

Open Geospatial Consortium

Date: 2011-02-14

External identifier of this OGC® document: <http://www.opengis.net/doc/spec/opensearchgeo/1.0>

Internal reference number of this document: OGC 10-032r3

Version: 0.0.3

Category: OpenGIS® Implementation Standard

Editor: Pedro Gonçalves

OGC® OpenSearch GeoSpatial and Temporal Extensions

Copyright © Open Geospatial Consortium, Inc.

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

Warning

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

Recipients of this document are invited to submit, with their comments, notification of any relevant patent rights of which they are aware and to provide supporting documentation.

Document type:	Candidate OpenGIS® Standard
Document subtype:	
Document stage:	Draft proposed version
Document language:	English

Contents	Page
i. Preface	iv
iii. Submitting organizations.....	iv
iv. Document contributor contact points.....	v
v. Revision history	v
vi. Changes to the OGC Abstract Specification.....	v
Foreword.....	vi
Introduction.....	vii
1 Scope.....	1
2 Compliance	1
3 Normative references	2
4 Terms and definitions	3
5 Conventions	3
5.1 Abbreviated terms.....	3
6 OpenSearch Geospatial and Temporal Extensions overview	4
7 Shared aspect	5
8 OpenSearch Description (Requirement).....	6
8.1 Introduction.....	6
9 OpenSearch Geospatial and Temporal Extensions operation.....	7
9.1 Introduction.....	7
9.2 Search operation request.....	8
9.2.1 Search request parameters.....	8
9.2.2 Search request KVP encoding (mandatory).....	11
9.3 Search operation response.....	12
9.3.1 Normal response parameters.....	12
9.3.2 Normal response XML encoding.....	12
9.3.3 ATOM 1.0 response example	17
9.3.4 ATOM 1.0 response example for Earth Observation Imagery.....	18
Annex A (normative) Abstract test suite	20
A.1 OpenSearch description document.....	20
A.2 ATOM response type.....	20
A.3 Bounding box search.....	20
A.4 Arbitrary geometry search	21
A.5 Point and radius search	21
A.6 Min spatial operator	21
A.7 Spatial operators.....	21
A.8 Get record by id.....	22

A.9 Search by name	22
A.10 Temporal search.....	22
Annex B (normative) XML Schema Documents.....	23
Annex C (informative) UML model.....	24
Annex D (informative) OpenSearch 1.1 Draft 4.....	25
D.1 Introduction.....	25
Namespace.....	25
OpenSearch description document	25
Extensibility	26
OpenSearch description elements.....	26
Autodiscovery	33
OpenSearch URL template syntax.....	35
OpenSearch 1.1 parameters	38
OpenSearch Query element	40
OpenSearch Response Elements.....	45
Annex E (informative) Example XML documents.....	49
E.1 OpenSearch Description with Geospatial and Temporal Extensions.....	49
E.2 ATOM Output with access to data resources	50
E.3 ATOM Output with access to Geospatial Search Services resources	53

Tables	Page
Table 1 – Conformance Classes.....	2
Table 2 — Parameters in a Search operation request	8
Table 3 — Search operation queryable mappings	10
Table 4 — Search operation request URL parameters	11
Table 5 —Elements of Search operation response in the atom:feed element describing the search service.....	13
Table 6 —Elements of Search operation response in the atom:entry elements describing each search result	14

i. Preface

This document is the result of work undertaken within the GENESI-DR (Ground European Network for Earth Science Interoperations - Digital Repositories) project funded by the 7th Framework program of the European (EC Grant Agreement no. 212073) and the follow-up project GENESI-DEC (Ground European Network for Earth Science Interoperations -Digital Earth Community) funded by the same program (Contract n° RI-261623).

ii. Document terms and definitions

This document uses the standard terms defined in Subclause 5.3 of [OGC 05-008], which is based on the ISO/IEC Directives, Part 2. Rules for the structure and drafting of International Standards. In particular, the word “shall” (not “must”) is the verb form used to indicate a requirement to be strictly followed to conform to this standard.

iii. Submitting organizations

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

- Terradue SRL
- FGDC
- FortiusOne, Inc.
- EDINA
- Cubewerx Inc.

iv. Document contributor contact points

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

Name	Organization	Email
Pedro Goncalves	Terradue Srl	pedro.goncalves@terradue.com
Jo Walsh	EDINA	jo.walsh@ed.ac.uk
Andrew Turner	FortiusOne, Inc.	andrew@fortiusone.com
Panagiotis A. Vretanos	Cubewerx Inc.	pvretano@cubewerx.com
Douglas D. Nebert	FGDC	ddnebert@usgs.gov

v. Revision history

Date	Release	Editor	Primary clauses modified	Description
2010-09-20	0.0.1	Pedro Gonçalves		Initial Text with GeoSpatial and Temporal Extension Include OpenSearch specification in Annex
2010-02-09	0.0.2	Pedro Gonçalves		Alignment with the Modular Specification document Addition of ATOM examples
	0.0.3	Pedro Gonçalves		Small typos and corrections Added example of CSW 2.0

vi. Changes to the OGC Abstract Specification

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

Foreword

This document is the specification for the OpenSearch Geo and Temporal protocol.

This candidate standard is intended to provide a very simple way to make spatial queries to a repository of geospatial content that contains geographic and temporal properties, to allow simple syndication between, and to provide a basic “federated” query of related repositories, whereby a single client can query several server instances and present a collection of results as one set.

The use cases at OpenSearch.org deal primarily with full-text search across textual data, but the protocol is equally applicable to simple searches across collections of structured data – particularly data with a geospatial component.

Attention is drawn to the possibility that some of the elements of this document may be the subject of patent rights. The Open Geospatial Consortium Inc. shall not be held responsible for identifying any or all such patent rights.

Recipients of this document are requested to submit, with their comments, notification of any relevant patent claims or other intellectual property rights of which they may be aware that might be infringed by any implementation of the standard set forth in this document, and to provide supporting documentation.

Introduction

The OpenSearch specification originated in a community effort built around Amazon's A9.com. It was intended to allow syndication of search results that could then be aggregated by one large index. The OpenSearch specification is made available under the Creative Commons Attribution-Sharealike 2.5 license. In addition, the OASIS Search Web Services group is publishing an Abstract Protocol Definition of the interface or “binding”, which coincides with the community specification published at <http://opensearch.org>. In 2007, a set of geospatial and temporal extensions were proposed through OpenSearch.org.

This document codifies those extensions in a form compatible with the OGC standards baseline and incorporates feedback from developers in the open source geospatial community. This document includes several annexes showing result sets in several possible formats and giving details that reflect a sample implementation.

This standard has been promoted and adopted in the context of the GENESI-DR (Ground European Network for Earth Science Interoperations - Digital Repositories) project by several European Earth Science data centres as the minimal compliance level to develop discovery mechanisms in heterogeneous sites and to transform the repository network into a virtual distributed repository. The GENESI-DR was a 2-year Framework Programme 7 project that is EC-funded under the INFRASTRUCTURES-2007-1, Capacity Programme, INFRA-2007-1.2.1 Scientific Digital Repositories. The project consortium was led by ESA and included 13 partners holding several Earth Observation data repositories.

OpenGIS[®]

1 Scope

This OGC candidate standard specifies a set of geospatial and temporal extensions to the OpenSearch query protocol. OpenSearch is a collection of simple formats for the sharing of search results.

The OpenSearch description document format can be used to describe a search engine so that it can be used by search client applications. The OpenSearch description format allows the use of extensions that allow search engines to request a specific and contextual query parameter from search clients.

The OpenSearch response elements can be used to extend existing syndication formats, such as RSS and Atom, with the extra metadata needed to return search results.

Services that support the OpenSearch Specification and the Geospatial and Temporal extension defined in this document are called OpenSearch Geospatial Services.

2 Compliance

Conformance with this standard shall be checked using all the relevant tests specified in Annex A (normative) of this document. The framework, concepts, and methodology for testing, and the criteria to be achieved to claim conformance are specified in the OGC Compliance Testing Policies and Procedures and the OGC Compliance Testing web site¹.

In order to conform to this OGC[™] interface standard, a software implementation shall choose to implement Any one of the conformance levels specified in Annex a (normative).

All requirements-classes and conformance-classes described in this document are owned by the specification identified as <http://www.opengis.net/spec/opensearchgeo/1.0>.

The following table describes the conformance classes defined in this standard:

¹ www.opengeospatial.org/cite

Table 1 – Conformance Classes

Conformance Class Name	Description	Test
OpenSearch Description	The server generates a valid OpenSearch description document.	A.1
ATOM Response Type	The server defines an URL template for the ATOM response type.	A.2
Bounding Box Search	The server implements a bounding box search.	A.3
Arbitrary Geometry Search	The server implements a spatial search capability using an arbitrary geometry.	A.4
Point and Radius Search	The server implements a spatial search using a point and a radius.	A.5
Minimum spatial operator	The server implements the “overlaps” spatial operator.	A.6
Spatial operators	The server implements the spatial operators “contains” and “disjoint”.	A.7
Get record by id.	The server can retrieve a record from its repository using the id of the record.	A.8
Search by name	The server implements a spatial search using the name or address of a location.	A.9
Temporal search	The server implements a temporal search capability.	A.10

3 Normative references

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

ATOM - The Atom Syndication Format (IETF RFC4287)

OASIS OpenSearch - *Search Web Services searchRetrieve Operation: Binding for OpenSearch Version 1.0* <http://docs.oasis-open.org/search-ws/v1.0/opensearch-v1.0.html>

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

OGC 06-050 - Open Geospatial Consortium: An Introduction to GeoRSS

OGC 06-121 - OWS Common Implementation Specification

OGC 06-103r4 - OpenGIS® Implementation Standard for Geographic information - Simple feature access - Part 1: Common architecture

OGC 07-147r2 - *OGC KML Specification*

W3C RDF - *W3C RDF Specification Suite* - <http://www.w3.org/RDF/>

RFC 3066 - Tags for the Identification of Languages - <http://tools.ietf.org/html/rfc3066>

RFC 3339 - *Date and Time on the Internet: Timestamps*. Internet profile of the ISO 8601 standard for representation of dates and times using the Gregorian calendar.
<http://www.ietf.org/rfc/rfc3339.txt>

RFC 5988 - *Web Linking* - <http://tools.ietf.org/html/rfc5988>
<http://www.iana.org/assignments/link-relations/link-relations.xhtml>

4 Terms and definitions

For the purposes of this standard, the definitions specified in Clause 4 of the OWS Common Implementation Specification [OGC 05-008] shall apply. In addition, the following terms and definitions apply.

4.1

OpenSearch

Draft specification for web search syndication, originating from Amazon's A9 project and given a corresponding interface binding by the OASIS Search Web Services working group.

4.2

OpenSearch Geospatial Service

Defines servers that comply with the OpenSearch Specification and the Geospatial extension defined in this document

4.3

OpenSearch Description Document

An XML document available at a consistent location describing metadata for the service and providing templates for queries.

5 Conventions

5.1 Abbreviated terms

Some of the abbreviated terms listed in Subclause 5.1 of the OWS Common Implementation Specification [OGC 06-121] apply to this document, plus the following abbreviated terms.

EC European Commission

ESA European Space Agency

EU	European Union
GENESI-DR Repositories	Ground European Network for Earth Science Interoperations – Digital Repositories
GeoJSON	Geospatial Extensions to JavaScript Object Notation
GeoRSS	Geospatial Extensions to Rich Site Syndication
INSPIRE	Infrastructure for Spatial Information in Europe
KML	Keyhole Markup Language
RDF	Resource Description Framework
REST	Representational State Transfer
SWS	Search Web Services
URL	Uniform Resource Locator
UUID	Universal Unique Identifier
WKT	Well Known Text

6 OpenSearch Geospatial and Temporal Extensions overview

As the operations of the main OpenSearch specification are expressed in the OASIS SWS bindings [OASIS OpenSearch], we confine ourselves here to the operations of the OpenSearch Geospatial and Temporal extensions. A short overview of how they fit within the primary OpenSearch specification is provided. Annex D covers the main aspects of the OpenSearch specification.

The basic concept of OpenSearch is to specify how to query a web resource, and additional metadata to support syndicating the results. Search clients can use OpenSearch description documents to learn about the public interface of a search engine. These description documents contain parameterized URL templates that indicate how the search client should make search requests. Search engines can use the OpenSearch response elements to add search metadata to results in a variety of content formats. For example, if a web site allows search by the URL:

<http://www.example.com?q=question>

OpenSearch provides a way to define where that search term goes. Essentially it would look like: <http://www.example.com?q={searchTerms}>, where {searchTerms} would be replaced by any general string. Using OpenSearch, aggregators and applications have a way to simply define a search service and let a user just type in their terms, but then search N search engines. For example, the Firefox search bar is powered by OpenSearch and allows the user to add new OpenSearch compliant site.

The Description element provides some basic metadata about the contents of the service and the contact information pertaining to it, along with a set of URL Templates which illustrate the parameters accepted by the service and the variety of output formats in which results can be obtained. The OpenSearch request interface is simple, consisting of a description of a HTTP GET request with a series of optional key-value parameters that can be used to constrain the search.

The Geospatial and Temporal Extensions specify a series of parameters that can be used to geographically constrain search results. These are discussed in more detail in Section 9. In short, provision is made to filter results by:

- a) A bounding box
- b) An arbitrary geometry defined using Well Known Text
- c) Within a certain radius from a given latitude-longitude point
- d) Having a certain containment relation (within, overlaps, disjoint) with a geographic constraint
- e) Matching a geographic name (not considered in our use cases, but forming part of the original specification)
- f) Matching a given unique identifier in the context of the repository
- g) A start and end of a temporal extent

All geographic information is assumed to be expressed using the EPSG 4326 (WGS84 2d)² coordinate reference system (CRS), and any output listing the extents of the original query or containing further geographic information about the search results, will similarly be expressed in EPSG 4326. There is precedent for this in related OGC “Mass Market” standards, specifically GeoRSS [OGC 06-050] as well as many other standards that encode location.

A sample of a Description showing the use of Geospatial Extensions with several output formats appropriate for geographic search results can be seen in Annex E.

7 Shared aspect

Not applicable

² <http://spatialreference.org/ref/epsg/4326/>

8 OpenSearch Description (Requirement)

8.1 Introduction

The OpenSearch Description Document allows clients to retrieve service metadata from a server. The response to a request for a Description shall be an XML document in the form specified by the SWS OpenSearch bindings published by OASIS. A brief summary of the Description's requirements is offered here.

The service metadata covered by the OpenSearch Description corresponds to the service metadata returned by e.g. a WMS GetCapabilities request. The request is not parameterised, but simply takes the form of an HTTP GET request to a URL (advertised on a web page or obtained from a repository of OpenSearch services which offer syndication of their result sets).

The Description document contains information on the maintainer of the service and their contact information. It includes a brief title and short abstract describing the contents of the service, along with a list of keywords and optionally a list of languages in which the contents may be returned.

The key feature of the Description document is the URL template, which instructs a client application how to issue queries to the service. The URL template represents a parameterized form of the URL by which a search engine is queried. Each response format supported by the service needs its own distinct URL template included in the Description. A service can extend the basic parameters included in the template.

EXAMPLE 1 The following XML document provides a sample response to a request for an OpenSearch Description from a repository supporting the Geospatial Extensions. Note that the response type is given in the type attribute and not implicitly on the URL path

```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/">
  <ShortName>Web Search</ShortName>
  <Description>Use Example.com to search the Web.</Description>
  <Tags>example web</Tags>
  <Contact>admin@example.com</Contact>
  <Url type="application/vnd.google-earth.kml+xml"
    template="http://example.com/mykml/?q={searchTerms}&pw={startPage?}&bbox={geo:box?}" />
  <Url type="application/rdf+xml"
    template="http://example.com/myrdf/?q={searchTerms}&pw={startPage?}&bbox={geo:box?}" />
  <Url type="application/geojson"
    template="http://example.com/geojson/?q={searchTerms}&pw={startPage?}&bbox={geo:box?}" />
  <Url type="text/html"
    template="http://example.com/?q={searchTerms}&bbox={geo:box?}&pw={startPage?}" />
  <LongName>Example.com Web Search</LongName>
  <Query role="example" searchTerms="cat" geo:box="10,10,12,12" />
  <Attribution>Copyright 2005, Example.com, Inc.</Attribution>
  <SyndicationRight>open</SyndicationRight>
</OpenSearchDescription>
```

For detail on the required and optional fields please refer to the information on Annex D.

9 OpenSearch Geospatial and Temporal Extensions operation

9.1 Introduction

OpenSearch Geospatial and Temporal Extensions are designed to provide a basic geographical and temporal search facility to the OpenSearch protocol. Two namespace are provided for use in URL templates that form part of the OpenSearch Description Document (see Annex E for an example), published by the server in XML at a stable URL.

Requirements Class	
http://www.opengis.net/spec/opensearchgeo/1.0/req/req-class-a	
Target type	Token
Requirement	<p>http://www.opengis.net/spec/opensearchgeo/1.0/req/req-class-a/namespaces</p> <p>These namespaces and a corresponding namespace prefix shall be included when the extensions are used in an OpenSearch Description document, to whit:</p> <ul style="list-style-type: none"> • http://a9.com/-/opensearch/extensions/geo/1.0/ • http://a9.com/-/opensearch/extensions/time/1.0/

OpenSearch has a lot more functionality and is especially useful when requesting syndication formats such as ATOM or RSS. With the OpenSearch geospatial extension to it is possible to formulate requests to all records found within a spatial area defined as a point-plus-radius, a bounding box, or a geometry. Together with the Time extension, OpenSearch can specify time start and finish slices for searching data.

The flexibility of the OpenSearch protocol allows one to return lists of search results in any format that a client can be persuaded to understand. A server provides a description document that a client reads to determine how to formulate a search/retrieve request and interpret the response. The OpenSearch Description Document includes a mandatory URL element containing a mandatory request template. Where several request templates are provided, a client may choose the one offering the most useful format (specified by MIME-type defined in the *type* attribute of the element) as shown in Example 1.

Requirements Class	
http://www.opengis.net/spec/opensearchgeo/1.0/req/req-class-a	
Target type	Token
Requirement	<p>http://www.opengis.net/spec/opensearchgeo/1.0/req/req-class-a/response_payload</p> <p>Any server supporting the geospatial or temporal extension shall at least return results in Atom 1.0 and use GeoRSS simple for geospatial information</p>

9.2 Search operation request

9.2.1 Search request parameters

The following parameters may be submitted as part of a search request.

Table 2 — Parameters in a Search operation request

Names ^a	Definition	Data type and values	Multiplicity and use
Geospatial Extension (http://a9.com/-/opensearch/extensions/geo/1.0/)			
box	Geographic bounding box	The box is defined by "west, south, east, north" coordinates of longitude, latitude, in a EPSG:4326 decimal degrees	One (optional)
geometry	Geographic area (geometry)	The geometry is defined using the Well Known Text and supports the following 2D geographic shapes: <ul style="list-style-type: none"> • POINT, • LINESTRING, • POLYGON, • MULTIPOINT • MULTILINESTRING • MULTIPOLYGON The Geometry shall be expressed using the EPSG:4326.	One (optional)
uid	Unique identifier of the record in the repository context	Character String	One (optional)
lat	The latitude of a given point	Latitude in decimal degrees in EPSG:4326	One (optional)
lon	The longitude of a given point	Longitude in decimal degrees in EPSG:4326	One (optional)
radius	A search radius from a lat-lon point	The distance in meters along the Earth's surface.	One (optional)
relation	Spatial relation to result set	Character String; One of "overlaps", "contains", "disjoint"	One (optional) default is "overlaps"
name	A string describing the location to search.	Character String	One (optional)
Temporal Extension (http://a9.com/-/opensearch/extensions/time/1.0/)			
start	A string describing the start of the temporal interval to search.	Character String; must match the RFC-3339 (also used by the Atom syndication format).	One (optional)
end	A string describing the end of the temporal interval to search.	Character String; must match the RFC-3339 (also used by the Atom syndication format).	One (optional)
a The name capitalization rules being used here are specified in Subclause 11.6.2 of [OGC 05-008].			

Requirements Class	
http://www.opengis.net/spec/ogpsearchgeo/m.n/req/req-class-b	
Target type	Token
Dependency	http://a9.com/-/opensearch
Requirement	<p>http://www.opengis.net/spec/ogpsearchgeo/1.0/req/req-class-b/geobox</p> <p>At a minimal, an OpenSearch GeoSpatial Server shall implement at least the capacity to support query by geo:box.</p>

The geo:geometry parameter **shall** support Well Know Text [WKT] describing the following 2D geometric objects:

- POINT
- LINESTRING
- POLYGON
- MULTIPOINT
- MULTILINESTRING
- MULTIPOLYGON

Note that the WKT coordinate pairs are in longitude, latitude order. Polygons are a collection of linear rings where the outer ring is expressed in counter-clockwise order. Internal rings have the opposite orientation, appearing as clockwise (OGC 06-103r4).

EXAMPLE 3 Example of geometries in WKT

```
POINT(6 10)
LINESTRING(3 4,10 50,20 25)
POLYGON((1 1,5 1,5 5,1 5,1 1),(2 2,2 3,3 3,3 2,2 2))
MULTIPOINT(3.5 5.6, 4.8 10.5)
MULTILINESTRING((3 4,10 50,20 25),(-5 -8,-10 -8,-15 -4))
MULTIPOLYGON(((1 1,5 1,5 5,1 5,1 1),(2 2,2 3,3 3,3 2,2 2)),((6
3,9 2,9 4,6 3)))
```

EXAMPLE 4 An example URL template supporting the geometry type

```
http://example.com/rss/?q={searchTerms}&pw={startPage?}&g={geo:geometry?}
```

EXAMPLE 5 Spaces must be URL encoded ('%20' or '+')

```
http://example.com/rss/?q=pizza&g=POLYGON((0.582%2040.496%2C%200.231
%2040.737%2C%200.736%2042.869%2C%203.351%2042.386%2C%203.263%2041.81
4%2C%202.164%2041.265%2C%200.978%20%20%2040.957%2C%200.802%2040.781%
2C%200.978%2040.649%2C%200.582%2040.496
```

The RFC 3339 was selected for the temporal values because it defines a profile of ISO 8601 commonly used in Internet protocols and standards that excludes durations and partial dates. The more complex formats such as week numbers and ordinal days are also not permitted. The start and end temporal parameters search for any item that intersects in any form the temporal range defined by them. The temporal characteristics of the searched items should focus on the actual data contents. When only one of the temporal queryables is present the OpenSearch GeoSpatial Server shall considered that client is requesting all items after a given start (when the end is missing) and conversely all items

before a given end when the start queryable is not present. According to the RFC-3339 the time:start and time:end parameters can contain complete date and time (e.g. 2003-01-17T20:03:05) or just the date value (e.g. 2003-01-17). In the case of date value without the time part, the server shall considered the temporal part of the date equal to 00:00:00 for both the start and end parameters.

Note that for the given key-value pairs, the key can be an arbitrary string, specified by one given instance of an OpenSearch repository. For example, one Description may provide a URL template asking for box={geo:box}, another specifying bbox={geo:box}. It is the responsibility of the client application to parse the URL template and create the appropriate keys for each key-value pair. These parameter sets are templates from which URLs can be constructed. The search client must replace every instance of a template parameter with a value before the search request is performed. If a search engine wishes to indicate that a template parameter is optional and can be replaced with the empty string, then the "?" notation described in the section on optional template parameters should be used. According to the OpenSearch specification (Annex D), all the "optional" parameters and data structures in the Search operation request **should** be implemented by all Search clients using the specified values for each implemented search to which that parameter or data structure applies. Similarly, all the "optional" parameters and data structures **should** be implemented by all OpenSearch GeoSpatial Servers, for each implemented Search to which that parameter or data structure applies.

All parameters of the OpenSearch query **should** be mapped to the appropriate catalogue fields. For example, the generic *searchTerms* parameter can be mapped to a given number of elements. The following table maps the Dublin Core and OGC queryable terms to the OpenSearch parameters.

Table 3 — Search operation queryable mappings

Opensearch Parameter type	Dublin Core element name	OGC queryable term	Atom Element
searchTerms	title	Title	atom:title
	description	AnyText	atom:summary
	subject	Abstract Subject	atom:category
geo:box geo:geometry geo:lat, geo:lon and geo:radius geo:relation geo:name	coverage	BoundingBox	georss:where
geo:uid	identifier	Identifier	atom:id
		Association	atom:link
time:start and time:end ^a			dc:date
^a The temporal queryables should be mapped to the intersection of the data content values			

9.2.2 Search request KVP encoding (mandatory)

Requirements Class	
http://www.opengis.net/spec/ogpsearchgeo/1.0/req/req-class-b	
Target type	Token
Requirement	<p>http://www.opengis.net/spec/ogpsearchgeo/1.0/req/req-class-b/http_get_transfer</p> <p>Servers shall implement HTTP GET transfer of the Search operation request, using KVP encoding.</p>
Requirement	<p>http://www.opengis.net/spec/ogpsearchgeo/1.0/req/req-class-b/url_parameters</p> <p>The KVP encoding of the Search operation request shall use the parameters specified in Table 4</p>
Requirement	<p>http://www.opengis.net/spec/ogpsearchgeo/1.0/req/req-class-b/search_parameters</p> <p>The parameters listed in Table 4 shall be as specified in Table 2 above.</p>

Table 4 — Search operation request URL parameters

Name and example ^a	Optionality and use	Definition and format
bbox={geo:box}	Optional	Rectangular bounding box
geom={geo:geometry}	Optional	Geometry in WKT
id={geo:uid}	Optional	Unique identifier of the record in the repository context
lat={geo:lat} & lon={geo:lon} & radius={geo:radius}	Optional	Centroid (latitude and longitude) and a search radius
rel={geo:relation}	Optional	Spatial relation (possible values are “overlaps”, “contains”, “disjoint”). The default is overlap
loc={geo:name}	Optional	Character string with a geographical name to be geocoded
startdate={time:start }	Optional	Character string with the start of the temporal interval according to RFC-3339.
stopdate={time:end}	Optional	Character string with the end of the temporal interval according to RFC-3339.

^a All parameter names are here listed using mostly lower case letters. However, any parameter name capitalization shall be allowed in KVP encoding, see Subclause 11.5.2 of [OGC 05-008].

EXAMPLE 6 An example Search operation request KVP encoded for HTTP GET is:

`http://foo.bar/foo?q=coverage&bbox=120,10,134,14`

9.3 Search operation response

9.3.1 Normal response parameters

The normal response to a valid Search operation request **shall** be in one of several formats that are specified in the OpenSearch Description for a given instance. The mandatory response format is ATOM but the server may support other formats.

9.3.2 Normal response XML encoding

The XML fragments shown in Appendix E specify the contents and structure of a Search operation response.

The normative response format for an OpenSearch service using Geospatial Extensions **shall** be in ATOM 1.0. The properties shown in the ATOM and OpenSearch namespaces are those mandated by the core specification.

Following the ATOM format the response is made of a *atom:feed* element that may contain several *atom:entry* elements (Figure 1). While latter describes the returned resources that respect the query performed the first describes the discovery service or search engine.

In the *atom:feed* element the response **should** include the elements listed in Table 5 and the *atom:entry* elements **should** include the elements listed in Table 6 (respecting the rules defined in [ATOM]). The metadata for each item in the result set **should** provide a link to a “full” view of the result, where more detailed metadata can be provided and, when available, a direct link to the resource using the *atom:link* element.

```
<?xml version="1.0" encoding="iso-8859-1"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:os="http://a9.com/~spec/opensearch/1.1/"
      xmlns:gml="http://www.opengis.net/gml"
      xmlns:georss="http://www.georss.org/georss">
  <title>
    Terradue Catalogue Feed for ENVI
  </title>
  <subtitle>Query Parameters used <
  <id>http://genesi-dec.local/catal
  <updated>2011-02-09T19:53:34Z</up
</entry>
  <entry>
    <title>
      ASA_WSM_1PNPDE20100414_054250_000
    </title>
    <id>
      ASA_WSM_1PNPDE20100414_054250_000
    </id>
    <updated>2010-09-08T10:07:25.948Z
    <published>2010-04-21T08:42:26.00
    <content type="html">
      &lt;b>Start &lt;/b>
      &lt;b>End &lt;/b>
      &lt;br/&lt;b>Acqui
      &lt;b>Processing Cente
      &lt;br/&lt;b>Orbi
    </content>
  </entry>
</feed>
```

Table 5
Elements of Search operation response in the *atom:feed* element describing the search service

Table 6
Elements of Search operation response in the *atom:entry* elements describing each search result

Figure 1 – Feed and entry elements in an ATOM response

Table 5 —Elements of Search operation response in the *atom:feed* element describing the search service

Term	OGC core returnable	Atom Element	Description	Mult.
Title	dc:title	atom:feed/ atom:title	A title for the search feed.	1 mandatory
Creator	dc:creator	atom:feed/ atom:author	An entity primarily responsible for making the content of the search feed	1 mandatory
Subject	dc:subject	atom:feed/ atom:category	A topic of the search feed. In this element where a Topic Category or other taxonomy could be applied	n optional
Abstract	dct:abstract	atom:feed/ atom:subtitle	An account of the content of the search feed	1 optional
Publisher	dc:publisher	atom:feed/ atom:generator	An entity or agent responsible for making the search feed.	1 optional
Contributor	dc:contributor	atom:feed/ atom:contributor	An entity responsible for making contributions to the content of the search feed	n optional
Modified	dc:date	atom:feed/ atom:updated	A date of a creation or update of the search feed (ISO-8601 date)	1 optional
Identifier	dc:identifier	atom:feed/ atom:id	An unambiguous reference to the identification of the search feed	1 mandatory
Source	dc:source	atom:feed/ atom:link[@atom:rel='search' and @atom:type= 'application/ opensearchdescripti on+xml']	A reference to a resource from which the present resource is derived. This points to the OpenSearch document that describes the search engine	1 optional
Language	dc:language	atom:feed/ @xml:lang	A language of resource's the content (RFC 3066)	1 optional
Envelope	dct:spatial	atom:feed/ georss:where	The maximal spatial extent of the search feed (GEORSS)	1 optional
Rights	dc:rights	atom:feed/ atom:rights	Information about rights held in and over the resource	1 optional
Relation (next page and previous pages)	dc:relation	atom:feed/atom:link [@atom:rel= 'next']	Reference to the next page of the search feed results	1 optional
		atom:feed/atom:link [@atom:rel= 'previous']	Reference to the previous page of the search feed results	1 optional
Query	-	atom:feed/ os:Query	OpenSearch element with query performed (see Annex D).	1 optional
Extent (total results, start index and items per page)	dct:extent	atom:feed/ os:totalResults	OpenSearch element with the number of search results	1 optional
		atom:feed/ os:startIndex	OpenSearch element with the index of the first search result.	1 optional
		atom:feed/ os:itemsPerPage	OpenSearch element with the number of search results returned per page.	1 optional

Table 6 —Elements of Search operation response in the *atom:entry* elements describing each search result

Term	OGC core returnable	Atom Element	Description	Mult.
Title	dc:title	atom:entry/ atom:title	A title given to the resource.	1 mandatory
Creator	dc:creator	atom:entry/ atom:author	An entity primarily responsible for making the content of the resource	1 optional
Subject	dc:subject	atom:entry/ atom:category	A topic of the content of the resource (a topic category or other taxonomy can be applied)	1 optional
Abstract	dct:abstract	atom:entry/ atom:summary	An account of the content of the resource	1 optional
Contributor	dc:contributor	atom:entry/ atom:contributor	An entity responsible for making contributions to the content of the resource	n optional
Modified	dc:date	atom:entry/ atom:updated	A date of a creation or update of the metadata resource (ISO-860)	1 optional
Date	-	atom:entry/ dc:date	A date or range of dates relevant to the resource (ISO-8601).	1 optional
Identifier	dc:identifier	atom:entry/ atom:id	An unambiguous reference to the identification of the resource within a given context	1 mandatory
Source	dc:source	atom:entry/ atom:link[@rel= 'via']	A reference to a resource from which the present resource is derived (e.g. metadata document from which the metadata of the resource is derived).	n optional
Language	dc:language	atom:entry/ @xml:lang	A language of the intellectual content of the resource as defined in RFC 3066	1 optional
Rights	dc:rights	atom:entry/ atom:rights	Information about rights held in and over the resource	1 optional
Envelope	dct:spatial	atom:entry/ georss:where	The spatial extent or scope of the content of the resource defined with an GEORSS element	1 optional
Relation	dc:relation	atom:entry/ atom:link[@atom:rel= 'search']	Reference to the OpenSearch description document when the resource is search service	1 optional
		atom:entry/ owc:Service	An OWC Service element if the resource is a OGC service	1 optional
		atom:entry/ atom:link[@rel= 'enclosure']	Reference to the location of the data resource described in the entry	n optional
		atom:entry/atom:link [@rel='icon']	Reference to a quicklook or browse image representing the entry	1 optional
		atom:entry/ atom:link[@rel= 'describedby']	Reference to a resource providing information or documentation about the entry's content	1 optional
Format	dc:format	atom:entry/atom:link [@rel='enclosure']/ @atom:type	The MIME types of the data resources available	n optional

The *atom:link* element with the *atom:rel* attribute equal to “alternate” **should** be used to link the entry to additional representations of the metadata. The type of the metadata **should** be defined by the *atom:type* attribute. To designate the resource that is the source of the information provided in the containing element (e.g. original metadata) the value “via” **should** be used [RFC 5988] [ATOM] (section 4.2.7.2).

EXAMPLE 7 Defining an alternate representation in HTML for the metadata in a *atom:entry*

```
<atom:link atom:rel="alternate" atom:type="text/html"
atom:title="Product metadata entry"
atom:href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1
PNPDK20090122_205935_000002082075_00444_36069_6663.N1/html"/>
```

Please note that according to [ATOM] (section 4.1.2) the *atom:link* element with the *rel* attribute equal to “alternate” must follow some rules like:

- *atom:entry elements that contain no child atom:content element MUST contain at least one atom:link element with a rel attribute value of "alternate".*
- *atom:entry elements MUST NOT contain more than one atom:link element with a rel attribute value of "alternate" that has the same combination of type and hreflang attribute values*

If possible, the access to the file or service that contains the data **should** be defined using a *atom:link* element with the *atom:rel* attribute equal to “enclosure”[RFC 5988]. To define the reference to a resource representing the quicklook or browse image that represents the entry, the *atom:rel* attribute should be equal to “icon”.

EXAMPLE 8 Defining the access to the product file of the *atom:entry*

```
<atom:link atom:rel="enclosure" atom:href="https://grid-eo-
engine02.esrin.esa.int/ASA_WSM_1P/2009/01/ASA_WSM_1PNPDK20090122_205935
_000002082075_00444_36069_6663.N1" atom:type="application/binary"/>
```

EXAMPLE 9 Defining the access to quicklook or browse image of the *atom:entry*

```
<atom:link
href="ftp://ladsftp.nascom.nasa.gov/allData/5/MOBRGB/2009/121/MOBRGB.A2
009121.0020.005.2009126214738.jpg" hreflang="en-US" type="image/jpeg"
title="" rel="icon" length="660988"></link>
```

If the resource described in the response entry is itself a search service an *atom:link* with the relation “search” **should** be used [RFC 5988].

EXAMPLE 10 Defining access to the product file of the *atom:entry*

```
<atom:link atom:rel="search" title="Search this resource"
type="application/opensearchdescription+xml"
href="http://search.acme.com/opensearch.xml" >
```

Furthermore according to the ATOM documentation [ATOM] (section 4.1.3.3), special conditions have to be considered when using the *atom:content* and *atom:summary* with the *atom:type* equal to "html". For example:

*If the value of "type" is "html", the content of atom:content MUST NOT contain child elements and SHOULD be suitable for handling as HTML [HTML]. The HTML markup MUST be escaped; for example, "
" as "
". The HTML markup SHOULD be such that it could validly appear directly within an HTML <DIV> element. Atom Processors that display the content MAY use the markup to aid in displaying it.*

EXAMPLE 11 HTML elements inside the atom:summary must be escaped [ATOM]

```
<summary type="html">
  &lt;b&gt; Start :&lt;/b&gt; 2010-04-13T20:38:16.000Z
  &lt;b&gt; End :&lt;/b&gt; 2010-04-13T20:40:00.000Z
  &lt;br/&gt;&lt;b&gt;Acquisition Station :&lt;/b&gt; PDAS-M
  &lt;b&gt;Processing Center :&lt;/b&gt; I-PAC
  &lt;br/&gt;&lt;b&gt;Orbit :&lt;/b&gt; 42453
</summary>
```

The *atom:summary* element usually takes precedence to *atom:content* element when client application display the entries listing. It's however recommended that a OpenSearch Geospatial Service does not use the *atom:content* element to expand or define new metadata elements. If such a feature is required one should instead add directly the required element under the *atom:feed* or *atom:entry* elements with the respective namespace.

EXAMPLE 12 Extending the response entry with elements from a different namespace

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:georss="http://www.georss.org/georss"
  xmlns:gml="http://www.opengis.net/gml"
  xmlns:owc="http://www.opengis.net/owc">
<!-- ... -->
  <owc:General>
    <owc:Version>0.7.0a</owc:Version>
  </owc:General>

  <entry>
<!-- ... -->
    <owc:Service type="urn:ogc:serviceType:WebFeatureService">
      <owc:Name
xmlns:owc="http://www.cubewerx.com/owc">owc:POPPLACE</owc:Name>
      <owc:ServiceVersion>1.1.0</owc:ServiceVersion>
      <owc:OnlineResource
xlink:href="http://portal.cubewerx.com/cubewerx/cubeserv/cubeserv.cgi?c
onfig=haiti&datastore=MINUSTAH"/>
      <owc:Hidden>>false</owc:Hidden>
    </owc:Service>
  </entry>
```


For easier client integration it is recommended to support result paging. To achieve this the *atom:feed* element **should** include *atom:link* elements with the attribute *atom:rel* equal to “self”, “prev” (or its synonym “previous”) and “next” [RFC 5988].

To support the execution of additional searches the *atom:feed* element **should** include *atom:link* elements with the attribute *atom:rel* equal to “search” [RFC 5988]. This element refers to the OpenSearch description document of the search engine that created the feed.

EXAMPLE 13 Including paging support on the ATOM response (XML fragment)

```
<feed xmlns=http://www.w3.org/2005/Atom
      xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
      xmlns:georss="http://www.georss.org/georss">
  <title>Terradue Catalogue ENVISAT ASAR Wide Swath Mode</title>
  <subtitle>Query Parameters used bbox=10,10,20,20</subtitle>
  <link rel="self" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startInd
ex=0&bbox=10,10,20,20" title="self"/>
  <link rel="next" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startInd
ex=20&bbox=10,10,20,20" title="next">
  <link rel="search" type="application/opensearchdescription+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/description"
title="search"/>
  <os:Query os:role="request" count="2" geo:box="10,10,20,20"/>
  <os:totalResults>782</os:totalResults>
  <os:startIndex>0</os:startIndex>
  <os:itemsPerPage>20</os:itemsPerPage>

<!-- ... -->
```

Also shown in the previous example is the usage of the OpenSearch elements (*os:totalResults*, *os:startIndex*, *os:itemsPerPage* and *os:Query* described in Annex D) that **should** be included in the response.

9.3.3 ATOM 1.0 response example

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
      xmlns:georss="http://www.georss.org/georss">
  <title>Example.com Search: New York history</title>
  <link href="http://example.com/New+York+history"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>Example.com, Inc.</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
  <opensearch:totalResults>4230000</opensearch:totalResults>
  <opensearch:startIndex>21</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
```

```

    <opensearch:Query role="request" searchTerms="New York History"
startPage="3" geo:box="-74.0667,40.69418,-73.9116,40.7722"/>
    <link rel="alternate"
href="http://example.com/New+York+History?pw=3&bbox=-74.0667,40.69418,-
73.9116,40.7722" type="text/html"/>
    <link rel="search" type="application/opensearchdescription+xml"
href="http://example.com/opensearchdescription.xml"/>
    <georss:box>40.69418 -74.0667 40.7722 -73.9116</georss:box>
    <entry>
      <title>New York History</title>
      <link
href="http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html"/>
      <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
      <updated>2003-12-13T18:30:02Z</updated>
      <georss:line>40.73763 -73.9972 40.73519 -73.99167 40.737015 -
73.99035 40.73643 -73.98914 40.734640 -73.990431 40.731617 -
73.991504</georss:line>
      <content type="text">
        ... Union Square.NYC - A virtual tour and information on
        businesses ...with historic photos of Columbia's own New York
        neighborhood ... Internet Resources for the City's History...
      </content>
    </entry>
  </feed>

```

9.3.4 ATOM 1.0 response example for Earth Observation Imagery

```

<?xml version="1.0" encoding="iso-8859-1"?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:os="http://a9.com/-/spec/opensearch/1.1/">
  <title>Feed for ENVISAT ASAR Wide Swath Mode (ASA_WSM_1P)</title>
  <link rel="self" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startInd
ex=0 " title="self"/>
  <link rel="first" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startInd
ex=0" title="first"/>
  <link rel="next" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startInd
ex=20" title="next"/>
  <link rel="last" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startInd
ex=900" title="last"/ >
  <link rel="search" type="application/opensearchdescription+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/description"
title="search"/>
  <os:totalResults>913</os:totalResults>
  <os:startIndex>0</os:startIndex>
  <os:itemsPerPage>20</os:itemsPerPage>
  <id>http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom</id>
  <updated>2010-09-19T17:44:17Z</updated>
  <generator uri="http://maps.terradue.com/catalogue/">
    Terradue GeoSpatial Search Team
  </generator>
  <author><name>Terradue</name></author>
<entry>

```

```

<title>
  ASA_WSM_1PNIPA20100413_092526_000001652088_00308_42446_1340
</title>
<id>
  http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PNIPA201
  00413_092526_000001652088_00308_42446_1340.N1
</id>
<updated>2010-09-08T10:07:25.777Z</updated>
<published>2010-04-21T08:32:51.000Z</published>
<dc:date>
  2010-04-13T09:25:26.000Z/2010-04-13T09:28:11.000Z</dc:date>
  <link rel="via" type="application/xhtml+xml" title="XML metadata
  entry"
  href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PN
  IPA20100413_092526_000001652088_00308_42446_1340.N1/xml"/>
  <link rel="alternate" type="text/html" title="Product metadata
  entry"
  href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PN
  IPA20100413_092526_000001652088_00308_42446_1340.N1/html"/>
  <summary type="html">
    <b>Start :</b> 2010-04-13T09:25:26.000Z
    <b>End :</b> 2010-04-13T09:28:11.000Z
    <br/><b>Acquisition Station :</b> PDAS-M
    <b>Processing Center :</b> I-PAC
    <br/><b>Orbit :</b> 42446
  </summary>
  <georss:polygon>
    11.2469 28.7937 6.99024 29.5309 5.0928 20.13 9.03964 19.3724
    11.2469 28.7937
  </georss:polygon>
  <link rel="enclosure" href="https://grid-eo-
  engine02.esrin.esa.int/gpfs/gpfs01/eo/ASA_WSM_1P/2010/04/ASA_WSM_1PN
  IPA20100413_092526_000001652088_00308_42446_1340.N1"
  type="application/binary"/>

</entry>

[...]
</feed>

```

The complete listing of the examples can be seen in Annex E.

Annex A (normative)

Abstract test suite

A.1 OpenSearch description document

Test Purpose: Verify that the server generates a valid OpenSearch description document.

Test Method: Execute an HTTP GET request to retrieve a server's description document. Verify that the response is correct XML and is valid according to the description in this standard.

References: Clause 8, Annex D

Test type: Capability

A.2 ATOM response type

Test Purpose: Verify that the server defines an URL template for the ATOM response type.

Test Method: Execute an HTTP GET request to retrieve a server's description document. Verify that an URL element with the type equal to “application/atom+xml” exists.

References: 9.1

Test type: Capability

A.3 Bounding box search

Test Purpose: Verify that the server conforms to the Bounding Box Search conformance class.

Test Method: Submit a series of bounding box requests to the server and verify that the server responds as described in the standard. The set of requests should include both valid and invalid requests to test for exceptions.

References: 9.2.1, Sub-clause 9.3

Test type: Capability

A.4 Arbitrary geometry search

Test Purpose: Verify that the server conforms to the Arbitrary Geometry Search conformance class.

Test Method: Submit a series of queries to the server using the geometry parameter. The geometry values should be encoded using WKT. Verify that the server responds as described in the standard. The set of requests should include both valid and invalid requests to test for exceptions.

References: 9.2.1, Sub-clause 9.3

Test type: Capability

A.5 Point and radius search

Test Purpose: Verify that the server conforms to the Point and Radius Search conformance class.

Test Method: Submit a series of queries to the server using the lat, lon and radius parameters. Verify that the server responds as described in the standard. The set of requests should include both valid and invalid requests to test for exceptions.

References: 9.2.1, Sub-clause 9.3

Test type: Capability

A.6 Min spatial operator

Test Purpose: Verify that the server conforms to the Min Spatial Operator conformance class.

Test Method: Submit a series of requests to the server and verify that all records in the response satisfy the “overlaps” operator.

References: 9.2.1,

Test type: Capability

A.7 Spatial operators

Test Purpose: Verify that the server conforms to the Spatial Operators conformance class.

Test Method: Submit a series of requests to the server and verify that the spatial information of the records in the response is coded in GeoRSS and

satisfy the spatial operator specified in the request.

References: 9.1, 9.2.1

Test type: Capability

A.8 Get record by id

Test Purpose: Verify that the server conforms to the Get record by id conformance class.

Test Method: Submit a series of request to the server that retrieve records using the records identifier. Verify that the id of the record in the response corresponds to the requested id.

References: 9.2.1

Test type: Capability

A.9 Search by name

Test Purpose: Verify that the server conforms to the Search by name conformance class.

Test Method: Submit a series of requests to the server that search using the name or address of a location. Verify that the records in the response correspond to the specified name or address.

References: 9.2.1,

Test type: Capability

A.10 Temporal search

Test Purpose: Verify that the server conforms to the Temporal Search conformance class.

Test Method: Submit a series of request to the server using the start and end parameters. Verify that the temporal property of the records in the response falls between the specified start and end times.

References: 9.2.1,

Test type: Capability

Annex B
(normative)

XML Schema Documents

Not Applicable.

Note: Schemas are defined in [ATOM] for the ATOM response format and in [OASIS OPENSEARCH] for the OpenSearch Description Document.

Annex C
(informative)

UML model

Annex D (informative)

OpenSearch 1.1 Draft 4

D.1 Introduction

The contents of the annex originated from [opensearch.org](http://www.opensearch.org) web site (http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_4) and made available by A9.com subject to the terms of the Creative Commons Attribution-ShareAlike 2.5 License. The author of this specification is DeWitt Clinton with the collaboration of Joel Tesler, Michael Fagan, Joe Gregorio, Aaron Sauve and James Snell.

This annex defines the OpenSearch description document, the OpenSearch Query element, the OpenSearch URL template syntax, and the OpenSearch response elements. Collectively these formats may be referred to as "OpenSearch 1.1" or simply "OpenSearch".

Search clients can use OpenSearch description documents to learn about the public interface of a search engine. These description documents contain parameterized URL templates that indicate how the search client should make search requests. Search engines can use the OpenSearch response elements to add search metadata to results in a variety of content formats.

Namespace

The XML Namespaces URI for the XML data formats described in this specification is:

```
http://a9.com/-/spec/opensearch/1.1/
```

OpenSearch description document

An OpenSearch description document can be used to describe the web interface of a search engine.

- Type

OpenSearch description documents are referred to via the following type:

```
application/opensearchdescription+xml
```

This type is pending IANA registration.

Extensibility

OpenSearch description documents can be extended with foreign markup provided that all foreign elements and attributes are associated with an explicit XML namespace distinct from that of the core OpenSearch format. When possible, the foreign XML namespace URI should resolve to a document that indicates the intention and format of the extension. Clients that encounter unrecognized foreign markup should continue to process the document as if the markup did not appear.

OpenSearch description elements

- OpenSearchDescription

The root node of the OpenSearch description document.

Parent: None

Requirements: The element must appear exactly once as the root node of the document.

- ShortName

Contains a brief human-readable title that identifies this search engine.

Parent: OpenSearchDescription

Restrictions: The value must contain 16 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This element must appear exactly once.

- Description

Contains a human-readable text description of the search engine.

Parent: OpenSearchDescription

Restrictions: The value must contain 1024 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This element must appear exactly once.

- Url

Describes an interface by which a client can make requests for an external resource, such as search results, search suggestions, or additional description documents.

Parent: OpenSearchDescription

Attributes:

template - The URL template to be processed according to the OpenSearch URL template syntax.

Requirements: This attribute is required.

type - The MIME type of the resource being described.

Restrictions: The value must be a valid MIME type.

Requirements: This attribute is required.

rel - The role of the resource being described in relation to the description document.

Restrictions: Contains a space-delimited list of valid rel value tokens. See the Url rel values specification for allowed rel values.

Default: "results"

Requirements: This attribute is optional.

indexOffset - The index number of the first search result.

Restrictions: The value must be an integer.

Default: "1"

Requirements: This attribute is optional.

pageOffset - The page number of the first set of search results.

Restrictions: The value must be an integer.

Default: "1"

Requirements: This attribute is optional.

Requirements: This element must appear one or more times.

Url rel values

Rel attribute strings can contain a space-delimited list of one or more semantically meaningful rel value tokens. An empty rel attribute value should be treated by the client as if the rel attribute was not present at all.

If a client does not recognize the semantic meaning of any rel value token, then the containing Url should be ignored by the client.

Rel value tokens may be either fully qualified tokens (e.g., "http://example.com/rel#foo") or unqualified tokens (e.g., "results").

All fully qualified tokens must be a valid URL. The semantic meaning of any fully qualified token is outside the scope of this specification, but convention dictates that the URL should resolve to a resource that describes the relationship.

All unqualified tokens must be a lowercase alphanumeric string of the form [a-z][a-z\-\-]+. Only those tokens listed below have meaning defined in this specification.

Rel values:

"results" (default)

Represents a request for search results in the specified format.

"suggestions"

Represents a request for search suggestions in the specified format. See the OpenSearch Suggestions extension for further details.

"self"

Represents the canonical URL of this description document.

"collection"

Represents a request for a set of resources.

- Contact

Contains an email address at which the maintainer of the description document can be reached.

Parent: OpenSearchDescription

Restrictions: The value must conform to the requirements of Section 3.4.1 "Addr-spec specification" in RFC 2822.

Requirements: This element may appear zero or one time.

- Tags

Contains a set of words that are used as keywords to identify and categorize this search content. Tags must be a single word and are delimited by the space character (' ').

Parent: OpenSearchDescription

Restrictions: The value must contain 256 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This element may appear zero or one time.

- LongName

Contains an extended human-readable title that identifies this search engine.

Search clients should use the value of the ShortName element if this element is not available.

Parent: OpenSearchDescription

Restrictions: The value must contain 48 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This element may appear zero or one time.

- Image

Contains a URL that identifies the location of an image that can be used in association with this search content.

Image sizes are offered as a hint to the search client. The search client will choose the most appropriate image for the available space and should give preference to those listed first in the OpenSearch description document. Square aspect ratios are recommended. When possible, search engines should offer a 16x16 image of type "image/x-icon" or "image/vnd.microsoft.icon" (the Microsoft ICON format) and a 64x64 image of type "image/jpeg" or "image/png".

Parent: OpenSearchDescription

Restrictions: The value must be a URI.

Requirements: This element may appear zero, one, or more times.

Attributes:

height – Contains the height, in pixels, of this image.

Restrictions: The value must be a non-negative integer.

Requirements: This attribute is optional.

width – Contains the width, in pixels, of this image.

Restrictions: The value must be a non-negative integer.

Requirements: This attribute is optional.

type – Contains the the MIME type of this image.

Restrictions: The value must be a valid MIME type.

Requirements: This attribute is optional.

- *Query*

Defines a search query that can be performed by search clients. Please see the OpenSearch Query element specification for more information.

OpenSearch description documents should include at least one Query element of role="example" that is expected to return search results. Search clients may use this example query to validate that the search engine is working properly.

Parent: OpenSearchDescription

Requirements: This element may appear zero or more times.

- *Developer*

Contains the human-readable name or identifier of the creator or maintainer of the description document.

The developer is the person or entity that created the description document, and may or may not be the owner, author, or copyright holder of the source of the content itself.

Parent: OpenSearchDescription

Restrictions: The value must contain 64 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This element may appear zero or one time.

- *Attribution*

Contains a list of all sources or entities that should be credited for the content contained in the search feed.

Parent: OpenSearchDescription

Restrictions: The value must contain 256 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This element may appear zero or one time.

- *SyndicationRight*

Contains a value that indicates the degree to which the search results provided by this search engine can be queried, displayed, and redistributed.

Parent: OpenSearchDescription

Values: The value must be one of the following strings (case insensitive):

"open" –

The search client may request search results.

The search client may display the search results to end users.

The search client may send the search results to other search clients.

"limited" –

The search client may request search results.

The search client may display the search results to end users.

The search client may not send the search results to other search clients.

"private" –

The search client may request search results.

The search client may not display the search results to end users.

The search client may not send the search results to other search clients.

"closed" -

The search client may not request search results.

Default: "open"

Requirements: This element may appear zero or one time.

- AdultContent

Contains a boolean value that should be set to true if the search results may contain material intended only for adults.

As there are no universally applicable guidelines as to what constitutes "adult" content, the search engine should make a good faith effort to indicate when there is a possibility that search results may contain material inappropriate for all audiences.

Parent: OpenSearchDescription

Values:

The values "false", "FALSE", "0", "no", and "NO" will be considered boolean FALSE; all other strings will be considered boolean TRUE.

Default: "false"

Requirements: This element may appear zero or one time.

- *Language*

Contains a string that indicates that the search engine supports search results in the specified language.

An OpenSearch description document should include one "Language" element for each language that the search engine supports. If the search engine also supports queries for any arbitrary language then the OpenSearch description document should include a Language element with a value of "*". The "language" template parameter in the OpenSearch URL template can be used to allow the search client to choose among the available languages.

Parent: OpenSearchDescription

Restrictions: The value must conform to the XML 1.0 Language Identification, as specified by RFC 3066. In addition, the value of "*" will signify that the search engine does not restrict search results to any particular language.

Default: "*".

Requirements: This element may appear zero, one, or more times.

- *InputEncoding*

Contains a string that indicates that the search engine supports search requests encoded with the specified character encoding.

An OpenSearch description document should include one "InputEncoding" element for each character encoding that can be used to encode search requests. The "inputEncoding" template parameter in the OpenSearch URL template can be used to require the search client to identify which encoding is being used to encode the current search request.

Parent: OpenSearchDescription

Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments.

Default: "UTF-8".

Requirements: This element may appear zero, one, or more times.

- *OutputEncoding*

Contains a string that indicates that the search engine supports search responses encoded with the specified character encoding.

An OpenSearch description document should include one "OutputEncoding" element for each character encoding that can be used to encode search responses. The "outputEncoding" template parameter in the OpenSearch URL template can be used to allow the search client to choose a character encoding in the search response.

Parent: OpenSearchDescription

Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments.

Default: "UTF-8".

Requirements: This element may appear zero, one, or more times.

Autodiscovery

Search engines that publish OpenSearch description documents can assist search clients in the discovery of OpenSearch interfaces through the use of "link" elements. Search engines that support OpenSearch should include a reference to the related OpenSearch description document on each page of search results.

Autodiscovery in RSS/Atom

RSS and Atom documents may reference related OpenSearch description documents via the Atom 1.0 "link" element, as specified in Section 4.2.7 of RFC 4287.

The "rel" attribute of the link element should contain the value "search" when referring to OpenSearch description documents. This relationship value is pending IANA registration. The reuse of the Atom link element is recommended in the context of other syndication formats that do natively support comparable functionality.

The following restrictions apply:

- The "type" attribute must contain the value "application/opensearchdescription+xml".
- The "rel" attribute must contain the value "search".
- The "href" attribute must contain a URI that resolves to an OpenSearch description document.
- The "title" attribute may contain a human-readable plain text string describing the search engine.

Example of Atom-based search results that include an OpenSearch autodiscovery link element:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
```

```

<!-- ... --->
<link rel="search"
      href="http://example.com/opensearchdescription.xml"
      type="application/opensearchdescription+xml"
      title="Content Search" />
<!-- ... --->
</feed>

```

Example of RSS-based search results that include an OpenSearch autodiscovery link element:

```

<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
      xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <!-- ... --->
    <atom:link rel="search"
              href="http://example.com/opensearchdescription.xml"
              type="application/opensearchdescription+xml"
              title="Content Search" />
    <!-- ... --->
  </channel>
</rss>

```

Autodiscovery in HTML/XHTML

HTML and XHTML documents may reference related OpenSearch description documents via the HTML 4.0 <link/> element.

The following restrictions apply:

- The "type" attribute must contain the value "application/opensearchdescription+xml".
- The "rel" attribute must contain the value "search".
- The "href" attribute must contain a URI that resolves to an OpenSearch description document.
- The "title" attribute may contain a human-readable plain text string describing the search engine.
- The HTML <head/> element should include a "profile" attribute that contains the value "http://a9.com/-/spec/opensearch/1.1/".

Example of an HTML document that includes OpenSearch autodiscovery link elements:

```

<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
"http://www.w3.org/TR/html4/strict.dtd">

```

```

<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en"
dir="ltr">
  <head profile="http://a9.com/-/spec/opensearch/1.1/">
    <!-- ... -->
    <link rel="search"
          type="application/opensearchdescription+xml"
          href="http://example.com/content-search.xml"
          title="Content search" />
    <link rel="search"
          type="application/opensearchdescription+xml"
          href="http://example.com/comment-search.xml"
          title="Comments search" />
    <!-- ... -->
  </head>
  <body>
    <!-- ... -->
  </body>
</html>

```

OpenSearch URL template syntax

The OpenSearch URL template format can be used to represent a parameterized form of the URL by which a search engine is queried.

The search client will process the URL template and attempt to replace each instance of a template parameter, generally represented in the form {name}, with a value determined at query time.

By default, parameter names are considered part of the OpenSearch 1.1 template namespace, and definitions for a set of core search parameter names are provided in this specification. However, search engines and search clients can adopt new parameter names using an extensibility mechanism based on the XML namespace prefix conventions.

Example of a search URL template that contains a template parameter:

```
http://example.com/search?q={searchTerms}
```

Example of a search URL template that contains an optional template parameter:

```
http://example.com/feed/{startPage?}
```

Example of a search URL template that contains an optional template parameter in an extended namespace, shown in the context of a `Url` element:

```

<Url type="application/rss+xml"
      xmlns:example="http://example.com/opensearchextensions/1.0/"
      template="http://example.com?q={searchTerms}&c={example:color?}"/>

```

This specification refers to the use of the OpenSearch URL template syntax specifically within the context of the "Url" element in an OpenSearch description document.

Template grammar

The grammar of an OpenSearch URL template is defined by the following set of ABNF rules, as specified in RFC 2234.

The grammar rules defined in this document build upon a subset of the rules defined for the Uniform Resource Identifier (URI): Generic Syntax in RFC 3986. For brevity, rules already stated in RFC 3986 are referenced in this document by rule name alone and are not restated here in their entirety.

```

ttemplate      = tscheme ":" thier-part [ "?" tquery ] [ "#" tfragment
]
tscheme        = *( scheme / tparameter )
thier-part     = "//" tauthority ( tpath-abempty / tpath-absolute /
tpath-rootless / path-empty )
tauthority     = [ tuserinfo "@" ] thost [ ":" tport ]
tuserinfo       = *( userinfo / tparameter )
thost          = *( host / tparameter )
tport          = *( port / tparameter )
tpath-abempty  = *( "/" tsegment )
tsegment       = *( segment / tparameter )
tpath-absolute = "/" [ tsegment-nz *( "/" tsegment ) ]
tsegment-nz    = *( segment-nz / tparameter )
tpath-rootless = tsegment-nz *( "/" tsegment )
tparameter    = "{" tqname [ tmodifier ]}"
tqname         = [ tprefix ":" ] tlname
tprefix        = *pchar
tlname         = *pchar
tmodifier      = "?"
tquery         = *( query / tparameter )
tfragment     = *( fragment / tparameter )

```

Substitution rules

The search client must replace every instance of a template parameter with a value before the search request is performed.

If a search engine wishes to indicate that a template parameter is optional and can be replaced with the empty string, then the "?" notation described in the section on optional template parameters should be used.

Parameter names

A parameter name consists of an optional parameter name prefix followed by the local parameter name. If the parameter name prefix is present then it will be separated from the local parameter name with the ":" character. All parameter names are associated with a

parameter namespace. In the case of unqualified parameter names, the local parameter name is implicitly associated with the OpenSearch 1.1 namespace. In the case of fully qualified parameter names, the local parameter name is explicitly associated with an external namespace via the parameter name prefix.

Case sensitivity of parameter names

Both the parameter name prefix and the local parameter name are case sensitive.

Parameter name prefix

A parameter name prefix associates a local parameter name with a parameter namespace. All parameter name prefixes must be previously declared as an XML namespace prefix on the containing element or ancestor elements.

The choice of prefix is at the discretion of the author of the OpenSearch description document. Search clients should make no assumption as to the meaning of any particular literal prefix string, and should rely exclusively on the mapping of prefix strings to XML namespace declarations when parsing fully qualified parameter names.

Example of two equivalent URL templates that will be processed identically by search clients:

```
<Url type="application/rss+xml"
      xmlns:a="http://example.com/extensions/"
      template="http://example.com?q={a:localname?}"/>

<Url type="application/rss+xml"
      xmlns:b="http://example.com/extensions/"
      template="http://example.com?q={b:localname?}"/>
```

Unqualified parameter names

Unqualified parameter names consist of only a local parameter name and do not include a parameter name prefix. Unqualified parameter names in OpenSearch URL templates are implicitly associated with the OpenSearch 1.1 namespace.

This specification includes an exhaustive list of all unqualified OpenSearch 1.1 parameter names.

Example of an unqualified parameter name:

```
<Url type="application/rss+xml"
      template="http://example.com/?q={searchTerms}"/>
```

Fully qualified parameter names

Fully qualified parameter names consist of a parameter name prefix, followed by the ":" character, followed by the local parameter name. Fully qualified parameter names are

associated with the namespace identified by the parameter name prefix, as it appears as an XML namespace declaration on the containing element or ancestor elements.

Example of a fully qualified parameter name:

```
<Url type="application/rss+xml"
      xmlns:example="http://example.com/opensearchextensions/1.0/"
      template="http://example.com?f={example:format?}"/>
```

Required template parameters

Required template parameters are template parameters that do not contain a template parameter modifier. The search client may use the default value if one is known, but may not use the empty string as a value.

Example of a required template parameter:

```
{searchTerms}
```

Optional template parameters

Optional template parameters are template parameters that contain a template parameter modifier equal to "?". The search client may use the empty string as a value if no other value is available.

Example of an optional template parameter:

```
{startPage?}
```

OpenSearch 1.1 parameters

The following local parameter names are identified with the OpenSearch 1.1 namespace. The list is exhaustive; only the local parameter names listed below may appear unqualified in an OpenSearch URL template.

Search clients should be prepared to substitute reasonable values for these parameter names when they appear in an OpenSearch URL template.

The "searchTerms" parameter

Replaced with the keyword or keywords desired by the search client.

Restrictions: The value must be URL-encoded.

The "count" parameter

Replaced with the number of search results per page desired by the search client.

Search clients should anticipate that the value of the "count" parameter may not be honored by the search engine, and should rely exclusively on the contents of the "itemsPerPage" response element in calculating actual page size.

Restrictions: The value must be a non-negative integer.

The "startIndex" parameter

Replaced with the index of the first search result desired by the search client.

Restrictions: The value must be an integer.

Default: The value specified by the "indexOffset" attribute of the containing Url element.

The "startPage" parameter

Replaced with the page number of the set of search results desired by the search client.

Restrictions: The value must be an integer.

Default: The value specified by the "pageOffset" attribute of the containing Url element.

The "language" parameter

Replaced with a string that indicates that the search client desires search results in the specified language.

An OpenSearch description document should include one "Language" element for each language that the search engine supports. If the search engine also supports queries for any arbitrary language then the OpenSearch description document should include a Language element with a value of "*". The "language" template parameter in the OpenSearch URL template can be used to allow the search client to choose among the available languages.

Restrictions: The value must conform to the XML 1.0 Language Identification, as specified by RFC 3066. In addition, a value of "*" will signify that the search client desires search results in any language.

Default: "*"

The "inputEncoding" parameter

Replaced with a string that indicates that the search client is performing the search request encoded with the specified character encoding.

An OpenSearch description document should include one "InputEncoding" element for each character encoding that can be used to encode search requests. The "inputEncoding"

template parameter in the OpenSearch URL template can be used to require the search client to identify which encoding is being used to encode the current search request.

Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments.

Default: "UTF-8"

The "outputEncoding" parameter

Replaced with a string that indicates that the search client desires a search response encoding with the specified character encoding.

An OpenSearch description document should include one "OutputEncoding" element for each character encoding that can be used to encode search responses. The "outputEncoding" template parameter in the OpenSearch URL template can be used to allow the search client to choose a character encoding in the search response.

Restrictions: The value must conform to the XML 1.0 Character Encodings, as specified by the IANA Character Set Assignments.

Default: "UTF-8"

OpenSearch Query element

The OpenSearch Query element can be used to define a specific search request that can be performed by a search client.

The Query element attributes correspond to the search parameters in a URL template. The core set of search parameters are explicitly defined as Query attributes, and custom parameters can be added via namespaces as needed.

Authors should provide at least one Query element of role="example" in each OpenSearch description document so that search clients can test the search engine. Search engines should include a Query element of role="request" in each search response so that search clients can recreate the current search.

Example of a Query element used in an OpenSearch description document to provide an example search request for search clients:

```
<Query role="example" searchTerms="cat" />
```

Example of a Query element used in an OpenSearch response to echo back the original search request:

```
<Query role="request" searchTerms="cat" startPage="1" />
```


Example of a Query element used in an OpenSearch response to correct the spelling of "OpenSurch":

```
<Query role="correction" searchTerms="OpenSearch"
totalResults="854000" title="Spelling correction" />
```

Example of a Query element using an extended parameter:

```
<Query xmlns:custom="http://example.com/opensearchextensions/1.0/"
role="example"
searchTerms="cat"
custom:color="blue"
title="Sample search" />
```

Example of a Query element using a extended role:

```
<Query xmlns:custom="http://example.com/opensearchextensions/1.0/"
role="custom:synonym"
title="Synonym of 'cat'"
searchTerms="feline" />
```

Detailed example of a set of Query elements used in the context of an Atom-based OpenSearch response:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <!-- ... -->
  <opensearch:Query role="request" searchTerms="General Motors annual
report" />
  <opensearch:Query role="related" searchTerms="GM" title="General
Motors stock symbol" />
  <opensearch:Query role="related" searchTerms="automotive industry
revenue" />
  <opensearch:Query role="subset" searchTerms="General Motors annual
report 2005" />
  <opensearch:Query role="superset" searchTerms="General Motors" />
  <!-- ... -->
</feed>
```

The "Query" element

Describes a specific search request that can be made by the search client.

Attributes:

role - Contains a string identifying how the search client should interpret the search request defined by this Query element.

Restrictions: See the role values specification for allowed role values.

Requirements: This attribute is required.

title - Contains a human-readable plain text string describing the search request.

Restrictions: The value must contain 256 or fewer characters of plain text. The value must not contain HTML or other markup.

Requirements: This attribute is optional.

totalResults - Contains the expected number of results to be found if the search request were made.

Restrictions: The value is a non-negative integer.

Requirements: This attribute is optional.

searchTerms - Contains the value representing the "searchTerms" as an OpenSearch 1.1 parameter.

Restrictions: See the "searchTerms" parameter.

Requirements: This attribute is optional.

count - Contains the value representing the "count" as a OpenSearch 1.1 parameter.

Restrictions: See the "count" parameter.

Requirements: This attribute is optional.

startIndex - Contains the value representing the "startIndex" as an OpenSearch 1.1 parameter.

Restrictions: See the "startIndex" parameter.

Requirements: This attribute is optional.

startPage - Contains the value representing the "startPage" as an OpenSearch 1.1 parameter.

Restrictions: See the "startPage" parameter.

Requirements: This attribute is optional.

language - Contains the value representing the "language" as an OpenSearch 1.1 parameter.

Restrictions: See the "language" parameter.

Requirements: This attribute is optional.

inputEncoding - Contains the value representing the "inputEncoding" as an OpenSearch 1.1 parameter.

Restrictions: See the "inputEncoding" parameter.

Requirements: This attribute is optional.

outputEncoding - Contains the value representing the "outputEncoding" as an OpenSearch 1.1 parameter.

Restrictions: See the "outputEncoding" parameter.

Requirements: This attribute is optional.

Example:

```
<Query role="example" searchTerms="cat" />
```

Query element extensibility

The Query element may contain additional attributes if the extended attributes are associated with a namespace. Search clients should interpret extended attributes to represent the corresponding template parameter by the same name in the specified namespace.

Example of a Query element representing a search request that contains an extended attribute that corresponds to an extended search parameter:

```
<OpenSearchDescription xmlns="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:custom="http://example.com/opensearchextensions/1.0/">
  <Url type="text/html"
    template="http://example.com/search?color={custom:color?}" />
  <Query role="example" custom:color="blue" />
  <!-- ... -->
</OpenSearchDescription>
```

Role values

A role value consists of an optional prefix followed by the local role value. If the prefix is present it will be separated from the local role value with the ":" character. All role values are associated with a namespace, either implicitly in the case of local role values, or explicitly via a prefix in the case of fully qualified role values.

Role extensibility

The role attribute may take on values beyond those specified in this document provided they are fully qualified with a prefix and associated with a declared namespace. Clients that encounter unrecognized role values should continue to process the document as if the Query element containing the unrecognized role value did not appear.

Role prefix

A role prefix associates a local role name with a namespace. All prefixes must be previously declared as an XML namespace prefix on the containing Query element or ancestor elements.

Local role values

Local role values are not preceded by a prefix. Local role values are associated with the OpenSearch 1.1 namespace.

The following role values are identified with the OpenSearch 1.1 namespace. The list is exhaustive; only the role values listed below may appear in the OpenSearch 1.1 namespace.

Role values:

"request"

Represents the search query that can be performed to retrieve the same set of search results.

"example"

Represents a search query that can be performed to demonstrate the search engine.

"related"

Represents a search query that can be performed to retrieve similar but different search results.

"correction"

Represents a search query that can be performed to improve the result set, such as with a spelling correction.

"subset"

Represents a search query that will narrow the current set of search results.

"superset"

Represents a search query that will broaden the current set of search results.

Example of a local role value:

```
<Query role="related"
  title="A related search"
  searchTerms="tiger" />
```

Fully qualified role values

Fully qualified role values are preceded by a prefix. Fully qualified role values are associated with the namespace identified by the prefix on the containing Query element or ancestor elements.

Example of a fully qualified role value:

```
<Query xmlns:custom="http://example.com/opensearchextensions/1.0/"
  role="custom:synonym"
  title="Synonyms of 'cat'"
  searchTerms="feline" />
```

OpenSearch Response Elements

The OpenSearch response elements can be used by search engines to augment existing XML formats with search-related metadata.

OpenSearch response elements are typically found augmenting search results returned in list-based XML syndication formats, such as RSS 2.0 and Atom 1.0, but may be used in other contexts without restriction.

Example of a page of search results in the RSS 2.0 format:

```
<?xml version="1.0" encoding="UTF-8"?>
<rss version="2.0"
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/"
  xmlns:atom="http://www.w3.org/2005/Atom">
  <channel>
    <title>Example.com Search: New York history</title>
    <link>http://example.com/New+York+history</link>
    <description>Search results for "New York history" at
Example.com</description>
    <opensearch:totalResults>4230000</opensearch:totalResults>
    <opensearch:startIndex>21</opensearch:startIndex>
    <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
    <atom:link rel="search"
type="application/opensearchdescription+xml"
href="http://example.com/opensearchdescription.xml"/>
    <opensearch:Query role="request" searchTerms="New York History"
startPage="1" />
    <item>
      <title>New York History</title>

<link>http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html</link>
      <description>
        ... Harlem.NYC - A virtual tour and information on
        businesses ... with historic photos of Columbia's own New
        York
        neighborhood ... Internet Resources for the City's History.
        ...
      </description>
    </item>
  </channel>
```

```
</rss>
```

Example of a page of search results in the Atom 1.0 format:

```
<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <title>Example.com Search: New York history</title>
  <link href="http://example.com/New+York+history"/>
  <updated>2003-12-13T18:30:02Z</updated>
  <author>
    <name>Example.com, Inc.</name>
  </author>
  <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
  <opensearch:totalResults>4230000</opensearch:totalResults>
  <opensearch:startIndex>21</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
  <opensearch:Query role="request" searchTerms="New York History"
startPage="1" />
  <link rel="alternate"
href="http://example.com/New+York+History?pw=3" type="text/html"/>
  <link rel="self"
href="http://example.com/New+York+History?pw=3&format=atom"
type="application/atom+xml"/>
  <link rel="first"
href="http://example.com/New+York+History?pw=1&format=atom"
type="application/atom+xml"/>
  <link rel="previous"
href="http://example.com/New+York+History?pw=2&format=atom"
type="application/atom+xml"/>
  <link rel="next"
href="http://example.com/New+York+History?pw=4&format=atom"
type="application/atom+xml"/>
  <link rel="last"
href="http://example.com/New+York+History?pw=42299&format=atom"
type="application/atom+xml"/>
  <link rel="search" type="application/opensearchdescription+xml"
href="http://example.com/opensearchdescription.xml"/>
  <entry>
    <title>New York History</title>
    <link
href="http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html"/>
    <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
    <updated>2003-12-13T18:30:02Z</updated>
    <content type="text">
      ... Harlem.NYC - A virtual tour and information on
      businesses ... with historic photos of Columbia's own New York
      neighborhood ... Internet Resources for the City's History. ...
    </content>
  </entry>
</feed>
```

The "totalResults" element

The number of search results available for the current search.

If the totalResults element does not appear on the page then the search client should consider the current page to be the last page of search results.

Restrictions: The value must be a non-negative integer.

Default: The default value is equal to the offset index of the last search result on the current page.

Requirements: The element may appear zero or one time.

The "startIndex" element

The index of the first search result in the current set of search results.

If the startIndex element does not appear on the page then the search client should consider the current page to be the first page of search results.

Restrictions: The value must an integer.

Default: The default value is equal to the value of the "indexOffset" attribute of the "Url" element" in the OpenSearch description document.

Requirements: The element may appear zero or one time.

The "itemsPerPage" element

The number of search results returned per page.

If the itemsPerPage element does not appear on the page then the search client should use the number of items of the current page as the default page size.

Restrictions: The value must a non-negative integer.

Default: The default value is equal to the number of search results on the current page.

Requirements: The element may appear zero or one time.

The "Query" element

Defines a search query that can be performed by search clients. Please see the OpenSearch Query element specification for more information.

Search results should include a Query element of type="request" that can be used to recreate the search request that generate the current search response.

Requirements: The element may appear zero or more times.

Response metadata in HTML/XHTML

OpenSearch response metadata may be included in well-formed HTML/XHTML via the HTML 4.0.1 "meta" element.

The following meta element "name" attribute values are recognized under the profile associated with the OpenSearch 1.1 namespace:

"totalResults" - Corresponds to value of the "totalResults" element.

"startIndex" - Corresponds to value of the "startIndex" element.

"itemsPerPage" - Corresponds to value of the "itemsPerPage" element.

Example of a page of search results in the XHTML 1.0 format:

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Strict//EN"
  "http://www.w3.org/TR/xhtml1/DTD/xhtml1-strict.dtd">
<html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en">
  <head profile="http://a9.com/-/spec/opensearch/1.1/" >
    <title>Example.com Search: New York history</title>
    <link rel="search"
      type="application/opensearchdescription+xml"
      href="http://example.com/opensearchdescription.xml"
      title="Example.com Web Search" />
    <meta name="totalResults" content="4230000"/>
    <meta name="startIndex" content="1"/>
    <meta name="itemsPerPage" content="10"/>
  </head>
  <body>
    <ul>
      <li>
        <a
href="http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html">
          New York History
        </a>
        <div>
          ... Harlem.NYC - A virtual tour and information on
          businesses ... with historic photos of Columbia's own New
York
          neighborhood ... Internet Resources for the City's History.
          ...
        </div>
      </li>
      <!-- ... -->
    </ul>
  </body>
</html>
```


Annex E (informative)

Example XML documents

E.1 OpenSearch Description with Geospatial and Temporal Extensions

```
<?xml version="1.0" encoding="utf-8"?>
<OpenSearchDescription
  xmlns=http://a9.com/-/spec/opensearch/1.1/
  xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"
  xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/" >

  <ShortName>mer_rr_1p</ShortName>
  <LongName>ENVISAT MERIS Level 1 Dataset Search</LongName>
  <Description>This Search Service performs spatial-temporal queries on the
  ENVISAT MERIS level 1 Series with access points for ATOM feeds and HTML
  </Description>

  <Image height="38" width="120"
  type="image/png">http://www.terradue.com/images/terradue.png</Image>

  <Image height="16" width="16"
  type="image/vnd.microsoft.icon">http://www.terradue.com/favicon.ico</Image>

  <Query role="example"
  searchTerms="MER_RR__1PNPDE20060821_033013_000005632050_00290_23390_4614"/>

  <Query role="example" time:start="2004-09-01T00:00:00" time:end="2004-09-
  01T23:39:59" geo:box="-20,-20,20,20"/>

  <Developer>GeoSpatial Search Team at Terradue</Developer>
  <Attribution>Copyright 2009-2011, Terradue, Srl.</Attribution>

  <SyndicationRight>open</SyndicationRight>
  <AdultContent>>false</AdultContent>
  <Language>en-us</Language>

  <OutputEncoding>UTF-8</OutputEncoding>
  <InputEncoding>UTF-8</InputEncoding>

  <Url type="application/atom+xml " indexOffset="0" pageOffset="0"
  template="http:// maps.terradue.com/catalogue/gpod/MER_RR__1P/atom/
  ?count={count?}&startPage={startPage?}&startIndex={startIndex?}&
  q={searchTerms?}&start={time:start?}&stop={time:end?}&bbox={geo:
  box?}&uid={geo:uid?}"/>

  <Url type="text/html" indexOffset="0" pageOffset="0" template="
  maps.terradue.com/catalogue/gpod/MER_RR__1P/html/
  ?count={count?}&startPage={startPage?}&startIndex={startIndex?}&
  q={searchTerms?}&start={time:start?}&stop={time:end?}&bbox={geo:
  box?}&uid={geo:uid?}" />

</OpenSearchDescription>
```

E.2 ATOM Output with access to data resources

```

<?xml version="1.0" encoding="iso-8859-1"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:os="http://a9.com/-/spec/opensearch/1.1/"
      xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"
      xmlns:georss="http://www.georss.org/georss">
  <title>Terradue Catalogue Search ENVISAT ASAR Wide Swath Mode</title>
  <subtitle>Query Parameters bbox=6.73,36.73,18.45,47.07,count=2</subtitle>
  <link rel="self" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startIndex=0&
;count=2&bbox=6.73,36.73,18.45,47.07" title="self"/>
  <link rel="first" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startIndex=0&
;count=2&bbox=6.73,36.73,18.45,47.07" title="first"/>
  <link rel="next" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startIndex=2&
;count=2&bbox=6.73,36.73,18.45,47.07" title="next"/>
  <link rel="last" type="application/atom+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?startIndex=1410&
amp;count=2&bbox=6.73,36.73,18.45,47.07" title="last"/>
  <link rel="search" type="application/opensearchdescription+xml"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/description"
title="search"/>
  <os:totalResults>1411</os:totalResults>
  <os:startIndex>0</os:startIndex>
  <os:itemsPerPage>2</os:itemsPerPage>
  <os:Query os:role="request" count="2" startIndex="0"
geo:box="6.73,36.73,18.45,47.07" />
  <id>http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/?bbox=6.73,36.73,18
.45,47.07&count=2</id>
  <updated>2011-02-09T17:51:35Z</updated>
  <author><name>Terradue Srl</name><email>info@terradue.com</email>
<uri>http://www.terradue.com/</uri>
</author>
  <generator uri="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/atom/">
    Terradue GeoSpatial Search Team
  </generator>

<entry>
  <title>
    ASA_WSM_1PNIPA20100413_203816_000001042088_00315_42453_1360.N1</title>
  <id>ASA_WSM_1PNIPA20100413_203816_000001042088_00315_42453_1360.N1</id>
  <updated>2010-09-08T10:07:25.866Z</updated>
  <published>2010-04-21T08:38:24.000Z</published>
  <dc:date xmlns:dc="http://purl.org/dc/elements/1.1/">2010-04-
13T20:38:16.000Z/2010-04-13T20:40:00.000Z</dc:date>
  <link rel="via" type="application/xhtml+xml" title="RDF metadata entry"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PNIPA20100413
_203816_000001042088_00315_42453_1360.N1/xml">
  </link>
  <link rel="alternate" type="text/html" title="Product metadata entry"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PNIPA20100413
_203816_000001042088_00315_42453_1360.N1/html">
  </link>
  <summary type="html">
    <b>Start :</b> 2010-04-13T20:38:16.000Z
    <b>End :</b> 2010-04-13T20:40:00.000Z
    <br/><b>Acquisition Station :</b> PDAS-M
    <b>Processing Center :</b> I-PAC
    <br/><b>Orbit :</b> 42453</summary>
  <georss:where>

```

```

    <gml:Polygon><gml:exterior><gml:LinearRing><gml:posList
srsDimension="2">39.1114 15.822 39.8387 20.6359 45.5998 19.3542 44.8669 14.0809
39.1114 15.822</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon>
    </georss:where>
    <link rel="enclosure" href="https://grid-eo-
engine02.esrin.esa.int/gpfs/gpfs01/eo/ASA_WSM_1P/2010/04/ASA_WSM_1PNIPA20100413
_203816_000001042088_00315_42453_1360.N1" type="application/binary">
    </link>
  </entry>

  <entry>
    <title>
      ASA_WSM_1PNPDK20100412_211120_000003242088_00301_42439_5186.N1</title>
    <id>ASA_WSM_1PNPDK20100412_211120_000003242088_00301_42439_5186.N1</id>
    <updated>2010-09-08T10:07:25.705Z</updated>
    <published>2010-04-21T08:29:00.000Z</published>
    <dc:date xmlns:dc="http://purl.org/dc/elements/1.1/">2010-04-
12T21:11:20.000Z/2010-04-12T21:16:45.000Z</dc:date>
    <link rel="via" type="application/xhtml+xml" title="RDF metadata entry"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PNPDK20100412
_211120_000003242088_00301_42439_5186.N1/xml">
    </link>
    <link rel="alternate" type="text/html" title="Product metadata entry"
href="http://maps.terradue.com/catalogue/gpod/ASA_WSM_1P/ASA_WSM_1PNPDK20100412
_211120_000003242088_00301_42439_5186.N1/html">
    </link>
    <summary type="html">
      <b>Start :</b> 2010-04-12T21:11:20.000Z
      <b>End :</b> 2010-04-12T21:16:45.000Z
      <br><b>Acquisition Station :</b> PDHS-K
      <b>Processing Center :</b> PDHS-K
      <br><b>Orbit :</b> 42439</summary>
    <georss:where><gml:Polygon><gml:exterior><gml:LinearRing><gml:posList
srsDimension="2">44.219 6.3787 44.9595 11.6666 63.6045 6.4537 62.7278 -1.8193
44.219
6.3787</gml:posList></gml:LinearRing></gml:exterior></gml:Polygon></georss:wher
e>
    <link rel="enclosure" href="https://grid-eo-
engine02.esrin.esa.int/gpfs/gpfs01/eo/ASA_WSM_1P/2010/04/ASA_WSM_1PNPDK20100412
_211120_000003242088_00301_42439_5186.N1" type="application/binary">
    </link>
  </entry>
</feed>

```

```

<?xml version="1.0" encoding="UTF-8"?>
<feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:geo="http://a9.com/-/opensearch/extensions/geo/1.0/"
  xmlns:georss="http://www.georss.org/georss/10"
  xmlns:time="http://a9.com/-/opensearch/extensions/time/1.0/"
  xmlns:echo="http://www.echo.nasa.gov/esip"
  xmlns:dc="http://purl.org/dc/elements/1.1/"
  xmlns:os="http://a9.com/-/spec/opensearch/1.1/">
  <updated>2011-02-09T22:15:15.397Z</updated>
  <id>https://api.echo.nasa.gov:443/echo-esip/search/granule.atom</id>
  <author><name>ECHO</name><email>echo@echo.nasa.gov</email></author>
  <title type="text">ECHO granule search results</title>
  <os:itemsPerPage>10</os:itemsPerPage>
  <os:startIndex>1</os:startIndex>
  <os:Query role="request" echo:shortName="MOD02QKM" echo:versionId="5"
echo:dataCenter="LAADS" time:startTime="2009-05-01T00:00:00-04:00"
time:endTime="2009-05-04T00:00:00-04:00" ></os:Query>

```

```

<subtitle type="text">Granule search for MOD02QKM.5 and the data center:
LAADS with start time: 2009-05-01T00:00:00-04:00 with end time: 2009-05-
04T00:00:00-04:00</subtitle>
<entry>
  <id>laads:250798497</id>
  <title type="text">LAADS:250798497</title>
  <updated>2010-09-03T05:09:24.104Z</updated>
  <link
href="ftp://ladsftp.nascom.nasa.gov/allData/5/MOD02QKM/2009/121/MOD02QKM.A20091
21.0020.005.2010241162125.hdf" hreflang="en-US" type="application/x-hdfeos"
title="" rel="enclosure"></link>
  <link
href="ftp://ladsftp.nascom.nasa.gov/allData/5/MOBRGB/2009/121/MOBRGB.A2009121.0
020.005.2009126214738.jpg" hreflang="en-US" type="image/jpeg" title=""
rel="icon" length="660988"></link>
  <link href="http://mcst.gsfc.nasa.gov/L1B/product.html" hreflang="en-US"
title="MODIS Level 1B Product Information Page at MCST (MiscInformation) "
rel="describedby"></link>
  <echo:datasetId>MODIS/Terra Calibrated Radiances 5-Min L1B Swath 250m
V005</echo:datasetId>
  <dc:date>2009-05-01T00:20:00.000Z/2009-05-01T00:25:00.000Z</dc:date>
  <georss:polygon>53.803143 -20.003695 49.0732 -53.81698 63.68175 -74.32202
72.10222 -19.447964</georss:polygon>
  <summary type="html">&lt;p&gt;&lt;b&gt;Dataset ID : &lt;/b&gt;MODIS/Terra
Calibrated Radiances 5-Min L1B Swath 250m
V005&lt;/p&gt;&lt;p&gt;&lt;b&gt;Temporal
Extent&lt;/b&gt;&lt;/p&gt;&lt;p&gt;&lt;b&gt;Start Time : &lt;/b&gt;2009-05-
01T00:20:00.000Z &lt;b&gt;End Time : &lt;/b&gt;2009-05-
01T00:25:00.000Z&lt;p&gt;&lt;b&gt;Spatial Extent&lt;/b&gt;&lt;/p&gt;Polygon :
(53.803143, -20.003695), (49.0732, -53.81698), (63.68175, -74.32202),
(72.10222, -19.447964)</summary>
  </entry>

<entry>
  <id>laads:250797839</id>
  <title type="text">LAADS:250797839</title>
  <updated>2010-09-03T05:09:24.797Z</updated>
  <link
href="ftp://ladsftp.nascom.nasa.gov/allData/5/MOD02QKM/2009/121/MOD02QKM.A20091
21.0025.005.2010241162738.hdf" hreflang="en-US" type="application/x-hdfeos"
title="" rel="enclosure"></link>
  <link
href="ftp://ladsftp.nascom.nasa.gov/allData/5/MOBRGB/2009/121/MOBRGB.A2009121.0
025.005.2009126222233.jpg" hreflang="en-US" type="image/jpeg" title=""
rel="icon" length="726966"></link>
  <link href="http://mcst.gsfc.nasa.gov/L1B/product.html" hreflang="en-US"
title="MODIS Level 1B Product Information Page at MCST (MiscInformation) "
rel="describedby"></link>
  <echo:datasetId>MODIS/Terra Calibrated Radiances 5-Min L1B Swath 250m
V005</echo:datasetId>
  <dc:date>2009-05-01T00:25:00.000Z/2009-05-01T00:30:00.000Z</dc:date>
  <georss:polygon>71.581665 -18.330479 63.81684 -73.79089 70.95812 -119.12558
87.56117 64.28039</georss:polygon>
  <summary type="html">&lt;p&gt;&lt;b&gt;Dataset ID : &lt;/b&gt;MODIS/Terra
Calibrated Radiances 5-Min L1B Swath 250m
V005&lt;/p&gt;&lt;p&gt;&lt;b&gt;Temporal
Extent&lt;/b&gt;&lt;/p&gt;&lt;p&gt;&lt;b&gt;Start Time : &lt;/b&gt;2009-05-
01T00:25:00.000Z &lt;b&gt;End Time : &lt;/b&gt;2009-05-
01T00:30:00.000Z&lt;p&gt;&lt;b&gt;Spatial Extent&lt;/b&gt;&lt;/p&gt;Polygon :
(71.581665, -18.330479), (63.81684, -73.79089), (70.95812, -119.12558),
(87.56117, 64.28039)</summary>
  </entry>
</feed>

```

E.3 ATOM Output with access to Geospatial Search Services resources

```

<?xml version="1.0" encoding="iso-8859-1"?>
<feed xmlns="http://www.w3.org/2005/Atom"
      xmlns:os="http://a9.com/-/spec/opensearch/1.1/">
  <title>Terradue Catalogue Search Feed for Dataset Series</title>
  <subtitle>Query Parameters used q=MER_RR__1P</subtitle>
  <link rel="self" type="application/atom+xml" href="http://maps.terradue.com
/catalogue/gpod/atom?q=MER_RR__1P/">
  <os:totalResults>1</os:totalResults>
  <os:startIndex>0</os:startIndex>
  <os:itemsPerPage>20</os:itemsPerPage>
  <os:Query os:role="request" searchTerms="MER_RR__1P"/>
  <id>http:// maps.terradue.com /catalogue/gpod/atom?q=MER_RR__1P</id>
  <updated>2011-02-10T19:23:53Z</updated>
  <generator uri="http://genesi-dec.local/catalogue/gpod/atom">
    Terradue GeoSpatial Search Team
  </generator>
  <author>
    <name>Terradue Srl</name>
    <email>info@terradue.com</email>
    <uri>http://www.terradue.com</uri>
  </author>

  <entry>
    <title>ENVISAT MERIS Level 1 Reduced Resolution (MER_RR__1P)</title>
    <id>MER_RR__1P</id>
    <updated>2011-02-10T20:08:05.000Z </updated>
    <published>2010-04-21T08:29:00.000Z</published>
    <dc:date>2002-04-09T11:48:46.000Z/2011-02-10T20:08:05.000Z</dc:date>
    <link rel="enclosure" type="application/atom+xml" title="Atom feed for this
series" href="http://maps.terradue.com/catalogue/gpod/MER_RR__1P/atom"/>
    <link rel="search" type="application/opensearchdescription+xml"
href="http:// maps.terradue.com/catalogue/gpod/MER_RR__1P/description/"
title="Search the MER_RR__1P"/>
    <summary type="html">
The MEdium Resolution Imaging Spectrometer Instrument (MERIS) is a 68.5 degree
field-of-view push-broom imaging spectrometer that measures the solar radiation
reflected by the Earth, at a ground spatial resolution of 300 m, in 15 spectral
bands, programmable in width and position, in the visible and near infrared
wavelengths. MERIS allows global coverage of the Earth in 3 days. The primary
mission of MERIS is the measurement of sea colour in the oceans and in coastal
areas. Knowledge of sea colour can be converted into a measurement of
chlorophyll pigment concentration, suspended sediment concentration and of
atmospheric aerosol loads over water. The instrument always takes measurements
with full resolution; i.e., 260 m x 290 m ground resolution. Onboard averaging
generates the Reduced Resolution (RR) images. The MERIS Level 2 are water
leaving radiance (reflectance) and geophysical product generated systematically
from MERIS L1B. It has a coverage of 1150 km x 17500 km and a radiometric
resolution of 1.42 X 10-4 at sea level at 442.5nm. The file size is 621 Mbytes
per orbit and 47 Mbytes per scene. Auxiliary data include: Surface
identification flags included in the level 1b product Orbit state vector, Time
correlation parameters, Latitude, Longitude, altitude and topographic
corrections Sun azimuth, Sun elevation, view azimuth, view elevation, Mean Sea
Level pressure, Total column ozone, Total column water vapour, Wind speed
    </summary>
  </entry>
</feed>

```