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

IC-DoD SOAP 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 restriction

Revision History

Revision	Revised By	Date	Description
0.1		20 November 2009	Initial draft for subgroup review.
0.2		18 December 2009	All sections updated.
0.4.1		01 January 2010	All sections updated. Changed the revision number to reflect the associated RA and Specification Framework documents, respectively.
0.4.4		15 January 2010	All sections updated.
0.9.0	CDR IPT	22 January 2010	First draft release for CDR SOAP Working Group review.
0.9.1	CDR IPT	28 January 2010	All sections updated.
0.9.2	CDR IPT	04 February 2010	All sections updated.
0.9.3	CDR IPT	16 February 2010	Minor updates.
0.9.4	CDR IPT	19 February 2010	Minor updates.
0.9.5	CDR IPT	26 February 2010	Minor updates.
1.0-Milestone 1	CDR IPT	09 March 2010	Minor updates.
1.0-Milestone 1	Farley	23 April 2010	Tech Edit
2.0 Draft 1	McKnight	15 July 2011	Harmonization Update- incomplete
2.0 Draft 2	McKnight	22 July 2011	Harmonization Update- complete
2.0 Draft 3	McKnight	29 July 2011	Added references and completeness
2.0 Draft 4	McKnight	9 August 2011	Modified to accommodate feedback on draft 3 and consistency with schema and WSDL
2.0 Draft 5	McKnight	08 September 2011	Modified to accommodate changes from adjudication review
2.0 Draft 6	McKnight	20 September 2011	Modified to accommodate changes from adjudication review
2.0 Draft 7	McKnight	26 September 2011	Modified to accommodate changes from adjudication review
2.0 Draft 8	McKnight	06 October 2011	Modified for adjusted view of Query Language and QueryId
2.0 Draft 9	McKnight	12 October 2011	Modified for ODNI style guide concerns
2.0 Draft 10	McKnight	02 November 2011	Modified for CDR IPT comments
2.0 Draft 11	McKnight	28 November 2011	Modified for SCA comments
2.0 Draft 12	Wigglesworth	14 December 2011	Modified for ASEC comments
3.0	Laskey	28 March 2012	Re-add queryID equivalent; synchronize with changes in other specs

Table of Contents

List of Figures

List of Tables

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 common service interfaces and a 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 web service using SOAP[1], hereafter termed a Search Service in this document. The content of this specification describes the Search Service's behavior, interface and other aspects in detail, providing enough information for Search Service providers and consumers to create and use CDR-conformant Search Services.

The Search Service exposes one required function and one optional function that together realize the 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.[2] 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, enables flexibility in identifying both the query language in which the query is expressed and the format of the results returned in the response. Section 1.7 discusses identifying query languages and separate guidance for the use of Atom is indicated throughout as a results format. However, the specification of specific query languages or results formats 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

[1] SOAP is a protocol used by web services in the exchange of structured information.[SOAP]

[2] The Search Component returns metadata about a resource, which may sometimes describe the underlying resource (e.g., an image), while 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.

provide key aspects of the overall guidance to achieve the goals and objectives for joint DoD/IC content discovery and retrieval. The following graphic, discussed in detail within the CDR Reference Architecture, illustrates this model.

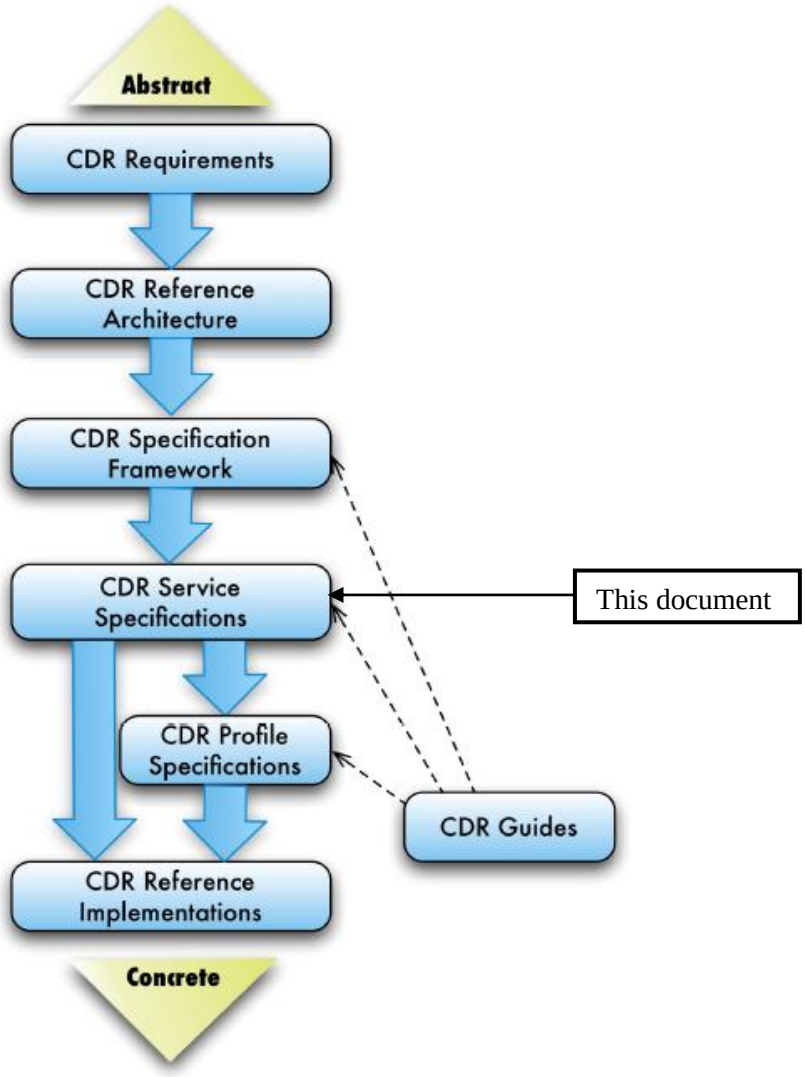

Figure 1: CDR Architectural Model

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 Service as a SOAP Web Service. It is intended to parallel the corresponding REST specification, the IC/DoD REST

Interface Encoding Specification for CDR Search [CDR-OS], 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. Optional parameters in a URL template are those whose name is followed by ?, e.g., {name?} and these 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

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.

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
atom	http://www.w3.org/2005/Atom	Atom syndication format
cdrs	urn:cdr:search:3.0	CDR IPT Search binding for SOAP implementations
opensearch	http://a9.com/-/spec/opensearch/1.1/	OpenSearch specification [OS] for search web services
relevance	http://a9.com/-/opensearch/extensions/relevance/1.0/	OpenSearch extension for relevance.[OS-rel]
soap	http://www.w3.org/2003/05/soap-envelope	SOAP 1.2 Envelope
wsa	http://www.w3.org/2005/08/addressing	WS-Addressing
xs	http://www.w3.org/2001/XMLSchema	XML Schema

1.7 Query Language URIs

Table 2 lists the URIs for query languages that are currently defined as part of or recognized by the CDR specification set. These are acceptable values for the `/Expression/@queryLanguage` attribute defined in Table 5. Table 2 references the query languages by name and specifies a URI (i.e., a URL or URN) that uniquely identifies the query language. URLs MUST link (or resolve) to a resource that normatively defines the query language; URNs MUST be associated with a means to retrieve a normative definition of the query language.

Table 2. Query Language URIs

Name	URI	Description
Keyword	urn:cdr:search:query:keyword	Generic definition for keyword queries
XQuery	http://www.w3.org/TR/xquery/	XQuery Recommendation
OGC Filter	http://www.opengis.net/ fes/2.0	Open Geospatial Consortium Search filter

Additional acceptable query language values may be defined in the future and MUST also be identified by Name and by a URI that is associated with a normative specification of the new query language.

Profile Specifications that describe how to utilize this Search specification in the context of the use of one of the query languages identified in Table 2 or defined in the future may be added to the CDR collection of artifacts.

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 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 and the Results Paging 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[3] 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.

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.

2.3 Results Presentation

The format, content, and ordering of the Search Results is referred to as Results Presentation. The support of a Search Service for results presentation can range from a single default format to support for alternate response formats from which the consumer could indicate a preference [4].

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 or a sort order can be expressed in the query expression itself.

3 Search Service Interface

The service interface contains the technical descriptions[5] of the functions through which the consumer will interact with the service. Support for input and output parameters for each

[3] Precise definitions of "Search Request", "Search Results", "Query" and other search related terminology are included in the CDR-SF.

[4] The availability and access to the information to be presented MAY also be controlled on the basis of the type of data being represented in the items and the authorization of the requestor to that data.

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

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.

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 and, in particular, the search request expression.

3.1.2 Input

The input to the CDR Search Service MUST be a valid SOAP[6] message that meets criteria identified in this section. The input should be directed to the SOAP Endpoint address identified by the implementer.

3.1.2.1 Header

The header of the SOAP message MUST contain the Action element, as defined in WS-Addressing [WS-A]. The purpose of this element is to convey to the service which behavior to invoke. Additional elements, such as other WS-Addressing elements, MAY be added to the SOAP header.

Table 3. Header Elements for Search Requests

Element Name and Description	Support
/wsa:Action This element (whose content is of type xs:anyURI) conveys the value of the [action] property and indicates to a web service which operation should be invoked.	MUST be supported by Service. MUST be provided by Consumer with a value of `urn:cdr:search:3.0:request`

3.1.2.2 Body

The body of the SOAP message must contain a single `/cdrs:SearchRequest` element, as defined in this document. The `/cdrs:SearchRequest` element contains the `/cdrs:Expression` element, which provides the query expression. Table 4 shows the attributes of the `/cdrs:SearchRequest`:[7]

[6] Consult the relevant standards registry (such as the ICSR or DISR) to determine the appropriate current version of the SOAP standard to use. Examples in this document use SOAP 1.2.

[7] See Appendix C "Changes From Prior Version" for a disposition of the request parameters from the prior version of this specification.

Table 4. Attributes of /cdrs:SearchRequest Element

Attribute Name and Description	Support
`/cdrs:SearchRequest/@startIndex`[8] 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.	MAY be supported by Service. MAY be provided by Consumer if supported by Service.
`/cdrs:SearchRequest/@startPage`[9] 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.[10] Must be expressed as an integer greater than or equal to one. Default is to use @startIndex. *A Search service SHOULD NOT use both startIndex and startPage, since their functions overlap, and the use of startIndex SHOULD take precedence.*	MAY be supported by Service. MAY be provided by Consumer.
`/cdrs:SearchRequest/@count`[11] 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 7) in calculating actual page size.	MAY be supported by Service. MAY be provided by Consumer if supported by Service.
`/cdrs:SearchRequest/@responseFormat` The response format desired by the client. If provided, the value MUST be a URI that corresponds to a definition of a search response format defined as part of the CDR specifications. If no alternate representations are requested by the Consumer, the Service MUST respond with its default results format and the default SHOULD be specified in its service description. If no default is specified, the Atom format [CDR-ATOM] MUST be used as the default.	MAY be supported by Service. MAY be provided by Consumer.

[8] Defined as part of OpenSearch specification; see 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 .

[9] Defined as part of OpenSearch specification; see http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22startPage.22_parameter .

[10] 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)..

[11] Defined as part of OpenSearch specification; see http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22count.22_parameter .

/cdrs:SearchRequest/@timeout The desired timeout period in milliseconds. If the timeout attribute is supported by the Service and provided by the Consumer, when the response time for the service exceeds the timeout value, the Service MUST respond with whatever partial results are available and a status indicating partial results MUST be indicated in the metadata of the results.	MAY be supported by Service. MAY be provided by Consumer.

Additional extension attributes MAY be supported by the Search Function and/or provided by the consumer to convey additional Search properties. However, any additional attributes supported by the Search Function MUST be OPTIONAL for the consumer to provide. If the consumer provides extension attributes that the Search Function does not support, these MUST be ignored.

The `/cdrs:SearchRequest` element has one child element, `/Expression`, which contains the query expression.

Table 5. Child Elements of /cdrs:SearchRequest Element

Element Name and Description	Support
/cdrs:SearchRequest/Expression The element provides the query expression for which the Search Service must return matching Search Results. The value MUST be of type string and the format of the value MUST be consistent with the query language referenced by the @queryLanguage attribute.	MUST be supported by Service. MUST be provided by Consumer.
/cdrs:SearchRequest/Expression/@queryLanguage The URI that identifies the query language in which the query is expressed.	MUST be supported by Service. MUST be provided by Consumer.

The query language (see Section 1.7.) used in the `/cdrs:SearchRequest/Expression` element MUST be a query language supported by the Search Service. The query language MAY allow the expression to contain information relating to result set sorting or ordering. Additional extension elements and/or attributes MAY be supported by the Search Service and/or provided by the consumer to convey additional Query Properties. However, any additional elements or attributes supported by the Search Service MUST be OPTIONAL for the consumer to provide. If the consumer provides additional elements or attributes that the Search Service does not support, these MUST be ignored.

3.1.2.3 Example

An example showing an XML document fragment with the SOAP envelope is shown in Figure 2.

```
<soap:Envelope>
  <soap:Header>
    <wsa:Action>urn:cdr:search:3.0:request</wsa:Action>
  </soap:Header>
  <soap:Body>
      <cdrs:SearchRequest startIndex=”1” count=”10”
      responseFormat=”urn:cdr:1.0:resultset:atom-1.0”>
         <cdrs:Expression queryLanguage=”urn:cdr:queryLanguage:keyword”>
            watson ibm
         </cdrs:Expression>
      </cdrs:SearchRequest>
  </soap:Body>
</soap:Envelope>
```

Figure 2: Example Search Input

This example shows a search for the keywords “watson” and “ibm”, using the Keyword query language, with an Atom response format requested.

3.1.3 Output

The output of the Search Service is a page of a Result Set, comprised of Results that describe resources matching the query provided in the Search Request. The format of the response is determined by the `/cdrs:SearchRequest/@responseFormat` attribute. This section describes a SOAP message format that can be used to encapsulate a pluggable, separately specified response format. It uses the [CDR-ATOM] response format as the basis for the examples in this section. For requests that result in an error, a SOAP fault message will be output.

3.1.3.1 Header

The header of the SOAP message MUST contain the Action element, as defined in WS-Addressing. The purpose of this element is to convey to the receiver which behavior was invoked.

Table 6. Required Header Elements for Search Responses

Element Name and Description	Support
`/wsa:Action` This element (content is of type xs:anyURI) conveys the value of the [action] property and indicates to a web service which operation should be invoked.	MUST be supported by Service. MUST be provided by the Consumer with a value of `urn:cdr:search:3.0:response`

Additional elements, such as other WS-Addressing elements, MAY be added to the SOAP header.

3.1.3.2 Body

The body of the SOAP message MUST consist of a single element, representing a page of the result set. The name and structure of the element is defined by the

`/cdrs:SearchRequest/@responseFormat`. The details in Figure 3 and Table 7 show this in the context of the Atom 1.0 Result Set Specification [CDR-ATOM]. Using this response format, the single element of the body of the SOAP message is `/atom:feed`. The child element definitions shown in Table 7 that correspond with those defined in the OpenSearch Specification are introduced in [CDR-ATOM]. If a different response format[12] is specified by `/cdrs:SearchRequest/@responseFormat`, the appropriate information as defined by that response format MUST be returned.

Table 7. Result Set Elements within Atom Feed

Element Name and Description	Support
`/opensearch:startIndex` As defined in Table 4.	as specified in [CDR-ATOM]
`/opensearch:itemsPerPage`[13] The number of items returned in a single response. This is expected to correspond to `/cdrs:SearchRequest/@count` as defined in Table 4.	as specified in [CDR-ATOM]
`/opensearch:totalResults`[14] The actual or estimated number of resources that match the current query.	as specified in [CDR-ATOM]
`/cdrs:resultSetID`[15] An identifier used as part of Results Paging (Section 3.2) to identify a result set corresponding to the search request.	as specified in [CDR-ATOM]

Table 7 also includes `/cdrs:resultSetID` as an OPTIONAL output for subsequent use with results paging.[16] In addition, the Search Service MAY echo the `/cdrs:SearchRequest` within the `/atom:feed`.

The element containing the result set SHOULD contain a series of result elements, each representing one resource that matched the query provided in the Search Request. In the case of the Atom Syndication Format [ATOM], the element corresponding to a Result is `/atom:entry`. This Result contains the metadata necessary to identify and retrieve the referenced resource. As defined in Table 8 and shown in Figure 3, the result MAY also include a relevance score represented in the form of the OpenSearch relevance extension [OS-rel].

---

[12] [CDR-S] examples show a more generic output skeleton within a single <cdss:SearchResponse> element.
[13] Defined as part of OpenSearch specification; see http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22itemsPerPage.22_element
[14] Defined as part of OpenSearch specification; see http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5#The_.22totalResults.22_element
[15] resultSetID replaces queryID that was used in previous CDR documents. The change in name is to better reflect the function of the identifier because it relates to the result set and not the query. A similar decision was made for using savedSearchID in Query Management.
[16] The use of identifiers is discussed in section 2.4 of [CDR-SF]. The format of and the mechanism for generating identifiers is currently out of scope for both [CDR-SF] and this specification.

Table 8. OpenSearch Result Extension

Element Name and Description	Support
`/atom:entry/relevance:score` 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.	as specified in [CDR-ATOM]

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.

3.1.3.3 Output Example

The following represents a Sample Output embedding the Atom Syndication Format style result set into the body of a SOAP message. Consult the Atom specification [ATOM] and the CDR Atom guide [CDR-ATOM] for specifics on using the Atom format.

```
<soap:envelope>
  <soap:header>
    <wsa:action>urn:cdr:search:3.0:response</wsa:action>
  </soap:header>
  <soap:body>
    <atom:feed>
      <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6</atom:id>
      <atom:title>Query Results for &amp;quot;watson ibm&amp;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>
       <cdrs:resultSetID>
          e86c698a-d922-445d-9785-85f2ba22a83
       </cdrs:resultSetID>
       <cdrs:SearchRequest startIndex="1" count="10"
       responseFormat="http://www.w3.org/2005/Atom">
         <cdrs:Expression queryLanguage="urn:cdr:queryLanguage:keyword">
            watson ibm
         </cdrs:Expression>
       </cdrs:SearchRequest>
       <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>
  </soap:body>
</soap:envelope>
```

Figure 3: Example Search Output

3.1.4 Post-Conditions

The following conditions MUST be met upon successful 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 this function has been audited according to applicable policy.[17]

3.1.5 Fault Conditions

An implementation of the Search function MAY provide any of the faults listed in Table 9 as a SOAP Fault to the consumer. The SOAP message MUST contain the header entry of `<wsa:Action>http://www.w3.org/2005/08/addressing/fault</wsa:Action>` and MUST contain a single SOAP Fault element as the only child of the SOAP Body element.

An example of the Search SOAP fault is shown in Figure 4. The fault response adheres to the SOAP 1.2 specification [SOAP]. The `/soap:Fault/Code/Value` element is used to convey the general type of error condition and MUST be from the enumeration `/soap:faultCodeEnum` as described in the SOAP 1.2 specification section 5.4.6. In addition, for the current use of this function, the fault MUST also contain the `/soap:Fault/Code/Subcode` element and its child element `/soap:Fault/Code/Subcode/Value`. The `/soap:Fault/Code/Subcode/Value` as listed in Table 9 supports automated processing of CDR specific errors. The `/soap:Fault/Reason` element has one or more `/soap:Fault/Reason/Text` elements as its children, where the value as listed in Table 9 for each `/soap:Fault/Reason/Text` element SHOULD be used to provide a human-readable explanation of the fault. The `/soap:Fault/Reason/Text` element MUST include the xml:lang attribute.

The following table outlines the service specific fault conditions that MAY be generated by CDR Search function implementations.

Table 9. List of Search Function Faults

/soap:Fault/Code/Value **/soap:Fault/Code/Subcode/Value** **/soap:Fault/Reason/Text**	**Fault Description**
soap:Sender cdr:search:soap:fault:security Unauthorized Access	The Consumer is either not authenticated or not authorized to perform the search query.
soap:Sender cdr:search:soap:fault:property Unsupported Search Property	The Search Service does not support the Search Property, such as timeout, that was specified.

[17] The use of this function may be audited according to applicable policy and may include auditing of the success or failure of the function.

soap:Sender cdr:search:soap:fault:syntax Unsupported Search Request Syntax	The Search Request expression syntax is not valid in accordance with the specified Query Language.
soap:Sender cdr:search:soap:fault:execution Service Execution Fault	The Search Service encounters an error during query execution.
soap:Sender cdr:search:soap:fault:qproperties Unsupported Query Properties	The Search Service does not support one or more of the query properties.
soap:Sender cdr:search:soap:fault:pagingValue Invalid Paging Value	The Start Index and/or Count values are not valid values (e.g., non-integer, negative).
soap:Sender cdr:search:soap:fault:pagingRange Paging Value Out of Range	Combination of paging values requests results outside of available range
soap:Sender cdr:search:soap:fault:resultFormat Unsupported Result Format	The Search Service does not support the result format specified by the response Format input parameter.
soap:Sender cdr:search:soap:fault:sorting Unsupported Result Sorting	The Search Service does not support the result sorting mechanism specified by the sort input parameter.

3.1.5.1 Fault Message Example

The following shows a fault message of type "Unsupported Search Request Syntax".

```
<soap:Envelope>

  <soap:Header>
      <wsa:Action>http://www.w3.org/2005/08/addressing/fault</wsa:Action>
  </soap:Header>

  <soap:Body>
    <soap:Fault>
       <soap:Code>
          <soap:Value>soap:Sender</soap:Value>
          <soap:Subcode>
            <soap:Value>cdr:search:soap:fault:syntax</soap:Value>
          </soap:Subcode>
       </soap:Code>
       <soap:Reason>
         <soap:Text xml:lang="en"> Unsupported Search Request Syntax
</soap:Text>
       </soap:Reason>
    </soap:Fault>
  </soap:Body>
</soap:Envelope>
```

Figure 4: Example Search SOAP Fault

3.2 Results Paging Function

A Search Service MAY implement the Results Paging Function.

Note, results paging functionality can be realized by using the Search Function and varying the values of `/cdrs:SearchRequest/@startIndex` or `/cdrs:SearchRequest/@startPage`. However, the caveat in Section 2.2 applies, i.e. there is no guarantee of continuity of search results across individual pages. The use of resultSetID as described in this section for use as part of the Results Paging Function MUST provide that continuity, e.g., if resultSetID points to a results set cache and results are selected from that cache. However, 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 results paging 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 returned a resultSetID in its initial response to the Search Request.
3. The results set identified by the resultSetID must still be accessible through reference to the resultSetID.

3.2.2 Input

The input to the CDR Search Service MUST be a valid SOAP[18] message that meets criteria identified in this section. The input should be directed to the SOAP Endpoint address identified by the implementer.

3.2.2.1 Header

The header of the SOAP message MUST contain the Action element, as defined in WS-Addressing [WS-A]. The purpose of this element is to convey to the service which behavior to invoke. Additional elements, such as other WS-Addressing elements, MAY be added to the SOAP header.

Table 10. Header Elements for Results Paging Requests

Element Name and Description	Support
/wsa:Action This element (whose content is of type xs:anyURI) conveys the value of the [action] property and indicates to a web service which operation should be invoked.	MUST be supported by Service. MUST be provided by Consumer with a value of `urn:cdr:search:3.0:paging`

[18] Consult the relevant standards registry (such as the ICSR or DISR) to determine the appropriate current version of the SOAP standard to use. Examples in this document use SOAP 1.2.

3.2.2.2 Body

The body of the SOAP message MUST contain a single `/cdrs:PagingRequest` element, as defined in this document. The `/cdrs:PagingRequest` element contains the `/cdrs:resultSetID` element. Table 11 shows the attributes of the `/cdrs:PagingRequest`:

Table 11. Attributes of /cdrs:PagingRequest Element

Attribute Name and Description	Support
`/cdrs:PagingRequest/@startIndex` Defined in Table 4 as applied to cdrs:SearchRequest.	MUST be supported by Service if supporting results paging. MAY be provided by Consumer.
`/cdrs:PagingRequest/@startPage` Defined in Table 4 as applied to cdrs:SearchRequest.	MAY be supported by Service. MAY be provided by Consumer.
`/cdrs:PagingRequest/@count` Defined in Table 4 as applied to cdrs:SearchRequest.	MUST be supported by Service if supporting results paging. MAY be provided by Consumer.
`/cdrs:SearchRequest/@responseFormat` Defined in `Table 4` as applied to cdrs:SearchRequest.	MAY be supported by Service. MAY be provided by Consumer.

Additional extension attributes MAY be supported by the Results Paging Function and/or provided by the consumer to convey additional Results Paging properties. However, any additional attributes supported by the Results Paging Function MUST be OPTIONAL for the consumer to provide. If the consumer provides extension attributes that the Results Paging Function does not support, these MUST be ignored.

The `/cdrs:PagingRequest` element has one REQUIRED child element, `/resultSetID`.

Table 12. Child Elements of /cdrs:PagingRequest Element

Element Name and Description	Support
`/cdrs:PagingRequest/resultSetID` Defined in Table 7.	MUST be supported by Service. MUST be provided by Consumer.

3.2.2.3 Example

An example of a SOAP message constituting a Results Paging request follows:

```
<soap:Envelope>
  <soap:Header>
    <wsa:Action>urn:cdr:search:3.0:paging</wsa:Action>
  </soap:Header>
  <soap:Body>
      <cdrs:PagingRequest startIndex=”1” count=”10”
      responseFormat=” urn:cdr:1.0:resultset:atom-1.0”>
         <cdrs:resultSetID>e86c698a-d922-445d-9785-85f2ba22a83</cdrs:resultSetID>
      </cdr:PagingRequest>
  </soap:Body>
</soap:Envelope>
```

Figure 5: Example Results Paging Input

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 successful completion of a search.

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.[19]

3.2.5 Fault Conditions

An implementation of the Results Paging function MAY provide any of the faults listed in Table 9 as a SOAP Fault to the consumer. Otherwise, the format of the SOAP Fault is the same as defined in section 3.1.5.

Table 13. List of Results Paging Function Faults

/soap:Fault/Code/Value /soap:Fault/Code/Subcode/Value /soap:Fault/Reason/Text	Fault Description
soap:Sender cdr:search:soap:fault:security Unauthorized Access	The Consumer is either not authenticated or not authorized to perform the search query.
soap:Sender cdr:search:soap:fault:syntax Unsupported Search Request Syntax	The Search Request expression syntax is not valid in accordance with the specified Query Language.

[19] The use of this function may be audited according to applicable policy and may include auditing of the success or failure of the function.

soap:Sender cdr:search:soap:fault:execution Service Execution Error	The Search Service encounters an error during query execution.
soap:Sender cdr:search:soap:fault:resultSetID Invalid ResultSetID	The resultSetID is not (no longer) valid.
soap:Sender cdr:search:soap:fault:pagingRange Paging Value Out of Range	The Search Service encounters an error during results paging execution.
soap:Sender cdr:search:soap:fault:pagingValue Invalid Paging Value	The Start Index and/or Count values are not valid values (e.g., non-integer, negative).
soap:Sender cdr:search:soap:fault:resultFormat Unsupported Result Format	The Search Service does not support the result format specified by the response Format input parameter.

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-OS] "IC/DoD REST Interface Encoding Specification for CDR Search3.0."; 2012.
[CDR-RA] "IC/DoD CDR IPT Reference Architecture 1.1."; 2011.
[CDR-S] "IC/DoD CDR Search Specification for SOAP Implementations 1.0"; 2010.
[CDR-SF] "IC/DoD Content Discovery & Retrieval Specification Framework 1.0."; 2011.
[OS] "OpenSearch 1.1 Draft 5". Available at http://www.opensearch.org/Specifications/OpenSearch/1.1/Draft_5
[OS-rel] "OpenSearch Relevance Extension". Available at http://www.opensearch.org/Specifications/OpenSearch/Extensions/Relevance/1.0
[SOAP] "SOAP Version 1.2 Part 1: Messaging Framework. Available at http://www.w3.org/TR/soap12-part1/.
[WS-A] "Web Services Addressing 1.0 - SOAP Binding"; 2006. Available at http://www.w3.org/TR/ws-addr-soap/

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

a. Input

Table 14 maps the Search Function inputs in the CDR-SF to the elements and attributes defined in this specification.

Table 14. Search Function Input Mapping to Specification Framework

Specification Framework Variable	SOAP Search Specification
Query	`/cdrs:SearchRequest/Expression`
Query Properties	`/cdrs:SearchRequest/Expression /@queryLanguage`
Search Properties	
Start Index	`/cdrs:SearchRequest/@startIndex`
(alternate to) Start Index	`/cdrs:SearchRequest/@startPage`
Results Per Page	`/cdrs:SearchRequest/@count`
Result Metadata Format	`/cdrs:SearchRequest/@responseFormat`
Timeout	`/cdrs:SearchRequest/@timeout`
Result Sorting Order	`(not supported)`

b. Output

Mapping of the Search Function outputs in the CDR-SF depends on the Result Metadata Format (i.e. responseFormat) that is chosen on input or is the default for the Search Service. Table 15 shows the mapping for a responseFormat conforming to the Atom 1.0 Result Set Specification [CDR-ATOM].

Table 15. Search Function Output Mapping to Specification Framework

Specification Framework Variable	SOAP Search Specification
Result Set	`/atom:feed`
Result Metadata	`/atom:entry/{any}`
Result Relevancy	`/atom:entry/relevance:score`
Retrieve Properties	`/atom:entry/atom:link`
Timestamp	`/atom:feed/atom:updated`
Query Identifier	`/cdrs:resultSetID`
Response Result Count	`/opensearch:itemsPerPage`
Total Result Count	`/opensearch:totalResults`

A.2. Results Paging Function

a. Input

The mapping of inputs for the Results Paging Function is the same as shown in Table 14 for Search Properties and Table 15 for Query Identifier.

b. Output

The output for the Results Paging Function is identical to that shown for the Search Function in Table 15.

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 SOAP Implementations, V1.0***
To the current version. These changes, shown in Table 16, 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 16: 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
Specific conformance statements in Section 1.3	More general conformance statement in terms of MUST	Remove implied differences between specific statements and use of MUST
	Updated namespaces in section 1.5	Included namespaces used but not defined
Main flow for service behavior in section 2.1.	Section removed.	Detail assumed more than needed about service implementation.
	WS-Addressing Action header added.	Common means to convey service behavior
	Query Languages introduced in section 1.6.	Provide unambiguous indication of how to interpret query expression
queryTypeURI attribute for searchRequest; Query Metadata specified as part of Query Type	queryLanguage added as attribute of Expression; Query Properties added to replace Query Type and Query Metadata	Query Properties replaces the previously separate inputs of Query Type and Query Metadata. There was not adequate distinction in the information that could be considered for the separate inputs, and the single input provides sufficient opportunity to attach information to assist at characterizing the query.
itemsPerPage part of searchRequest	count replaces itemsPerPage	Wrong OpenSearch attribute was used.
queryID used to identify previously executed request	resultSetID replaces queryID	More descriptive name: identifier is used to control moving through Result Set.
Generic output example fragments used.	Single Atom format example used.	More information in single example.
Single Search interface defined and combined use with Results Paging.	Separate interfaces defined and related.	Greater clarity.
	Updated faults and output examples.	Greater clarity.
Security discussed in section 4.4.	Security statement in section 1.7.	Leverage standard security statement for all service specifications.
Specification Usage in section 4.	Section deleted.	Relevant material moved to body of specification and speculative material removed.