searchRetrieve: Part 4. APD Binding for OpenSearch Version 1.0

Candidate OASIS Standard 01

25 October 2012

Specification URIs
This version:
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part4-opensearch/searchRetrieve-v1.0-cos01-part4-opensearch.doc (Authoritative)
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part4-opensearch/searchRetrieve-v1.0-cos01-part4-opensearch.html
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part4-opensearch/searchRetrieve-v1.0-cos01-part4-opensearch.pdf

Previous version:
N/A

Latest version:
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part4-opensearch.doc (Authoritative)
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part4-opensearch.html
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/searchRetrieve-v1.0-part4-opensearch.pdf

Technical Committee:
OASIS Search Web Services TC

Chairs:
Ray Denenberg (rden@loc.gov), Library of Congress
Matthew Dovey (m.dovey@jisc.ac.uk), JISC Executive, University of Bristol

Editors:
Ray Denenberg (rden@loc.gov), Library of Congress
Larry Dixson (ldix@loc.gov), Library of Congress
Ralph Levan (levan@oclc.org), OCLC
Janifer Gatenby (Janifer.Gatenby@oclc.org), OCLC
Tony Hammond (t.hammond@nature.com), Nature Publishing Group
Matthew Dovey (m.dovey@jisc.ac.uk), JISC Executive, University of Bristol

Additional artifacts:
This prose specification is one component of a Work Product which also includes:
* XML schemas: http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/schemas/
* *searchRetrieve: Part 0. Overview Version 1.0.*
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part0-overview/searchRetrieve-v1.0-cos01-part0-overview.html
* *searchRetrieve: Part 1. Abstract Protocol Definition Version 1.0.*
http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part1-apd/searchRetrieve-v1.0-cos01-part1-apd.html

- *searchRetrieve: Part 2. searchRetrieve Operation: APD Binding for SRU 1.2 Version 1.0*. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part2-sru1.2/searchRetrieve-v1.0-cos01-part2-sru1.2.html
- *searchRetrieve: Part 3. searchRetrieve Operation: APD Binding for SRU 2.0 Version 1.0*. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part3-sru2.0/searchRetrieve-v1.0-cos01-part3-sru2.0.html
- *searchRetrieve: Part 4. APD Binding for OpenSearch Version 1.0*. (this document) http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part4-opensearch/searchRetrieve-v1.0-cos01-part4-opensearch.html
- *searchRetrieve: Part 5. CQL: The Contextual Query Language Version 1.0*. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part5-cql/searchRetrieve-v1.0-cos01-part5-cql.html
- *searchRetrieve: Part 6. SRU Scan Operation Version 1.0*. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part6-scan/searchRetrieve-v1.0-cos01-part6-scan.html
- *searchRetrieve: Part 7. SRU Explain Operation Version 1.0*. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part7-explain/searchRetrieve-v1.0-cos01-part7-explain.html

Related work:

This specification is related to:

- OpenSearch » 1.1 » Draft 5 specification. http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5

Abstract:

This document, "*APD Binding for OpenSearch*" is a binding of the OASIS SWS Abstract Protocol Definition to the OpenSearch version 1.1 Draft 5 Specification. This is one of a set of documents for the OASIS Search Web Services (SWS) initiative.

Status:

This document was last revised or approved by the OASIS Search Web Services TC on the above date. The level of approval is also listed above. Check the "Latest version" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "Send A Comment" button on the Technical Committee's web page at http://www.oasis-open.org/committees/search-ws/.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (http://www.oasis-open.org/committees/search-ws/ipr.php).

Citation format:

When referencing this specification the following citation format should be used:

[SearchRetrievePt4]

searchRetrieve: Part 4. APD Binding for OpenSearch Version 1.0. 25 October 2012. Candidate OASIS Standard 01. http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/cos01/part4-opensearch/searchRetrieve-v1.0-cos01-part4-opensearch.html.

Notices

Copyright © OASIS Open 2012. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "OASIS IPR Policy"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "AS IS" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on OASIS' procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The name "OASIS" is a trademark of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see http://www.oasis-open.org/policies-guidelines/trademark for above guidance.

Table of Contents

1 Introduction

This is one of a set of documents for the OASIS Search Web Services (SWS) initiative.

This document, "*APD Binding for OpenSearch"* is a binding of the OASIS SWS Abstract Protocol Definition.

This specification is intended to be fully compatible with http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5

The set of documents includes the Abstract Protocol Definition (APD) for searchRetrieve operation, which presents the model for the SearchRetrieve operation and serves as a guideline for the development of *application protocol bindings* describing the capabilities and general characteristic of a server or search engine, and how it is to be accessed.

The collection of documents also includes three bindings (3, 4, and 5 in the list below). This document is one of the three.

The eight documents in this collection of specifications are:

1. Overview
2. APD
3. SRU1.2
4. SRU2.0
5. OpenSearch (this document)
6. CQL
7. Scan
8. Explain

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in [RFC2119].

1.2 References

All references for the set of documents in this collection are supplied in the Overview document:

searchRetrieve: Part 0. Overview Version 1.0

http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/csd01/part0-overview/searchRetrieve-v1.0-csd01-part0-overview.doc

1.3 Namespace

All XML namespaces for the set of documents in this collection are supplied in the Overview document:
searchRetrieve: Part 0. Overview Version 1.0

http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/csd01/part0-overview/searchRetrieve-v1.0-csd01-part0-overview.doc

2 Model

This document describes the OpenSearch model, request parameters, response elements, and description document.

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.

2.1 Relationship to Abstract Protocol Definition

The APD defines abstract request parameters and abstract response elements. A binding lists those abstract parameters and elements applicable to that binding and indicates the corresponding actual name of the parameter or element to be transmitted in a request or response.

```
Example.
The APD defines the abstract parameter: startPosition as "The position within
the result set of the first item to be returned."
And OpenSearch refers to that abstract parameter and notes that its name, as
used in the OpenSearch specification is 'startIndex'.  Thus the request
parameter 'startRecord' in OpenSearch represents the abstract parameter
startPosition in the APD.
```

Different bindings may use different names to represent this same abstract parameter, and its semantics may differ across those bindings as the binding models differ. It is the responsibility of the binding to explain these differences in terms of their respective models.

2.2 Processing Model

A server provides a description document that a client reads to determine how to formulate a search/retrieve request and interpret the response. The client may send a request, including search terms, to the server, who replies with a response that includes results based on the search terms.

The server returns results either as a stream ("stream mode") or a page ("page mode"). A stream is an arbitrary range of results, for example, results 10 through 100. In page mode, the server groups the results into pages, and returns one page. The server will always return results as a stream or always as a page, and indicates one or the other in its description file.

If the server returns a page, the request may include the 'count' parameter, suggesting how many results there should be per page. The request may also include the 'startPage' parameter indicating which page is desired. (See note 1.) The server may ignore the 'count' parameter and determine the number of results per page itself. (See note 2.)

If the server returns a stream, the request may include the parameter 'startIndex' to indicate the desired position within the result set of the first result within the stream. For example if the value of the 'startIndex' parameter is 61, and if the server returns 30 results, the stream will consist of results 61 through 90. The request may also include the 'count' parameter (for example, a value of 30, if the client wants results 61 through 90) but the server may ignore it. (See note 3.)

The response includes the element <totalResults>, the number of results found by the search. This element will be omitted only if the last of the available results is included in the response.

So the client can scroll through the results by issuing repeated requests until there is a response which omits the <totalResults> element, the omission signaling that there are no further results. Each request uses the same value for the parameter 'searchTerms', and :

- In stream mode: the value of the parameter 'startIndex' is the previous value plus the number of results included in the previous response.

- In page mode: the value of the parameter 'startPage' is the previous value plus one (1).

Notes:

83   1. The server returns one page only, contrary to the implication of the parameter name, 'startPage'.
84   2. If the server has ignored the count parameter, then the startPage parameter that the client has
85   suggested will not retrieve the specific results that the client had in mind.
86   3. The 'count' parameter is defined as "desired number of results per page", but it applies not only in
87   page mode, but also in stream mode: In stream mode the entire list of results is considered a
88   single page.

2.3 Result Set Model

90 There are no explicit (named) result sets in openSearch. It is assumed that if multiple requests are issued
91 to a search engine with the same value of parameter 'searchTerms' the results will be identical, that is,
92 the same set of results in the same order. Therefore the parameter 'searchTerms' can be considered to
93 represent a result set.

2.4 Data Model

95 The data model of the Abstract Protocol Model says that a "*datastore* is a collection of units of data. Such
96 a unit is referred to as an *item…*"

```
97      In this binding:
98          •  A data store is referred to as a search engine.
99          •  For an openSearch response, the abstract element <item> corresponds to
100             an element defined by the response schema, for example an <entry> or
101             <item> in ATOM 1.0 or RSS 2.0 respectively.
102         •  An item is sometimes referred to as a "result".
```

103 The Abstract Protocol Model further notes that "associated with a datastore are one or more formats that
104 may be used for the transfer of items from the server to the client. Such a format is referred to as an *item*
105 *type..*"

```
106     In this binding:
107         •  There is no parameter equivalent to itemType; the format is internally
108            defined by the response format.
```

109 The Abstract Protocol Model further notes that "The server may also partition the result set into *result*
110 *groups.*"

```
111     In this binding:
112         •  'groups are referred to as 'pages'.
```

2.5 Diagnostic Model

114 OpenSearch does not include specific diagnostics. HTTP diagnostics are returned when a URL is badly
115 formed or the server is unable to perform the search contained within the URL.

116 If the server is able to interpret but not process a request it can send back the OpenSearch Description
117 Document that explains how to correctly construct a request.

2.6 Description and Discovery Model

119 OpenSearch mandates an OpenSearch Description Document that is consistent with the requirements of
120 the Abstract Protocol Definition. There are six groups of data that may be included:

121   *1.* **General Description of the Server and its Capabilities.** The OpenSearch Description
122   Document includes a *shortName*, and *longName* and also *tags* which are keywords that describe
123   the server's *content* (*datastore*)*.*

124   2. **How to Formulate a Request.** The OpenSearch Description Document includes a mandatory
125   URL element containing a mandatory request template.

126   3. **Query Grammar.** There is no explicit search grammar associated with OpenSearch.

127   4. **How to Interpret a Response.** The type attribute of the URL element indicates the MIME type
128   (*format*) of the response.

129		5.	**How to Process Results.** The OpenSearch Description Document may include extra elements explaining how to process and display the search results. These include an *image* and *attribution* for display against the results, an indication of *adultContent* and *syndicationRight*.
130			
131			
132		6.	**Auto-Discovery Process.** An OpenSearch description documents may include a reference to other OpenSearch description documents.
133			

134 The OpenSearch URL template represents a parameterized form of the URL by which a search engine is
135 queried. The client processes the template, replacing each instance of a template parameter, with the
136 value for that parameter. The template parameters are the request parameters shown below.

137 3 OpenSearch Request

138 3.1 Actual Request Parameters for this Binding

139 *Table 1: Summary of Actual Request Parameters*

Parameter Name	Description	Type/Value
searchTerms	keyword or keywords	string
startIndex	index of first search result desired by the client	positive integer
count	Number of search results desired by the client.	positive integer
startPage	page number of the set of search results desired by the search client.	positive integer
language	desired language for search results.	RFC 5646, or '*' to mean "any language"
inputEncoding	character encoding of the search request.	IANA Character Set Assignments, default UTF-8
outputEncoding	character encoding requested for the search results. The default is UTF-8	IANA Character Set Assignments, default UTF-8

140 3.2 Abstract Vs. Actual Parameters

141 The following table lists the Abstract parameters defined in the Abstract Protocol Definition, and the
142 openSearch actual parameters, in two columns, with corresponding parameters in the same row.

143 *Table 2: Abstract Vs. Actual parameters*

Abstract Parameter Name from APD	openSearch Parameter
responseType	(None. See type attribute of <url> element)
query	searchTerms
startPosition	startIndex
maximumItems	count
group	startPage
responseItemType	(None. See Data Model, fourth bullet.)
sortOrder	*(None)*
(None)	language
(None)	inputEncoding
(None)	outputEncoding

4 OpenSearch Response

4.1 Response Elements

This section summarizes the OpenSearch response elements and compares them with the abstract elements defined in the Abstract Protocol Definition.

4.1.1 Actual Response Elements

The following table describes the actual XML response elements.

Table 3: Summary of Actual Response Elements

Element	Type	Occurrence	Meaning
<totalResults>	xs:integer	zero or one	number of search results.
<startIndex>	xs:positiveInteger	zero or one	index of the first search result in the response.
<itemsPerPage>	xs:positiveInteger	zero or one	number of search results returned per page.
<query>	xs:string	zero or more	See "Query".

4.1.2 Abstract Vs. Actual Elements

The following table lists abstract elements from the Abstract Protocol Definition, and the openSearch actual elements, in two columns, with corresponding elements in the same row.

Table 4: Abstract Vs. Actual elements

Abstract Element From APD	openSearch Element
<numberOfItems>	<totalResults>
<numberOfGroups>	(none)
<resultSetId>	(none)
<item>	defined by the response schema, for example an <entry> in ATOM 1.0 or <item>RSS 2.0.
<nextPosition>	*In page mode*: find the <link> element where the value of the 'rel' attribute is "next". Within the corresponding query ('href' attribute) the value of the parameter corresponding to startPage is the number of the next page. *In stream mode:* <startIndex> + <itemsPerPage> - 1.
<nextGroup>	(none)
<diagnostics>	(none)
<echoedRequest>	the value of the 'href' attribute for the <link> element where the value of the 'rel' attribute is "self".
(none)	startIndex
(none)	itemsPerPage

(none)	Query

4.2 OpenSearch Response Examples

154

155 **Example 1:** A page of search results in Atom 1.0

156 The line numbers on the left are added for reference in the analysis below.

```
157     <?xml version="1.0" encoding="UTF-8"?>
158      <feed xmlns="http://www.w3.org/2005/Atom"
159             xmlns:OpenSearch="http://a9.com/-/spec/OpenSearch/1.1/">
160        <title>Example.com Search: New York history</title>
161        <link href="http://example.com/New+York+history"/>
162        <updated>2003-12-13T18:30:02Z</updated>
163        <author>
164          <name>Example.com, Inc.</name>
165        </author>
166        <id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</id>
167     1. <OpenSearch:totalResults>4230000</OpenSearch:totalResults>
168     2. <OpenSearch:startIndex>21</OpenSearch:startIndex>
169     3. <OpenSearch:itemsPerPage>10</OpenSearch:itemsPerPage>
170        <OpenSearch:Query
171     4.     role="request" searchTerms="New York History" startPage="1" />
172        <link
173          rel="alternate" href="http://example.com/New+York+History?pw=3"
174          type="text/html"/>
175        <link
176     5.    rel="self"
177          href= "http://example.com/New+York+History?pw=3&amp;format=atom"
178          type="application/atom+xml"/>
179        <link
180     6.    rel="first"
181          href="http://example.com/New+York+History?pw=1&amp;format=atom"
182          type="application/atom+xml"/>
183        <link
184     7.    rel="previous"
185          href="http://example.com/New+York+History?pw=2&amp;format=atom"
186          type="application/atom+xml"/>
187     8. <link
188           rel="next"
189          href="http://example.com/New+York+History?pw=4&amp;format=atom"
190          type="application/atom+xml"/>
191     9. <link
192           rel="last"
193          href="http://example.com/New+York+History?pw=4229991&amp;format=atom"
194          type="application/atom+xml"/>
195        <link
196           rel="search" type="application/OpenSearchdescription+xml"
197           href="http://example.com/OpenSearchdescription.xml"/>
198        <entry>
199          <title>New York History</title>
200          <link
201            href="http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html"/>
202          <id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</id>
203          <updated>2003-12-13T18:30:02Z</updated>
204          <content type="text">
205             ... Harlem.NYC - A virtual tour and information on
206            businesses ...  with historic photos of Columbia's own New York
207            neighborhood ... Internet Resources for the City's History. ...
208          </content>
209        </entry>
```

210

211

Analysis of the above example.

213   'pw' is the name of the parameter corresponding to the openSearch parameter 'startPage', for this server.

214   • Lines 1-3 indicate that there were 4,230,000 results associated with the search term "New York

215   History". This response includes 10 results beginning with result 21 (thus results 21-30).

216   • Line 4 (<query role="request"…>) indicates how to regenerate the request from the beginning of

217   the results (parameters searchTerms="New York History" and startPage="1")

218   • Line 5 indicates that the URL to generate the same request that generated this response (<link

219   rel="self…>) with a response in Atom format (type="application/atom+xml"), is

220   "http://example.com/New+York+History?pw=3&format=atom"

221   • line 6   (rel="first") indicates that the URL to get the first page of results, in atom, is

222   href="http://example.com/New+York+History?pw=1&format=atom".

223   • line 7   (rel="previous") indicates that the URL to get the previous page of results is

224   href="http://example.com/New+York+History?pw=2&format=atom".

225   • line 8   (rel="next") indicates that the URL to get the next page of results is

226   href="http://example.com/New+York+History?pw=4&format=atom".

227   • line 9   (rel="last") indicates that the URL to get the last page of results is

228   href="http://example.com/New+York+History?pw=4229991&format=atom".

229

230   **Example 2:** a page of search results in the RSS 2.0 format

```
231    <?xml version="1.0" encoding="UTF-8"?>
232    <rss version="2.0"
233        xmlns:OpenSearch="http://a9.com/-/spec/OpenSearch/1.1/"
234        xmlns:atom="http://www.w3.org/2005/Atom">
235      <channel>
236        <title>Example.com Search: New York history</title>
237        <link>http://example.com/New+York+history</link>
238        <description>Search results for "New York history" at
239    Example.com</description>
240        <OpenSearch:totalResults>4230000</OpenSearch:totalResults>
241        <OpenSearch:startIndex>21</OpenSearch:startIndex>
242        <OpenSearch:itemsPerPage>10</OpenSearch:itemsPerPage>
243        <atom:link
244          rel="search" type="application/OpenSearchdescription+xml"
245          href="http://example.com/OpenSearchdescription.xml"/>
246        <OpenSearch:Query
247           role="request" searchTerms="New York History" startPage="1" />
248        <item>
249          <title>New York History</title>
250          <link>http://www.columbia.edu/cu/lweb/eguids/amerihist/nyc.html</link>
251          <description>
252             ... Harlem.NYC - A virtual tour and information on
253            businesses ...  with historic photos of Columbia's own New York
254            neighborhood ... Internet Resources for the City's History. ...
255          </description>
256        </item>
257      </channel>
258    </rss>
```

259

260

Example 3 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>
```

292 5 Open Search Description Document

293   A server providing an OpenSearch interface provides a description document to describe the interface.

294   OpenSearch description documents have the following mime type (pending IANA registration):

295   application/OpenSearchdescription+xml

296   OpenSearch description elements (table below) have the following XML Namespaces URI

297   http://a9.com/-/spec/OpenSearch/1.1/

298 5.1 Description Elements

299   *Table 5: Description Elements*

Element	Occurence	Description/ Restrictions
OpenSearchDescription	Must occur exactly once (as the root node of the document).	
ShortName	Must occur exactly once.	16 or fewer characters of plain text (no HTML or other markup).
Description	Must occur exactly once.	1024 or fewer characters of plain text (no HTML or other markup).
Url	Must occur one or more times.	See URL Element.
Contact	May occur zero or one time.	Email address for owner of the description document
Tags	May occur zero or one time.	keywords describing search content. One or more single words delimited by spaces. Total 1024 or fewer characters of plain text (no HTML or other markup).
LongName	May occur zero or one time.	An extended human-readable title that identifies this search engine. 48 or fewer characters of plain text (no HTML or other markup).
Image	May occur zero or more times.	URL for an image that can be used in association with this search content. Attributes: height, width, type (MIME); all optional

Element	Occurence	Description/ Restrictions
Query	May occur zero or one time.	See Query Element.
Developer	May occur zero or one time.	human-readable name or identifier for creator or maintainer of the description document. 64 or fewer characters of plain text (no HTML or other markup).
Attribution		a list of all entities to be credited for the content in the search feed. 256 or fewer characters of plain text (no HTML or other markup).
SyndicationRight		the degree to which search results provided by this search engine can be queried, displayed, and redistributed See table below.
AdultContent	May occur zero or one time.	boolean: true if the search results may contain material intended only for adults. "false", "FALSE", "0", "no", and "NO" will be considered boolean FALSE; all other strings will be considered boolean TRUE. Default: "false"
Language	May occur zero or more times.	one "Language" element for each language that the search engine supports. Values from RFC 5646. A value of "*" (default) signifies that the search engine does not restrict search results to any particular language.
InputEncoding	May occur zero or more times. (One for each character encoding that can be used to encode search requests.)	as specified by the IANA Character Set Assignments. Default: "UTF-8".

300   **Values for Parameter SyndicationRight**

value | V right →	The search client may request search results	may display the search results to end users	client may send the search results to other search clients
"open",	yes	yes	yes
"limited"	yes	yes	no
"private"	yes	no	no
"closed"	no	no	no

5.1.1 URL Element

The Url element has the form as shown in the following example:

```
<Url
    type= "application/rss+xml"
    indexOffset="0"
    rel="results"
    template=
"http://example.com/search?q={searchTerms}&amp;start={startIndex?}&amp;format=
rss"/>
```

5.1.1.1 Attributes of the URL Element

indexOffset, pageOffset. The starting number for the first search result or first page of search results, for index-based and page-based results respectively. Defaults are "1"; the "indexOffset" and "pageOffset" attributes may be used to inform search clients of different starting values.

type. The MIME type of the search result format. The 'type' attribute of the <url> element is what the client uses to determine how to request a specific response format. There may be several <url> elements, each with a type attribute of a different value. The one with the desired value (mime type) is the one belonging to the template to use for that response format. This attribute is **required**.

Rel. The role of the resource being described in relation to the description document. A space-delimited list of valid rel value, each either a URI or one of the following:

- "results" (default)

  Requests search results in the specified format.

- "suggestions"

  Request search suggestions in the specified format.

- "self"

  Represents the canonical URL of this description document.

- "collection"

  Requests a set of resources.

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 meaning of a rel value it should ignore it.

template. See Template Syntax.

5.1.1.2 Template Syntax

The OpenSearch URL template represents 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.

All parameter names are associated with a namespace; the OpenSearch 1.1 namespace is the default if no other is indicated. Parameter names are case sensitive.

A template parameter is designated as optional by using the "?" as shown in the two examples below.

The template parameters are the openSearch request parameters in table 1.

Examples

Example 1: a search URL template that contains a template parameter:

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

In this example, the openSearch parameter 'searchTerms', in curly brackets, is an abstract parameter to be replaced by the actual parameter for this search engine, in this case 'q'. {searchTerms}" is required as indicated by the absence of "?"

Example 2: optional template parameter:

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

This example, the question mark, "?", is used to mean that the parameter startPage is optional.

5.1.2 Query Element

The Query element may appear in a description document or search response and is used to supply search requests that can be performed by a search client.

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

At least one Query element with role="example" should be provided in each description document so that search clients can test the search engine. In addition a Query element with role="request" in each search response so that search clients can recreate the current search.

5.1.2.1 Attributes of the Query Element

The query element may contain the following attributes defined in the OpenSearch namespace, as well as attributes from external namespace.

- **role.** Required. Values:
    - o "request" : the search query can be performed to retrieve the same set of search results.
    - o "example"
    - o   "related" :thequery can be performed to retrieve similar but different search results.
    - o "correction" : corrected query (e.g. a spelling correction) which can be performed to improve results set,
    - o "subset": a query that will narrow the current set of search results.
    - o "superset": a query that will broaden the current set of search results.
- **title. P**lain text string describing the search request. 256 or fewer characters. optional.
- **totalResults.** Expected number of results to be found if the search request were made. Optional.
- **searchTerms, count, startIndex, startPage, language, inputEncoding, outputEncoding**. The value representing these parameters. All are optional.

5.1.2.2 Query Element Examples

372

373 **Example 1:** Query element in a description document to provide an example search request

374
```
<Query role="example" searchTerms="cat" />
```
375

376 **Example 2:** Query element in a response to echo back the original search request

377
```
<Query role="request" searchTerms="cat" startPage="1" />
```

378 **Example 3**: Query element in a response to correct the spelling of "OpenSurch":

379
380
```
<Query role="correction" searchTerms="OpenSearch" totalResults="854000"
title="Spelling correction"/>
```

381 **Example 4:** An extended parameter

382
383
384
385
386
```
<Query xmlns:custom="http://example.com/OpenSearchextensions/1.0/"
       role="example"
       searchTerms="cat"
       custom:color="blue"
       title="Sample search" />
```

387 **Example 5:** an extended role

388
389
390
391
```
<Query xmlns:custom="http://example.com/OpenSearchextensions/1.0/"
       role="custom:synonym"
       title="Synonym of 'cat'"
       searchTerms="feline" />
```

392

393 **Example 6:** a set of Query elements used in the context of an Atom-based OpenSearch response

394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
```
<?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>
```

5.2 Example Description Documents

409

410 **Example 1:** a simple OpenSearch description document

411
412
413
414
415
416
417
418
419
420
421
```
<?xml version="1.0" encoding="UTF-8"?>
<OpenSearchDescription xmlns="http://a9.com/-/spec/OpenSearch/1.1/">
      <ShortName>Web Search</ShortName>
      <Description>Use Example.com to search the Web.</Description>
      <Tags>example web</Tags>
      <Contact>admin@example.com</Contact>
      <Url rel="results"
          type="application/rss+xml"
          template=
"http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=rss"/>
</OpenSearchDescription>
```

422

423 **Example 2:** a detailed OpenSearch description document

```
424    <?xml version="1.0" encoding="UTF-8"?>
425     <OpenSearchDescription xmlns="http://a9.com/-/spec/OpenSearch/1.1/">
426             <ShortName>Web Search</ShortName>
427             <Description>Use Example.com to search the Web.</Description>
428             <Tags>example web</Tags>
429             <Contact>admin@example.com</Contact>
430             <Url
431              rel="results"
432              type="application/atom+xml"
433              template=
434    "http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=atom"/>
435             <Url
436              rel="results"
437              type="application/rss+xml"
438              template=
439    "http://example.com/?q={searchTerms}&amp;pw={startPage?}&amp;format=rss"/>
440             <Url
441              rel="results"
442              type="text/html"
443              template="http://example.com/?q={searchTerms}&amp;pw={startPage?}"/>
444           <LongName>Example.com Web Search</LongName>
445             <Image height="64" width="64"
446    type="image/png">http://example.com/websearch.png</Image>
447             <Image height="16" width="16"
448    type="image/vnd.microsoft.icon">http://example.com/websearch.ico</Image>
449             <Query role="example" searchTerms="cat" />
450             <Developer>Example.com Development Team</Developer>
451             <Attribution>
452        Search data Copyright 2005, Example.com, Inc., All Rights Reserved
453             </Attribution>
454             <SyndicationRight>open</SyndicationRight>
455             <AdultContent>false</AdultContent>
456             <Language>en-us</Language>
457           <OutputEncoding>UTF-8</OutputEncoding>
458             <InputEncoding>UTF-8</InputEncoding>
459     </OpenSearchDescription>
```

460 5.3 Extensibility

461 OpenSearch description documents can be extended provided that all foreign elements and attributes are
462 associated with an explicit XML namespace. Clients that encounter unrecognized foreign elements
463 should ignore them and continue to process the document as if these elements did not appear.

464 5.4 Autodiscovery

465 An OpenSearch description documents may include a reference to other OpenSearch description
466 documents by including "link" elements on search results, with the following attributes/values:

467   • type ="application/OpenSearchdescription+xml".
468   • rel="search".
469   • href= [URI of an OpenSearch description document].
470   • title= [human-readable plain text string describing the search engine].

471 And in addition, for HTML and XHTML documents:

472   • The HTML <head/> element should include the attribute/value pair:
473   profile="http://a9.com/-/spec/OpenSearch/1.1/".

474 Autodiscovery Examples

475 **Example 1:** Atom-based search results with an OpenSearch autodiscovery link element

476

```
477     <?xml version="1.0" encoding="UTF-8"?>
478     <feed xmlns="http://www.w3.org/2005/Atom"
479         xmlns:OpenSearch="http://a9.com/-/spec/OpenSearch/1.1/">
480     ……..
481       <link rel="search"
482           href="http://example.com/OpenSearchdescription.xml"
483           type="application/OpenSearchdescription+xml"
484           title="Content Search" />
485       ………
486     </feed>
```

487

488

489 **Example 2:** RSS-based search results with an OpenSearch autodiscovery link element

490

```
491     <?xml version="1.0" encoding="UTF-8"?>
492     <rss version="2.0"
493         xmlns:atom="http://www.w3.org/2005/Atom">
494       <channel>
495             …….
496         <atom:link rel="search"
497                   href="http://example.com/OpenSearchdescription.xml"
498                   type="application/OpenSearchdescription+xml"
499                   title="Content Search" />
500                       ……..
501       </channel>
502     </rss>
```

503

504

505 **Example 3:** An HTML document that includes OpenSearch autodiscovery link elements

506

```
507     <!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01//EN"
508     "http://www.w3.org/TR/html4/strict.dtd">
509     <html xmlns="http://www.w3.org/1999/xhtml" xml:lang="en" lang="en" dir="ltr">
510       <head profile="http://a9.com/-/spec/OpenSearch/1.1/">
511         <!--- ... --->
512         <link rel="search"
513               type="application/OpenSearchdescription+xml"
514               href="http://example.com/content-search.xml"
515               title="Content search" />
516         <link rel="search"
517               type="application/OpenSearchdescription+xml"
518               href="http://example.com/comment-search.xml"
519               title="Comments search" />
520         <!--- ... --->
521       </head>
522       <body>
523         <!--- ... --->
524       </body>
525     </html>
```

526

527 6 Conformance

528 6.1 Client Conformance

529 An OpenSearch client conforms to this specification if:

530 Having retrieved an OpenSearch description document, it MUST be able to extract the <Url> element and
531 interpret its attributes.

- 532 • In particular, from the 'type' attributes it MUST be able to determine which response formats it
  533   supports; and for each:
- 534 • From the template attribute it MUST be able to replace each instance of a template parameter,
  535   with a value (which may be empty) for that parameter, and formulate a request.

536 This conformance clause does not specify any particular format that a client must support, but it MUST
537 support at least one.

538 6.2 Server Conformance

539 An OpenSearch serve conforms to this specification if there exists a discoverable and accessible

540 OpenSearch description document of type application/opensearchdescription+xml. It must be an XML

541 document with:

542 • Root element <OpenSearchDescription>.

543 • Namespace http://a9.com/-/spec/opensearch/1.1/

544 • Elements:

545   o   <ShortName> occurring exactly once.

546   o   <Description>" occurring exactly once.

547   o   <Url> occurring one or more times, and each occurrence including the attributes:

548   ▪   'template', conforming to the template syntax.

549   ▪   'type' – a valid MIME type.

550

551

Appendix A. Acknowledgements

Acknowlegements are supplied in the Overview document:

searchRetrieve: Part 0. Overview Version 1.0

http://docs.oasis-open.org/search-ws/searchRetrieve/v1.0/csd01/part0-overview/searchRetrieve-v1.0-csd01-part0-overview.doc