema from the example in subclause 8.6.3.3 and let's say that the database contains the following information:

name	income	Lat	lon
Homer Simpson	6000	43.6	-79.3
Mr. Burns	250000000	43.2	-79.4

In response to the following **GetFeature** request:

```
http://www.springfield.com/wfss/wfss.cgi?REQUEST=GetFeature&OUTPUTFORMAT="application/bxfs+xml; subtype=bxfs/0.0.3"
```

a WFS-S shall generate the following XML document:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<FeatureCollection
  xmlns="http://www.opengis.net/wfss"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation="http://www.opengis.net/wfss
http://www.ogcnetwork.net/schemas/wfssimple/0.6/wfss.xsd"
  version="0.6"
  name="mydatabasetable"
  srsName="urn:x-ogc:def:crs:EPSG:6.3:4326"
  featureCount="2">
  <gml:Envelope>
    <gml:lowerCorner>43.2 -79.3</gml:lowerCorner>
    <gml:upperCorner>43.6 -79.4</gml:upperCorner>
  </gml:Envelope>
  <Properties>
    <Property name="name" type="xs:string" />
    <Property name="income" type="xs:integer" />
```

OGC 07-004

```
<Property name="location" type="gml:Point" />
</Properties>
<Feature fid="mydatabasetable_1">
  <Val>Homer Simpson</Val>
  <Val>6000</Val>
  <Val>
    <gml:Point>
      <gml:pos>43.6 -79.3</gml:pos>
    </gml:Point>
  </Val>
</Feature>
<Feature fid="mydatabasetable_2">
  <Val>Mr. Burns</Val>
  <Val>250000000</Val>
  <Val>
    <gml:Point>
      <gml:pos>43.2 -79.4</gml:pos>
    </gml:Point>
  </Val>
</Feature>
</FeatureCollection>
```

8.7.3.2 Atom+GeoRSS response

One of the original motivations for defining the WFS-S service was to provide a simple, standard query interface for Atom+GeoRSS feeds. A feed is typically an XML document, conforming to the Atom or RSS formats, that acts as a wrapper for pieces of regularly and sequentially-updated content. GeoRSS is a specification (OGC 06-050r2) describing how to embed location information into a feed.

Providing a WFS-S interface to a feed allows the content of the feed to be filtered based on spatial, temporal and/or non-spatial query constraints. Thus one could, for example, localize a feed providing earthquake information to a specific area of interest. In conjunction with an OPS-B service described in clause 7, feeds can be filtered and pushed to subscribers.

Example

The following example accesses a WFS-S from a fictional organization called “Seismic Events” that provides a feed of world-wide seismic events. The request asks the server to present seismic events during a specific hour of a specific day:

<http://www.se.org/wfss.cgi?SERVICE=WFSS&REQUEST=GetFeature&BBOX=-180,-90,180,90&TIME=2007-03-13T12:00:00Z/2007-03-13T13:00:00Z&OUTPUTFORMAT=application/atom+xml;subtype=atom+georss/2.0>

The response might be:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<rss xmlns="http://blogs.law.harvard.edu/RSS20.xsd"
  xmlns:georss="http://www.georss.org/georss/1.0"
  xmlns:gml="http://www.opengis.net/gml"
  version="2.0">
  <channel>
    <title>Seismic Event WFS-S Reponse</title>
    <link>http://www.se.org/wfss.cgi?request=GetCapabilities</link>
```

```

<description>
  Request Parameters:
    REQUEST=GetFeature&amp;
    BBOX=-180,-90,180,90&amp;
    TIME=2007-03-13T12:00:00Z/2007-03-13T13:00:00Z&amp;
    OUTPUTFORMAT=application/atom+georss
</description>
<language>en-us</language>
<pubDate>Tue, 13 Mar 2007 12:10:35 PDT</pubDate>
<managingEditor>theman@se.org</managingEditor>
<webMaster>nerd@se.org</webMaster>
<item>
  <title>M 2.6, Baja California, Mexico</title>
  <link>http://earthquake.se.org/Quakes/ci10235501.php</link>
  <pubDate>Tue, 13 Mar 2007 12:28:50 GMT</pubDate>
  <guid>ci10235501</guid>
  <georss:where>
    <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
      <gml:pos>32.0228 -116.2473</gml:pos>
    </gml:Point>
  </georss:where>
</item>
<item>
  <title>M 3.0, Andreanof Islands, Aleutian Islands, Alaska</title>
  <link>http://earthquake.se.org/Quakes/ak00077624.php</link>
  <pubDate>Tue, 13 Mar 2007 12:44:29 GMT</pubDate>
  <guid>ak00077624</guid>
  <georss:where>
    <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
      <gml:pos>51.2961 -179.9594</gml:pos>
    </gml:Point>
  </georss:where>
</item>
<item>
  <title>M 2.9, Southern Alaska</title>
  <link>http://earthquake.se.org/Quakes/ak00077611.php</link>
  <pubDate>Tue, 13 Mar 2007 12:50:17 GMT</pubDate>
  <guid>ak00077611</guid>
  <georss:where>
    <gml:Point srsName="urn:ogc:def:crs:EPSG:6.6:4326">
      <gml:pos>61.7548 -151.7645</gml:pos>
    </gml:Point>
  </georss:where>
</item>
</channel>
</rss>

```

Annex A (Normative)

Web Feature Service – Simple Schema (wfss.xsd)

A.1 Introduction

The schema presented in Annex A.2 is used to define the WFS-S response messages, to the **GetCapabilities**, **DescribeFeatureType** and **GetFeature** requests, using elements from BXFS. As has been described in clause 8, BXFS is a preferred output format.

A.2 WFS-S response schema

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<xs:schema
  targetNamespace="http://www.opengis.net/wfss"
  xmlns:wfss="http://www.opengis.net/wfss"
  xmlns:bxfs="http://www.opengis.net/bxfs/0.0.3"
  xmlns:ows="http://www.opengis.net/ows"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:xs="http://www.w3.org/2001/XMLSchema"
  elementFormDefault="qualified"
  version="0.6">

  <xs:import namespace="http://www.opengis.net/bxfs/0.0.3"
    schemaLocation="../../bxfs/0.0.3/bxfs.xsd"/>
  <xs:import namespace="http://www.opengis.net/ows"
    schemaLocation="../../ows/1.0.0/owsAll.xsd"/>

  <xs:annotation>
    <xs:documentation>
      This schema defines the WFS-S response messages.
    </xs:documentation>
  </xs:annotation>

  <xs:element name="FeatureDescription">
    <xs:annotation>
      <xs:documentation>
        The FeatureDescription element is the root element of
        the response to a DescribeFeatureType request. The
        content of the FeatureDescription element is one
        Properties element describing the properties of a
        feature.

        The optional name element may be used to assigne a
        name to the feature collection that this element
        is describing.
      </xs:documentation>
    </xs:annotation>
    <xs:complexType>
      <xs:sequence>
        <xs:element ref="bxfs:Properties" minOccurs="0"/>
      </xs:sequence>
      <xs:attribute name="name" type="xs:string" use="optional"/>
      <xs:attribute name="version" type="xs:string"

```


The 'name' attribute must be specified to indicate the name of the property being described. The name must be unique since it acts as a computer-readable ID/key.

The 'title' attribute is available to provide a human-readable description of 'name'. This value may be used, for example, in a GUI.

The 'queryable' attribute is used to indicate whether the corresponding property can be included in the GetFeature request as the target of a query predicate.

The 'type' attribute covers a range of geographic, XML and RDBMS data types. These map, more or less, directly to GML Simple Features Level 0. If the wfss:Property is the root element of a set of nested wfss:Property element then the type attribute must be omitted.

The attributes 'maxLength', 'precision', and 'scale' are used to encode various facts of the type named using the 'type' attribute.

```

</xs:documentation>
</xs:annotation>
<xs:complexContent>
  <xs:extension base="bxf:AbstractPropertyType">
    <xs:sequence>
      <xs:element ref="wfss:Property"
        minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required"/>
    <xs:attribute name="title" type="xs:string" use="optional"/>
    <xs:attribute name="queryable" type="xs:boolean" use="optional"/>
    <xs:attribute name="type" use="optional" default="xs:string">
      <xs:simpleType>
        <xs:restriction base="xs:QName">
          <xs:enumeration value="xs:string"/>
          <xs:enumeration value="xs:integer"/>
          <xs:enumeration value="xs:double"/>
          <xs:enumeration value="xs:boolean"/>
          <xs:enumeration value="xs:dateTime"/>
          <xs:enumeration value="xs:anyURI"/>
          <xs:enumeration value="xs:base64Binary"/>
          <xs:enumeration value="gml:PointPropertyType"/>
          <xs:enumeration value="gml:LineStringPropertyType"/>
          <xs:enumeration value="gml:PolygonPropertyType"/>
          <xs:enumeration value="gml:GeometryPropertyType"/>
          <xs:enumeration value="gml:EnvelopeType"/>
        </xs:restriction>
      </xs:simpleType>
    </xs:attribute>
    <xs:attribute name="maxLength" type="xs:nonNegativeInteger"/>
    <xs:attribute name="precision" type="xs:positiveInteger"/>
    <xs:attribute name="scale" type="xs:integer"/>
  </xs:extension>
</xs:complexContent>
</xs:complexType>

<xs:element name="WFS_Simple_Capabilities">
  <xs:complexType>
    <xs:annotation>
      <xs:documentation>
        XML encoded WFS-S GetCapabilities operation response. This
        document provides clients with service metadata about a
        specific service instance, including metadata about the
        tightly-coupled data served. If the server does not implement

```

the updateSequence parameter, the server shall always return the complete Capabilities document, without the updateSequence parameter. When the server implements the updateSequence parameter and the GetCapabilities operation request included the updateSequence parameter with the current value, the server shall return this element with only the "version" and "updateSequence" attributes. Otherwise, all optional elements shall be included or not depending on the actual value of the Contents parameter in the GetCapabilities operation request.

```
</xs:documentation>
</xs:annotation>
<xs:complexContent>
  <xs:extension base="ows:CapabilitiesBaseType"/>
</xs:complexContent>
</xs:complexType>
</xs:element>

</xs:schema>
```

Annex B (normative)

Basic XML Feature Schema (BXFS)

B.1 XML schema for BXFS

```

<?xml version="1.0" encoding="UTF-8"?>
<schema
  targetNamespace="http://www.opengis.net/bxfs/0.0.3"
  xmlns="http://www.w3.org/2001/XMLSchema"
  xmlns:xlink="http://www.w3.org/1999/xlink"
  xmlns:bxfs="http://www.opengis.net/bxfs/0.0.3"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:ows="http://www.opengis.net/ows"
  elementFormDefault="qualified"
  version="0.0.3">

  <import namespace="http://www.opengis.net/ows"
    schemaLocation="../../ows/1.0.0/owsAll.xsd"/>

  <!-- type importing one lower level XSD -->
  <import namespace="http://www.opengis.net/gml"
    schemaLocation="../../gml/3.1.1/base/geometryAggregates.xsd"/>

  <element name="FeatureCollection">
    <annotation>
      <documentation>
        A FeatureCollection is a heterogeneous collection of one or
        more FeatureSet element. A FeatureSet element contains one
        or more homogeneous feature instances.

        A FeatureCollection contains may optionally contain a
        FeatureSetSchemas element that describes the properties
        each FeatureSet in the collection.
      </documentation>
    </annotation>
    <complexType>
      <complexContent>
        <extension base="bxfs:BaseObjectType">
          <sequence>
            <element name="FeatureSetSchemas"
              type="bxfs:FeatureSetSchemasType"
              minOccurs="0"/>
            <element name="FeatureSet"
              type="bxfs:FeatureSetType"
              maxOccurs="unbounded"/>
          </sequence>
          <attribute name="version" type="string"
            use="required" fixed="0.0.3"/>
        </extension>
      </complexContent>
    </complexType>
  </element>

  <complexType name="FeatureSetSchemasType">
    <annotation>
      <documentation>

```

A FeatureSetSchemass element is a catalog of schemas describing the properties of each homogeneous FeatureSet contained within a FeatureCollection element.

This is basically of copy of the information that is already contained within the Properties element inside the FeatureSet element.

The reason for allowing both is to support streaming as well as random-access to the file.

A compliant document may specify the feature set schemas in one or both locations (i.e. within the FeatureSetSchemas element or the Properties element) If the feature set schema is specified in both locations, then the schemas in both places must match. If the FeatureSetSchemas element is used, the schemas must be defined in the order in which the FeatureSet elements will be presented in the collection.

```

</documentation>
</annotation>
<complexContent>
  <extension base="bxf:BaseObjectType">
    <sequence>
      <element name="FeatureSetSchema"
        type="bxf:FeatureSetSchemaType" maxOccurs="unbounded"/>
    </sequence>
  </extension>
</complexContent>
</complexType>
<complexType name="FeatureSetSchemaType">
  <annotation>
    <documentation>
      A FeatureSetSchema element defines the properties of one
      FeatureSet. The property definition includes the name and
      type of each FeatureSet property.

      The value of the 'name' attribute must match the name of
      the FeatureSet that the FeatureSetSchema described.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="bxf:BaseObjectType">
      <sequence>
        <element ref="bxf:Properties"/>
      </sequence>
      <attribute name="name" type="string" use="required"/>
    </extension>
  </complexContent>
</complexType>
<complexType name="FeatureSetType">
  <annotation>
    <documentation>
      A feature set is a homogeneous collection of feature instances.
      Homogeneous features have the same properties. A FeatureSet is
      composed of an optional Properties element and zero or more
      Feature elements.

      The Properties element is used to declare the schema of the
      feature by enumerating the name, type and other facets of
      each property of the feature set.

      The Feature element is used to encode the values of each
      property of one instance of a feature. The FeatureSet

```

```

    element can contain an unlimited number of Feature element
    encoding the property values of an unlimited set of feature
    instances.
  </documentation>
</annotation>
<complexContent>
  <extension base="bxf:s:BaseObjectType">
    <sequence>
      <element ref="bxf:s:Properties" minOccurs="0"/>
      <element ref="bxf:s:Feature"
        minOccurs="0" maxOccurs="unbounded"/>
    </sequence>
    <attribute name="name" type="string" use="required">
      <annotation>
        <documentation>The name of the feature set.</documentation>
      </annotation>
    </attribute>
    <attribute name="srsName" type="anyURI" use="optional">
      <annotation>
        <documentation>
          The srsName attribute is the name of the default SRS.
          Any geometry in the set not otherwise labeled should
          be assumed to be in the default SRS.
        </documentation>
      </annotation>
    </attribute>
    <attribute name="featureCount" type="integer" use="optional">
      <annotation>
        <documentation>
          A count of the number of feature instances in the set.
        </documentation>
      </annotation>
    </attribute>
  </extension>
</complexContent>
</complexType>

<element name="Feature" type="bxf:s:FeatureType"/>
<complexType name="FeatureType">
  <annotation>
    <documentation>
      A Feature is a single instance of a feature in an FeatureSet.
      A Feature contains one or more value elements each encoding
      the value of one property of the feature. The property values
      MUST be presented in the order in which they are defined within
      the Properties element contained within the FeatureSetSchemas
      and/or FeatureSet elements. In other words, the set of property
      values must positionally match the properties defined in the
      schema of the feature set.

      Each feature instance must be assigned a feature identifier
      using the fid attribute.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="bxf:s:BaseObjectType">
      <sequence>
        <element ref="bxf:s:AbstractValue" maxOccurs="unbounded"/>
      </sequence>
      <attribute name="fid" type="anyURI" use="required"/>
    </extension>
  </complexContent>
</complexType>

<element name="AbstractValue"
  type="bxf:s:AbstractValueType" abstract="true"/>

```

```

<complexType name="AbstractValueType" abstract="true">
  <annotation>
    <documentation>
      Head of a substitution group for the allowed property value
      elements. This is mostly included to support single values
      as well as arrays of values.
    </documentation>
  </annotation>
</complexType>

<element name="Value" type="bxf:ValueType"
  substitutionGroup="bxf:AbstractValue"/>
<complexType name="ValueType" mixed="true">
  <annotation>
    <documentation>
      The Value element is used to encode a single value for a
      property. The content of the Value element is either a
      text string representing the value of the property or it
      is a geometry expressed using GML.

      If the isNull attribute is specified and is set to true,
      the content of the Value element shall be empty. If the
      isNull attribute is true and the content of the Value
      element is not empty, the content shall be ignored and
      the value shall be interpreted as NULL.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="bxf:AbstractValueType">
      <sequence>
        <any processContents="strict" namespace="##other"/>
      </sequence>
      <attribute name="isNull" type="boolean"
        use="optional" default="false"/>
    </extension>
  </complexContent>
</complexType>

<element name="ValueArray" type="bxf:ValueArrayType"
  substitutionGroup="bxf:AbstractValue"/>
<complexType name="ValueArrayType">
  <annotation>
    <documentation>
      The ValueArray element is used when the value of a property
      is actually an array of values. Arrays can be represented
      in one of two ways.

      1. Space-separated list of values.(e.g. <ValueList>1 2 3
        4 5</ValueList>;)

      2. List of <Value> elements.(e.g. <Value>1</Value>
        <Value>2</Value><Value>3</Value>
        <Value>4</Value><Value>5</Value>;)

      The first representation is suitable for stuff like lists of
      numbers (1,2,3,4,5) while the latter representation is suitable
      for representing lists of complex structures like GML geometries.

      The ordered attribute may be used to indicate whether the
      values in the list are ordered or not.
    </documentation>
  </annotation>
  <complexContent>
    <extension base="bxf:AbstractValueType">
      <choice>
        <element ref="bxf:ValueList"/>
      </choice>
    </extension>
  </complexContent>
</complexType>

```

```

        <element ref="bxfs:Value"
            minOccurs="0" maxOccurs="unbounded"/>
    </choice>
    <attribute name="ordered" type="boolean"
        use="optional" default="false"/>
</extension>
</complexContent>
</complexType>
<element name="ValueList" type="bxfs:ValueListType"/>
<simpleType name="ValueListType">
    <list itemType="string"/>
</simpleType>

<element name="Properties" type="bxfs:PropertiesType"/>
<complexType name="PropertiesType">
    <annotation>
        <documentation>
            The Properties element is used to specify the names, types and
            other facets of all the properties in a feature set. The
            The order in which the properties are defined is also the
            order in which the property values must be presented within
            the Feature element.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="bxfs:BaseObjectType">
            <choice>
                <element ref="bxfs:AbstractProperty" maxOccurs="unbounded"/>
            </choice>
        </extension>
    </complexContent>
</complexType>

<element name="AbstractProperty"
    type="bxfs:AbstractPropertyType" abstract="true"/>
<complexType name="AbstractPropertyType" abstract="true">
    <annotation>
        <documentation>
            This abstract element allows application schemas using bxfs
            to redefine what a property is ... either making it simpler
            or extending it.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="bxfs:BaseObjectType"/>
    </complexContent>
</complexType>

<element name="Property" type="bxfs:PropertyType"
    substitutionGroup="bxfs:AbstractProperty"/>
<complexType name="PropertyType">
    <annotation>
        <documentation>
            The Property element is used to define a single property
            of a feature. The Property element can contain an
            optional ValueDescription element (used to describe
            particular values that the property can have or
            enumerate a list of values) and an optional nested
            Property element.
        </documentation>
    </annotation>
    <complexContent>
        <extension base="bxfs:AbstractPropertyType">
            <sequence>
                <choice>
                    <element ref="bxfs:ValueDescriptions" minOccurs="0"/>
                </choice>
            </sequence>
        </extension>
    </complexContent>
</complexType>

```

```

<element ref="bxf:Property"
  minOccurs="0" maxOccurs="unbounded">
  <annotation>
    <documentation>
      Nested Property elements may be used to
      describe an aggregate or complex property.
      In this case, the top-level Property element
      shall only include the following list of
      attributes:

          name
          nullable
          isArray
          maxArrayLength
          querytable

      All other property attributes are prohibited
      at the top level.
    </documentation>
  </annotation>
</element>
</choice>
</sequence>
<attribute name="name" type="string" use="required">
  <annotation>
    <documentation>
      The name of the property.
    </documentation>
  </annotation>
</attribute>
<attribute name="type">
  <annotation>
    <documentation>
      The type of the property.

      This attribute shall not be specified for
      aggregate properties.
    </documentation>
  </annotation>
  <simpleType>
    <restriction base="QName">
      <enumeration value="string"/>
      <enumeration value="integer"/>
      <enumeration value="double"/>
      <enumeration value="dateTime"/>
      <enumeration value="boolean"/>
      <enumeration value="anyURI"/>
      <enumeration value="base64Binary"/>
      <enumeration value="gml:PointPropertyType"/>
      <enumeration value="gml:LineStringPropertyType"/>
      <enumeration value="gml:PolygonPropertyType"/>
      <enumeration value="gml:GeometryPropertyType"/>
      <enumeration value="gml:EnvelopeType"/>
      <enumeration value="gml:MultiPointPropertyType"/>
      <enumeration value="gml:MultiLineStringPropertyType"/>
      <enumeration value="gml:MultiPolygonPropertyType"/>
      <enumeration value="gml:MultiGeometryPropertyType"/>
    </restriction>
  </simpleType>
</attribute>
<attribute name="maxLength" type="nonNegativeInteger">
  <annotation>
    <documentation>
      The maxLength attribute is used to specify the
      maximum length (in # of characters) of a string
      where the type attribute is set to 'string'.
    </documentation>
  </annotation>
</attribute>

```

```

        This attribute shall not be specified for
        aggregate properties.
    </documentation>
</annotation>
</attribute>
<attribute name="precision" type="positiveInteger">
    <annotation>
        <documentation>
            The total number of digits in a number when
            the type attribute is set to xs:integer or
            xs:double.

            This attribute shall not be specified for
            aggregate properties.
        </documentation>
    </annotation>
</attribute>
<attribute name="scale" type="integer">
    <annotation>
        <documentation>
            The number of digits after decimal point with
            the type attribute is set to xs:double.

            This attribute shall not be specified for
            aggregate properties.
        </documentation>
    </annotation>
</attribute>
<attribute name="nullable" type="boolean" default="true">
    <annotation>
        <documentation>
            The nullable attribute is used to indicate whether
            the value of the property can be NULL.
        </documentation>
    </annotation>
</attribute>
<attribute name="isArray" type="boolean" default="false">
    <annotation>
        <documentation>
            The isArray attribute is used to indicate whether
            the property is an array of the specified type.
            The attribute arrayLength may optionally be used
            to set the number of elements of the array.
        </documentation>
    </annotation>
</attribute>
<attribute name="maxArrayLength" type="nonNegativeInteger">
    <annotation>
        <documentation>
            If the isArray attribute is set to true, the maxArrayLength
            attribute may be specified to set the max length of
            the array in terms of number of elements.
            Otherwise the array length is assumed to be unbounded.
        </documentation>
    </annotation>
</attribute>
<attribute name="queryable" type="boolean" default="true">
    <annotation>
        <documentation>
            The queryable attribute is used to indicate if
            the property may be used in a query predicate.
            The typical use case for this capability is to
            prevent reverse phone-number lookup which is
            illegal in many jurisdictions.
        </documentation>
    </annotation>
</attribute>

```

```

        </annotation>
      </attribute>
    </extension>
  </complexContent>
</complexType>

<element name="ValueDescriptions" type="bxf:ValueDescriptionsType"/>
<complexType name="ValueDescriptionsType">
  <annotation>
    <documentation>
      The ValueDescriptions element is meant to serve two
      purposes:

      1. Allow a description to be attached to specified
         values that a property can have. For example,
         the description 'sea level' might be attached
         to the value 0 for a property named 'elevation'.

      2. Enumerate a list of values that the property can
         have. Such lists might, for example, be used to
         create pick lists in a GUI.

      The ValueDescriptions element may also be used with
      arrays of values to assign descriptions to particular
      combinations of values. For example, the combination
      {255,255,255} may be assigned the description "white"
      for an array property called 'colour' where the three
      values are interpreted as being RGB values.

      The list of values contained in the ValueDescriptions
      element can be a CLOSED list or an OPEN list.
    </documentation>
  </annotation>
  <sequence>
    <element name="ValueDescription"
      type="bxf:ValueDescriptionType" maxOccurs="unbounded"/>
  </sequence>
  <attribute name="bounded" type="boolean" default="false">
    <annotation>
      <documentation>
        The bounded attribute indicates whether the list
        of values contained within the ValueDescriptions
        element, is a CLOSED list or an OPEN list.

        A CLOSED list means that the property can only have
        the values described within the ValueDescriptions
        element.

        An OPEN list means that the property can have
        value outside the list of values contained within
        the ValueDescriptions element.

        - if you have a open list does not make sense to
          values that don't have a description
        -
      </documentation>
    </annotation>
  </attribute>
</complexType>
<complexType name="ValueDescriptionType">
  <annotation>
    <documentation>
      A ValueDescription consists of a Value element containing
      the value (or array of values) being described and an
      optional Description element containing the actual
      description.
    </documentation>
  </annotation>

```

OGC 07-004

The Description element is optional to accomodate the fact that if this is a closed list-of-values, not every value may have a description.

```
</documentation>
</annotation>
<sequence>
  <element ref="bxf:AbstractValue"/>
  <element ref="bxf:Description" minOccurs="0"/>
</sequence>
</complexType>

<element name="Description" type="ows:IdentificationType"/>
<complexType name="BaseObjectType">
  <annotation>
    <documentation>
      A description is a very lightweight structure for attaching
      metadata to various elements. More detailed metadata (such
      as that that may be stored in a registry) can be referenced
      by using the MetadataURL element. The optional standard
      attribute may be used to indicate the type of metadata
      being pointed to. For example, FGDC or ISO19119 or ...
      The url content is the pointer to the detailed metadata and
      may in fact point to metadata stored in a catalog.
    </documentation>
  </annotation>
  <sequence>
    <element ref="bxf:Description" minOccurs="0"/>
  </sequence>
</complexType>

</schema>
```

Annex C

Bibliography