



**Intelligence Community and Department of Defense  
Content Discovery & Retrieval Integrated Project  
Team**

***IC-DoD REST Interface Encoding Specification  
for CDR Search***

***Version 3.0***

***Approval Date: 3-OCT-2012***

## **Distribution Notice**

This document has been approved for Public Release and is available for use without restrictions

## REVISION/HISTORY

<b>Doc Revision</b>	<b>Revised By</b>	<b>Revision Date</b>	<b>Revisions</b>
0.1		29 January 2010	Initial draft for subgroup review.
0.2	Dave Lemen, Dan Wigglesworth	1 February 2010	Second draft, limiting specification to format-agnostic OpenSearch implementation.
0.3	Dave Lemen	16 February 2010	
0.4	Dave Lemen	23 February 2010	
0.5	Dave Lemen	9 March 2010	
1.0	Mary Farley	14 April 2010	Technical Edits
1.1	Dave Lemen	27 May 2011	Updates based on specification framework changes
1.2	Dave Lemen	24 June 2011	Changes based on 23 June adjudicated comments
1.3	Dave Lemen	10 Aug 2011	Changes based on comments to v1.2
1.4	Dave Lemen	20 Oct 2011	Changes based on DMO comments on v1.3
1.4-20111128	Matt McKnight	28 Nov 2011	Changes based on SCA comments to v1.3
1.4-20111214	Wigglesworth	14 Dec 2011	Modified for ASEC Comments
2.0			
3.0	Ken Laskey	28 March 2012	Synchronize changes in other specs

## TABLE OF CONTENTS

<b>1</b>	<b>Introduction.....</b>	<b>6</b>
1.1	Service Overview.....	6
1.2	Scope.....	6
1.3	Artifact Overview.....	6
1.4	Notational Convention.....	8
1.5	Conformance.....	8
1.6	Namespaces.....	9
1.7	License.....	9
1.8	Security.....	9
<b>2</b>	<b>Search Service Behavior.....</b>	<b>9</b>
2.1	Search.....	9
2.2	Paging.....	10
2.3	Result Presentation.....	10
<b>3</b>	<b>Search Service Interface.....</b>	<b>10</b>
3.1	Search Function.....	10
3.1.1	Preconditions.....	10
3.1.2	Input.....	11
3.1.3	Output.....	14
3.1.4	Post-conditions.....	16
3.1.5	Fault Conditions.....	16
3.2	Results Paging Function.....	17
3.2.1	Preconditions.....	17
3.2.2	Input.....	17
3.2.3	Output.....	18
3.2.4	Post-conditions.....	18
3.2.5	Fault Conditions.....	18
<b>4</b>	<b>References.....</b>	<b>19</b>
<b>Appendix A. Mapping to Specification Framework.....</b>		<b>20</b>
A.1.	Search Function.....	20
	Input.....	20
	Output.....	20
A.2.	Results Paging Function.....	21
	Input.....	21
	Output.....	21
<b>Appendix B. Changes from Prior Version.....</b>		<b>22</b>

## LIST OF FIGURES

Figure 1: CDR Architecture Documents.....	7
Figure 2: Result Set Structure .....	14
Figure 3: Example output using Atom format .....	15
Figure 4: Paging Links used in Atom Feed .....	18

## LIST OF TABLES

Table 1. Referenced XML Namespaces .....	9
Table 2. OpenSearch Input Parameters.....	12
Table 3. OpenSearch Response Elements used within Atom Feed .....	15
Table 4. OpenSearch Result Extension.....	16
Table 5. Fault Conditions and HTTP Responses .....	17
Table 6: Search Function Input Mapping to Specification Framework.....	20
Table 7. Search Function Output Mapping to Specification Framework .....	21
Table 8: Summary of Changes from V1.0.....	22

# 1 Introduction

## 1.1 Service Overview

The Search Component, as defined by the Intelligence Community/Department of Defense (IC/DoD) Content Discovery and Retrieval (CDR) Specification Framework [CDR-SF], serves as the primary mechanism to expose content collections for discovery and accessibility. This component provides a common service interface and behavioral model for IC and DoD content collections, enabling content consumers to discover relevant content resources from disparate collections across the IC/DoD enterprise.

This specification defines requirements and provides guidance for the realization of the CDR Search Component as a RESTful<sup>1</sup>, OpenSearch [OS] web service, hereafter termed a Search Service in this document. The content of this specification describes a Search Service's behavior, interface and other aspects in detail, providing enough information for Search Service providers and consumers to create and use CDR-compliant Search Services.

The Search Service exposes a single Search function that is responsible for three activities that underpin Content Discovery capabilities: Search, Results Presentation, and Results Paging. This is discussed further in Section 2.

As discussed in the CDR Specification Framework, a Search Service's results are generally resource metadata rather than actual content resources. In the context of Search, resource metadata generally refers to a subset of a resource's available metadata, not the entire underlying record<sup>2</sup>. Some of the information contained within each Search result may provide the information necessary for a consumer to retrieve or otherwise use the referenced resource.

## 1.2 Scope

The Search Service, as defined, leverages the OpenSearch specification and its extensions. The definition of a more generic RESTful Search Service that does not build off OpenSearch is beyond the scope of the current specification.

## 1.3 Artifact Overview

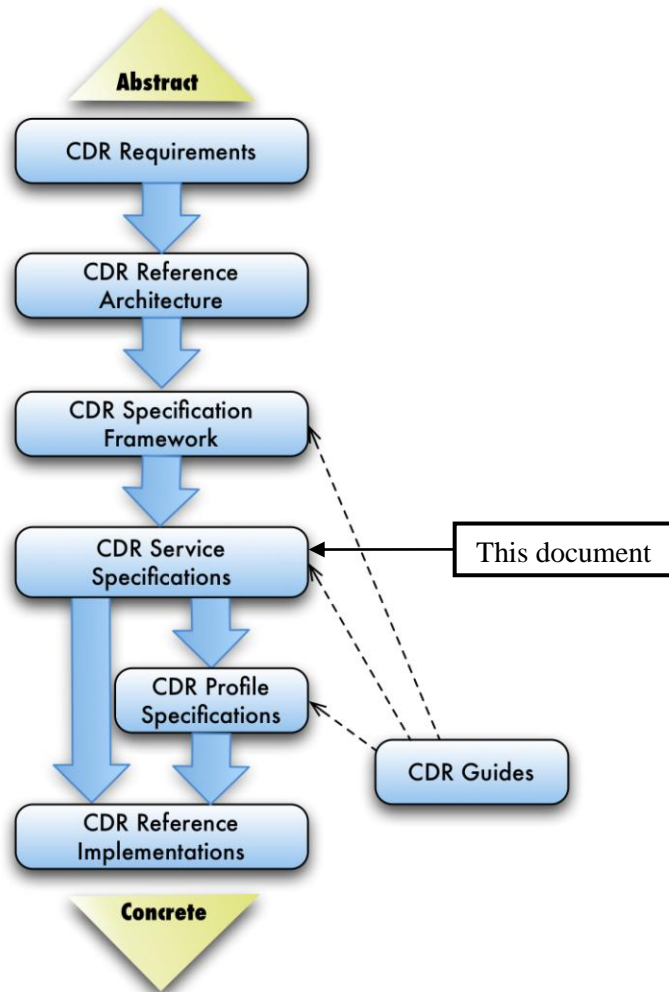
This specification is a part of the set of specifications that define the concrete, implementation-specific guidance for the services defined under the auspices of the Content Discovery & Retrieval (CDR) Integrated Project Team (IPT). The CDR Reference Architecture [CDR-RA] prescribes an abstract-to-concrete model for the development of architecture elements and guidance for content discovery and retrieval. Each layer or tier of the model is intended to provide key aspects of the overall guidance to achieve the goals and objectives for joint DoD/IC

---

<sup>1</sup> REST is an architectural style that encapsulates the design principles of the World Wide Web (WWW)

<sup>2</sup> The Search Component returns metadata about a resource, which may sometimes describe the underlying resource (e.g., an image), while at other times representing a sub-set of the data that makes up a resource (e.g., a collection of attributes). In some cases, the metadata returned from an instantiation of the Search function and the Retrieve function, which returns a resource itself, may happen to be the same, though this is considered an edge condition.

content discovery and retrieval. The following graphic, discussed in detail within the CDR Reference Architecture, illustrates this model.



**Figure 1: CDR Architecture Documents**

As illustrated in Figure 1, the CDR Specification Framework [CDR-SF] derives from the CDR Reference Architecture [CDR-RA] and describes behavior in terms of the capabilities, components, and usage patterns defined in the RA. Multiple CDR Service Specifications are derived from the CDR-SF, with separate specifications associated with the components of the architecture (e.g., Search) and, for each service, separate specifications to address Representational State Transfer (REST) and SOAP implementations.

This document is a specification for implementing the CDR Search Component using the RESTful OpenSearch [OS] standard. It is intended to parallel the corresponding SOAP specification, the IC/DoD SOAP Interface Encoding Specification for CDR Search [CDR-SS], as closely as possible, to minimize the difficulties in interoperating. Additional CDR Guides,

Profile Specifications, or Reference Implementations may provide additional guidance on implementing this specification in a particular context.

## 1.4 Notational Convention

The key words "MUST," "MUST NOT," "REQUIRED," "SHALL," "SHALL NOT," "SHOULD," "SHOULD NOT," "RECOMMENDED," "MAY," and "OPTIONAL" in this specification are to be interpreted as described in the IETF RFC 2119. When these words are not capitalized, they are meant in their natural-language sense.

When describing concrete XML schemas and example XML documents, this specification uses XPath as the notational convention. Each member of an XML schema is described using an XPath notation (e.g., `/x:RootElement/x:ChildElement/@Attribute`). The use of `{any}` indicates the presence of an element wildcard (`<xs:any/>`). The use of `@{any}` indicates the presence of an attribute wildcard (`<xs:anyAttribute>`).

In a URL template, a parameter contained in curly brace, generally represented in the form `{name}`, is meant to be replaced with an actual value determined at run-time. An optional parameter in a URL template is one whose name is followed by `?`, e.g., `{name?}` and it MAY be replaced by an empty string.

Examples in this text are distinguished by a black border. These are meant to be illustrative and only one way that the described syntax can be used.

```
<atom:entry>
  <atom:title>This is an example.</atom:title>
</atom:entry>
```

References as enumerated in Section 4 are indicated in the text by square brackets [ ].

## 1.5 Conformance

Search services MUST support OpenSearch 1.1 Draft 5 [OS]<sup>3</sup>.

This specification defines an interface to a Search Service to which an implementation and a subsequent deployment MUST conform. A deployment is an instance of an implementation. For an implementation to conform to this Search specification, it MUST adhere to all REQUIRED aspects of the specification.

---

<sup>3</sup> The current draft of the OpenSearch specification can be found at <http://www.opensearch.org/Specifications/OpenSearch/1.1/>. Previous drafts can be found by appending `Draft_n` to the end of the URL, where `n` is the draft of interest. This specification was originally written using `Draft_4` as its basis and is now based on `Draft_5`.



## 1.6 Namespaces

Namespaces referenced in this document and the prefixes used to represent them are listed in the following table. The namespace prefix of any XML Qualified Name (QName) used in any example in this document should be interpreted using the information below.

**Table 1. Referenced XML Namespaces**

Prefix	URI	Description
opensearch	<a href="http://a9.com/-/spec/opensearch/1.1/">http://a9.com/-/spec/opensearch/1.1/</a>	OpenSearch 1.1
atom	<a href="http://www.w3.org/2005/Atom">http://www.w3.org/2005/Atom</a>	Atom syndication format [ATOM]
relevance	<a href="http://a9.com/-/opensearch/extensions/relevance/1.0/">http://a9.com/-/opensearch/extensions/relevance/1.0/</a>	OpenSearch extension for relevance.[OS-RELV]

## 1.7 License

This specification is licensed under the Creative Commons Attribution-ShareAlike 2.5 Generic License (<http://creativecommons.org/licenses/by-sa/2.5/>), because it builds on the OpenSearch [OS] standard, which is licensed with the share-alike clause.

## 1.8 Security

This specification does not directly address security concerns. It will be necessary for any implementation of this specification to address security concerns relevant to the systems with which they interact and the corresponding governance bodies. Several aspects of search, to include returning only the results for which the requesting entity is authorized, should be addressed in the detailed security plan of an implementation, but are out of scope for this document.

## 2 Search Service Behavior

As defined in the CDR-SF, Search behavior is realized through three activities – search, results paging, and results presentation – and is accessed through the use of the Search Function interface. In this specification, the Results Paging Function is realized through use of the same Search Function interface. The results presentation activity supports the return of results from the search and paging functions but does not provide standalone functionality to warrant its own interface.

### 2.1 Search

The Search behavior accepts a Search Request<sup>4</sup> from the service consumer, identifies the query and processes it against the collection of information available to the Search Service to build a set of items, called Search Results, which are the Search Service’s response to the service consumer’s query.

---

<sup>4</sup> Precise definitions of “Search Request”, “Search Results”, “Query” and other search related terminology are included in the CDR-SF.

## 2.2 Paging

The Search Service may accept parameters that allow service consumers to request a particular subset of the Search Results. Depending on how it is implemented, the paging mechanism supported by Search Services may not guarantee continuity of search results while switching pages. Consequently, it may not be possible to guarantee consumers will be able to reconstruct the contents of the entire result set at a particular time. Data assets may be added, updated, or removed in the period of time between page requests. Therefore, service consumers should not assume continuity among paged result sets, unless such continuity is explicitly supported.

For OpenSearch, search results can be traversed using the information from the original Search Service request combined with the endpoint information provided by the OpenSearch Description Document (OSDD) which describes the Search Service from which the current result set was generated. The Search Service OSDD MAY define mechanisms that allow a service consumer to issue a paging request for another identifiable "page" of data, such as the next or previous page.

## 2.3 Result Presentation

The format, content, and ordering of the Search Results is referred to as Results Presentation. While the OpenSearch template does specify the format of the result set (e.g. HTML, RSS, or Atom), neither the metadata returned as part of the result set nor the metadata for each result is defined in this specification.

The ordering of results can be based on any attribute common across a result set, such as date created or relevance. Result relevance is generally a measure of how well a specific result matches the original query, and providing a result relevance measure allows better matched results to be prioritized relative to other results. Sorting preferences can be expressed in the response format used for results presentation.

# 3 Search Service Interface

The service interface contains the technical descriptions<sup>5</sup> of the functions through which the consumer will interact with the service. Support for input and output parameters for each function is described in the following tables in terms of what is expected of the Search Service and what is expected in terms of a consumer interacting with the service.

## 3.1 Search Function

A Search Service MUST implement the Search Function.

### 3.1.1 Preconditions

The following preconditions MUST be satisfied if the search function is to correctly process input and generate results and post-conditions as described.

---

<sup>5</sup> The Search Service is intended to conform as described by the Search Component section of the Specification Framework [CDR-SF].

1. The requester is authenticated and authorized according to applicable policy requirements for this function.
2. The Search Service implementation is capable of accepting and interpreting the search request as expressed per conventions documented in the OpenSearch specification.

### 3.1.2 Input

The Search Service is the application of an HTTP/HTTPS GET method (request) to a single information resource, as identified by a Uniform Resource Locator (URL).

#### 3.1.2.1 HTTP Method

The Search Function **MUST** use the HTTP GET method.

#### 3.1.2.2 URL Template

The Url element is defined in the OpenSearch specification as

```
<Url type="{type}" template="{template}"/>
```

where

type

Contains the MIME type of the result set format

template

Contains the query URL to be processed by the target search implementation according to the specified syntax rules

The query URL **MUST** duplicate the value substituted for {template} and **MUST** replace every instance of a template parameter (a parameter name within { }) with a value before the search request is called. An optional parameter is one whose name is followed by ? (e.g., {startPage?}) and it **MAY** be replaced by an empty string.

For example, the Url element in the OpenSearch Description Document could be:

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

The type parameter in the example specifies the return format of the result set to be Atom. The template parameter in the example makes use of the OpenSearch parameters searchTerms, startIndex, and count, where the example specifies searchTerms and count as **REQUIRED** and startIndex as **OPTIONAL**. A sample URL conforming to this template is shown in Section 3.1.2.5.

Significant parameters that are applicable to the URL template within an OSDD include:

**Table 2. OpenSearch Input Parameters**

Parameter Name and Description	Support
<p><b>/opensearch:searchTerms<sup>6</sup></b> One or more keywords for which the search is being performed.</p>	<p>Support by service as specified in Url element per Section 3.1.2.2. Support by consumer consistent with support provided by service.</p>
<p><b>/opensearch:startIndex<sup>7</sup></b> The index into the ordered Search Results of the first Search Result desired by the consumer. MUST be expressed as an integer greater than or equal to 1. Default value equals 1.</p>	<p>Support by service as specified in Url element per Section 3.1.2.2. Support by consumer consistent with support provided by service.</p>
<p><b>/opensearch:startPage<sup>8</sup></b> This is an alternate method of specifying the start index by providing the page number of the Search Results desired by the consumer. The @startPage minus 1, multiplied by @count, plus 1 gives the @startIndex.<sup>9</sup> Must be expressed as an integer greater than or equal to one. Default is to use @startIndex. <i>A Search service SHOULD NOT use both startIndex and startPage, since their functions overlap.</i></p>	<p>Support by service as specified in Url element per Section 3.1.2.2. Support by consumer consistent with support provided by service.</p>
<p><b>/opensearch:count<sup>10</sup></b> The number of results to include in the returned page of a result set. MUST be expressed as an integer greater than zero if used. The default value if not specified is 10. Search clients should anticipate that the value of @count may not be honored by the search engine (e.g., a search engine may have a maximum number it will return for one page), and should rely exclusively on the contents of the opensearch:itemsPerPage response element (see Table 3) in calculating actual page size.</p>	<p>Support by service as specified in Url element per Section 3.1.2.2. Support by consumer consistent with support provided by service.</p>

<sup>6</sup> Defined as part of the OpenSearch specification; see

[http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22searchTerms.22\\_parameter](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22searchTerms.22_parameter)

<sup>7</sup> Defined as part of OpenSearch specification; see

[http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22startIndex.22\\_parameter](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22startIndex.22_parameter) and [http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22startIndex.22\\_element](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22startIndex.22_element).

<sup>8</sup> Defined as part of OpenSearch specification; see

[http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22startPage.22\\_parameter](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22startPage.22_parameter).

<sup>9</sup>  $startIndex = (startPage - 1) * count + 1$ . For example, if startPage = 3 (i.e., the 3rd page of results) and count = 10 (i.e., there are 10 results per page), then startIndex = 21 (i.e., the 21st entry in the Results Set).

<sup>10</sup> Defined as part of OpenSearch specification; see

[http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22count.22\\_parameter](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22count.22_parameter).

While the majority of OpenSearch implementations provide keywords using the searchTerms parameter as the object of the search, OpenSearch supports a number of different ways to specify the query. For example, it may be possible that a geo or time term is sufficient to express a meaningful query, in which case a search implementation may allow the consumer to leave the searchTerms field empty. In such cases, the consumer of the Search Service **MUST** use these alternate terms if searchTerms is not used.

In addition to the parameters defined within the specification, OpenSearch provides a means to define extensions. Two extensions of particular note that have been defined are:

- OpenSearch Geo Extension [OS-GEO] – Defines parameters for supporting bounding box, point-radius, polygon, and other geospatial query types.
- OpenSearch Time Extension [OS-TIME] – Defines parameters for specifying a time range. This specification does not describe how to interpret the temporal query terms. For example, it does not specify that the temporal terms apply to the date a resource was created or posted, or that it may apply to a historical event described by the resource.<sup>11</sup>

The CDR IPT effort has developed the Brokered Search extension for OpenSearch Implementations [CDR-BS], and additional extensions may be developed by the CDR IPT in the future. If extensions are used that sufficiently provide an alternate means of specifying a query, these **MAY** be used in place of searchTerms, as discussed.

For further guidance regarding the Url element and the Url template syntax, refer to the OpenSearch specification.

### 3.1.2.3 HTTP Message Header

There are no additional entries in the request message header beyond those defined in the HTTP specification [HTTP].

### 3.1.2.4 HTTP Message Body

There is no request message body for this function.

### 3.1.2.5 Search Request - Message Example

```
http://example.com/?q=watson+ibm&startIndex=31&count=10
```

results in

```
GET /?q=watson+ibm&startIndex=31&count=10 HTTP/1.1  
Host: example.com
```

---

<sup>11</sup> Guidance related to interpretation can be found in implementation guidance, such as the DoD Discovery Metadata Specification Implementation Guide [DDMS].

### 3.1.3 Output

The output of the Search Service is a set of Results that describe resources matching the query provided in the Search Request. In addition to the requirements imposed by the CDR Specification Framework, the Search service Search function output is additionally constrained by the requirements specified in the OpenSearch Specification.

For requests that result in an error, a HTTP Error code MUST be returned.

#### 3.1.3.1 HTTP Status Code

If the GET request is successful, the service will respond with a '200 OK' Status Code and the content resource.

#### 3.1.3.2 HTTP Message Header

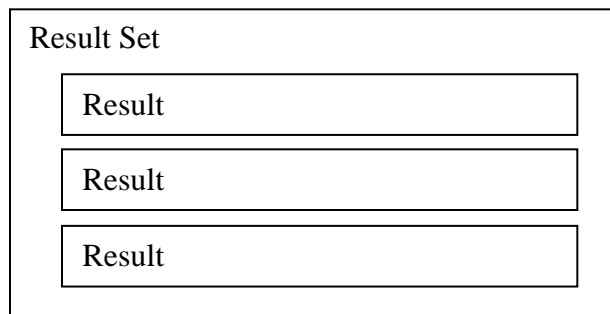
The response header SHOULD include the content type (i.e., the content-type header field with a MIME type as the assigned value) and content encoding of the content resource so that the consumer may anticipate how it should be processed and to inform any user agents (outside the scope of CDR) that may be available in the future to assist in mediating the referenced information resource to a preferred format.

#### 3.1.3.3 HTTP Message Body

The body of the HTTP message MUST consist of the result set consisting of zero or more results.

#### 3.1.3.4 Output Example

Figure 2 illustrates the high level components of a response message (containing a result set of unspecified type with zero or more results) from a Search service:



**Figure 2: Result Set Structure**

The results presentation SHOULD follow a documented Result Set specification.<sup>12</sup> Figure 3 represents a sample output using the Atom format for the result set augmented with OpenSearch response elements as identified in Table 3. Search service implementations SHOULD include these OpenSearch response elements, although the exact mechanism and syntax for including

<sup>12</sup> The CDR Specification set includes a single predefined Result Type definition that IC/DoD organizations can leverage in their Search service implementations, the IC/DoD Content Discovery and Retrieval Atom 1.0 Result Set Specification [CDR-ATOM]. Implementers SHOULD consult appropriate policy and implementation guidance to determine requirements or recommendations concerning the use of particular Result Types.

these elements may vary depending on the Result Set specification being used. Consult the Atom specification [ATOM] and the CDR Atom guide [CDR-ATOM] for specifics on using the Atom format.

```

HTTP/1.1 200 OK
Content-Length: 300
Content-Type: application/atom+xml

<atom:feed>
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</atom:id>
  <atom:title>Query Results for &quot;watson ibm&quot;</atom:title>
  <atom:updated>2003-12-13T18:30:02Z</atom:updated>
  <atom:author><atom:name>Enterprise Catalog</atom:name></atom:author>
  <opensearch:totalResults>492420</opensearch:totalResults>
  <opensearch:startIndex>1</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
  <atom:link rel="self"
    href="http://example.com/?q=watson+ibm&startIndex=1&count=10"
    type="application/atom+xml"/>
  <atom:entry>
    <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af7</atom:id>
    <atom:title>IBM - Watson</atom:title>
    <atom:updated>2011-02-21T00:00:00Z</atom:updated>
    <atom:link rel="alternate"
      href="http://www-03.ibm.com/innovation/us/watson/index.html"/>
    <relevance:score>0.97</relevance:score>
  </atom:entry>
  ...
</atom:feed>

```

Figure 3: Example output using Atom format

Table 3. OpenSearch Response Elements used within Atom Feed

Element Name and Description	Support
<b>/opensearch:startIndex</b> As defined in Table 2.	as specified in [CDR-ATOM]
<b>/opensearch:itemsPerPage<sup>13</sup></b> The number of items returned in a single response. This is expected to correspond to /opensearch:count as defined in Table 2.	as specified in [CDR-ATOM]
<b>/opensearch:totalResults<sup>14</sup></b> The actual or estimated number of resources that match the current query.	as specified in [CDR-ATOM]

<sup>13</sup> Defined as part of OpenSearch specification; see [http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22itemsPerPage.22\\_element](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22itemsPerPage.22_element)

<sup>14</sup> Defined as part of OpenSearch specification; see [http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5#The\\_.22totalResults.22\\_element](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22totalResults.22_element)

Figure 3 also illustrates use of atom:link with the rel="self" role to include the search request that produced the output. This is meant to facilitate paging capabilities and to provide service consumers the ability to re-execute their queries,

Also illustrated in Figure 3 is the use of the OpenSearch Relevance extension [OS-REV] defined in Table 4.

**Table 4. OpenSearch Result Extension**

Element Name and Description	Support
<p><b>/atom:entry/relevance:score</b> The relevance score for an entry. The range of values allowed is any decimal between 0 to 1, inclusive, with 1 being the most relevant and 0 the least.</p>	<p>as specified in [CDR-ATOM]</p>

This scheme does not define the mechanism by which the relevance score is determined. In addition, comparing scores calculated under this scheme by different Search service instances may not provide a true comparison of relevancy.

Depending on the underlying data resources and the type of search request being executed, Search services MAY return metadata about each resource beyond that required by the Result Set specification. That specification controls the mechanism and syntax for including any additional metadata and whether or not such inclusion is permitted. Search services that support a particular Result Set format in its response MUST follow the requirements in the associated Result Set specification.

### 3.1.4 Post-conditions

The following conditions MUST be met upon completion of a search.

1. The results available to be returned to the requester are relevant to the input query.
2. The response will consist of a list of results or an appropriate fault.
3. The results are in the correct format.
4. The authenticated requester has been authorized to receive each result in the response.
5. The use of this function has been audited according to applicable policy.<sup>15</sup>

### 3.1.5 Fault Conditions

The following are common fault conditions that a Search Service implementation SHOULD handle, and Table 5 indicates the HTTP status codes that SHOULD be referenced.

Related specifications, such as query extensions and Result Set types, MAY create additional Fault Conditions, as necessary.

Table 5 maps the CDR Specification Framework fault conditions to the HTTP status that SHOULD be returned for each.

<sup>15</sup> The use of this function may be audited according to applicable policy and may include auditing of the success or failure of the function.



**Table 5. Fault Conditions and HTTP Responses**

<b>CDR Framework Fault Condition</b>	<b>HTTP Status</b>	<b>HTTP Description</b>
Unauthorized Access	403	Forbidden
Unsupported Search Request Syntax	400	Bad Request
Unsupported Search/Query Property	400	Bad Request
Invalid Paging Value	400	Bad Request
Paging Value Out of Range	404	Not Found
Service Execution Fault	500	Internal Service Error

## **3.2 Results Paging Function**

A Search Service SHOULD implement the Results Paging Function.

Note, results paging functionality can be realized by using the same URL template as the Search Function and varying the values of `/opensearch:startIndex` or `/opensearch:startPage`. However, the caveat in Section 2.2 applies, i.e. there is no guarantee of continuity of search results across individual pages. While this specification describes how a consumer can make use of results paging if it is provided by the Search Service, it does not define how such paging is to be implemented.

### **3.2.1 Preconditions**

The following preconditions MUST be satisfied if the search function is to correctly process input and generate results and post-conditions as described.

1. The requester is authenticated and authorized according to applicable policy requirements for this function.
2. The Search Service implementation is capable of accepting and interpreting the search request as expressed per conventions documented in the OpenSearch specification.

### **3.2.2 Input**

The input for the Results Paging Function is identical to that described in Section 3.1.2 for the Search Function. The search parameters as identified in Table 2 and any other parameters, e.g. the OpenSearch Geo and Time Extensions, must have identical values as the initial search request EXCEPT for the `startIndex`, `startPage`, and `count` parameters whose values define the results included in the next page of the response.

Paging can also be accomplished through links provided with the current page of output to the first, last, previous, or next page of results. As an example, assume the full result set has 88 results and result 31 is the first entry of the current page with 10 results on the page. Then, in concert with the self-link, these additional paging links can be expressed in the context of an Atom formatted output as shown in Figure 4.

```
<atom:feed>
...
<atom:link rel="previous"
  href="http://example.com/?q=watson+ibm&startIndex=21&count=10"
  type="application/atom+xml"/>
<atom:link rel="next"
  href="http://example.com/?q=watson+ibm & StartIndex=41&count=10"
  type="application/atom+xml"/>
<atom:link rel="self"
  href="http://example.com/?q=watson+ibm & StartIndex=31&count=10"
  type="application/atom+xml"/>
<atom:link rel="first"
  href="http://example.com/?q=watson+ibm & StartIndex=1&count=10"
  type="application/atom+xml"/>
<atom:link rel="last"
  href="http://example.com/?q=watson+ibm & StartIndex=79&count=10"
  type="application/atom+xml"/>
...
</atom:feed>
```

**Figure 4: Paging Links used in Atom Feed**

### 3.2.3 Output

The output of results paging is identical to the output of the Search function as described in Section 3.1.3.

### 3.2.4 Post-conditions

The following conditions **MUST** be met upon completion of this function.

1. The response will consist of a list of results or an appropriate fault.
2. The results are in the correct format.
3. The authenticated requester has been authorized to receive each result in the response.
4. The use of this function has been audited according to applicable policy.<sup>16</sup>

### 3.2.5 Fault Conditions

Faults are the same as for Search Function; see Section 3.1.5.

---

<sup>16</sup> The use of this function may be audited according to applicable policy and may include auditing of the success or failure of the function.

## 4 References

- [ATOM] “The Atom Syndication Format”; IETF RFC 4287; 2005. Available at <http://www.ietf.org/rfc/rfc4287> .
- [CDR-ATOM] “IC/DoD Content Discovery and Retrieval Atom 1.0 Result Set Specification”, 1.0, March 2010
- [CDR-BS] “IC/DoD Content Discovery & Retrieval Brokered Search Service Specification for OpenSearch Implementations”, V1.0-20101025.
- [CDR-RA] “IC/DoD Content Discovery and Retrieval Reference Architecture”, 1.1, 25 Feb 2011
- [CDR-SF] “IC/DoD Content Discovery & Retrieval Specification Framework”, 1.0 DRAFT, 9 May 2011
- [CDR-SS] “IC-DoD SOAP Interface Encoding Specification for CDR Search”, 3.0, 2012
- [DDMS] “DoD Discovery Metadata Specification”, 3.0, 2010
- [HTTP] “Hypertext Transfer Protocol -- HTTP/1.1” <http://www.ietf.org/rfc/rfc2616.txt>, 1999.
- [OS] “OpenSearch 1.1 Draft 5”. Available at [http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft\\_5](http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5)
- [OS-GEO] “OpenSearch Geo Extension”  
[http://www.opensearch.org/Specifications/OpenSearch/Extensions/Geo/1.0/Draft\\_1](http://www.opensearch.org/Specifications/OpenSearch/Extensions/Geo/1.0/Draft_1), 1.0, Draft 1, 2009
- [OS-RELV] “OpenSearch Relevance Extension”,  
<http://www.opensearch.org/Specifications/OpenSearch/Extensions/Relevance/1.0>, 1.0, Draft 1, 2007
- [OS-TIME] “OpenSearch Time Extension”  
[http://www.opensearch.org/Specifications/OpenSearch/Extensions/Time/1.0/Draft\\_1](http://www.opensearch.org/Specifications/OpenSearch/Extensions/Time/1.0/Draft_1), 1.0, Draft 1, 2010

## Appendix A. Mapping to Specification Framework

This section explicitly ties the items in this specification to the requirements of the CDR-SF. The CDR-SF identifies the requirements for creating specifications, while implementation details are outlined in this document.

### A.1. Search Function

#### *Input*

Table 6 maps the Search Function inputs in the CDR-SF to the elements and attributes defined in this specification.

**Table 6: Search Function Input Mapping to Specification Framework**

Specification Framework Variable	REST Search Specification
Query	The query is an aggregate of the parameters in the request.
Query Properties	OpenSearch requests are assumed to be a general keyword search. Additional parameters may be supported using OpenSearch extensions, as indicated in the URL template parameters in an OSDD.
Search Properties Start Index (alternate to) Start Index Results Per Page Result Metadata Format  Timeout Result Sorting Order	<p>/opensearch:startIndex /opensearch:startPage /opensearch:count</p> <p>Implicit – Service <b>MUST</b> respond with the default results format and the default <b>SHOULD</b> be specified in its service description. If no default is specified, the Atom format <b>MUST</b> be used as the default. Therefore this input variable is not needed</p> <p><i>Not Supported</i></p> <p>Default sorting by relevancy is <b>RECOMMENDED</b>. Individual query types <b>MAY</b> define input variables or query syntax to control custom sorting; otherwise sorting order input is not supported and no variable is defined.</p>

#### *Output*

Mapping of the Search Function outputs in the CDR-SF depends on the Result Metadata Format that is associated through the Search Service binding. Table 7 relates the disposition of each variable defined in the Specification Framework to this specification.

**Table 7. Search Function Output Mapping to Specification Framework**

<b>Specification Framework Variable</b>	<b>REST Search Specification</b>
Result Set	A Search service <b>MUST</b> return a formatted set of results. Search service implementations <b>SHOULD</b> support HTML and Atom response formats.
Result Metadata	CDR Result Set specifications <b>MAY</b> require certain types of data to be returned as part of the result set or individual result entries. Those specifications <b>MAY</b> also allow other types of metadata to be included and describe the mechanism for doing so. A Search service that supports a particular result type <b>MUST</b> follow the syntax and processing rules defined by that type.
Result Relevancy	relevance:score in each result
Retrieve Properties	An element describing the linkage to the Retrieve service <b>SHOULD</b> be included in the results.
Timestamp	CDR Result Type may require timestamp element (i.e., feed/updated).
Query Identifier	<i>Not Supported – OpenSearch services typically use <code>opensearch:searchTerms</code> and other parameters defined in the URL template to identify a result set for paging purposes.</i>
Response Result Count	<code>/opensearch:itemsPerPage</code>
Total Result Count	<code>/opensearch:totalResults</code>

## **A.2. Results Paging Function**

### ***Input***

The mapping of inputs for the Results Paging Function is the same as shown in Table 6 for Search Properties and Table 7 for Query Identifier.

### ***Output***

The output for the Results Paging Function is identical to that shown for the Search Function in Table 7.

## Appendix B. Changes from Prior Version

This section outlines the significant changes that were made from *IC/DoD Content Discovery & Retrieval Search Service Specification for OpenSearch Implementations, V1.0.* to the current version. These changes, shown in Table 8, were made to accommodate changes in the CDR-RA and CDR-SF, to harmonize the content of this specification with that contained in the other specifications produced by this group, and to incorporate feedback on the specification from pilot implementations.

**Table 8: Summary of Changes from V1.0**

V1.0	V3.0	Rationale for Change
First service specification developed.	Structure of document revised to reflect experience gained in writing other service specs.	Leverage consistent structure to make specs easier to read.
Based on OpenSearch 1.1, Draft 4	Based on OpenSearch 1.1, Draft 5	Changes from Draft 4 to Draft 5 were not significant; alignment with latest draft avoids unnecessary reference to previous draft.
Main flow for service behavior in section 2.1.	Section removed.	Detail assumed more than needed about service implementation.
	Explanation surrounding OpenSearch Url element expanded.	Improve clarity of specification but should not affect implementation based on specification.
resultsPerPage part of input.	count replaces resultsPerPage.	Wrong OpenSearch attribute was used.
Generic output example fragments used.	Single Atom format example used.	More information in single example.
Contains details of OpenSearch paging elements in section 3.3.4.	Replaced by less detail from OpenSearch spec but link to specific information.	Less duplication of OpenSearch spec and more direct reference to normative document.
Security discussed in section 4.5.	Security statement in section 1.7.	Leverage standard security statement for all service specifications.
Implementation narrative in section 4.	Section deleted.	Relevant material moved to body of specification and speculative material removed.