

**UNCLASSIFIED**

IC/DoD Query Management Service Specification for REST Implementations  
Version 1.0-20111214, December 14 2011

---



---

**Intelligence Community and Department of Defense**

**Content Discovery & Retrieval Integrated Project Team**

**IC-DoD Representational State Transfer (REST)  
Interface Encoding Specification for CDR Query  
Management**

---

**V1.0-20111214**

**14 December 2011**

**UNCLASSIFIED**

**UNCLASSIFIED**

**REVISION/HISTORY**

Doc Revision	Revised By	Revision Date	Revisions
V1.0	CDR-IPT	August 2011	Initial release

**UNCLASSIFIED**

## TABLE OF CONTENTS

<b>1. Introduction .....</b>	<b>1</b>
1.1 Service Overview .....	1
1.2 Relationship to Other CDR Architecture Elements .....	3
1.3 Scope .....	4
1.4 Notational Convention .....	4
1.5 Conformance .....	5
1.6 Namespaces .....	5
<b>2. Service Behavior .....</b>	<b>6</b>
<b>3. Service Interfaces.....</b>	<b>7</b>
3.1 QM-Create Function .....	9
3.1.1 Preconditions .....	9
3.1.2 Request.....	9
3.1.3 Response .....	14
3.1.4 Post-conditions.....	17
3.2 QM-Read Function .....	17
3.2.1 Preconditions .....	17
3.2.2 Request .....	18
3.2.3 Response .....	19
3.2.4 Post-conditions.....	21
3.3 QM-Update Function.....	21
3.3.1 Preconditions .....	22
3.3.2 Request .....	22
3.3.3 Response .....	25
3.3.4 Post-conditions.....	27
3.4 QM-Delete Function.....	27
3.4.1 Preconditions .....	27
3.4.2 Request .....	28
3.4.3 Response .....	29
3.4.4 Post-conditions.....	29
3.5 QM-Execute Function.....	30
3.5.1 Preconditions .....	30
3.5.2 Request .....	30
3.5.3 Response .....	32
3.6 QM-Search Function.....	34
3.6.1 Preconditions .....	34
3.6.2 Request .....	34

3.6.3 Response ..... 35

**4. References ..... 37**

4.1 Content Discovery and Retrieval Specifications..... 37

## LIST OF FIGURES

Figure 1. Query Management Resource Model .....	2
Figure 2. CDR Architecture Model .....	3
Figure 3. Example Notation Convention .....	5
Figure 4. Create Request.....	13
Figure 5. Create Response .....	16
Figure 6. Read Request .....	18
Figure 7. Read Response.....	21
Figure 8. Update Request.....	24
Figure 9. Update Response .....	26
Figure 10. Delete Request.....	28
Figure 11. Delete Response .....	29
Figure 12. Execute Request .....	31
Figure 13. Execute Response.....	33
Figure 14. Search Request .....	35
Figure 15. QM Search Response .....	36

## LIST OF TABLES

Table 1. Namespaces .....	5
Table 2. Query Management Service Function Requirements .....	8
Table 3. Summary of Rest Query Management Functions.....	9
Table 4. Specification Framework To Service Specification Mapping .....	14
Table 5. Specification Framework To Service Specification Mapping .....	17
Table 6. Specification Framework To Service Specification Mapping .....	19
Table 7. Specification Framework To Service Specification Mapping .....	21
Table 8. Specification Framework To Service Specification Mapping .....	25
Table 9. Specification Framework To Service Specification Mapping .....	27
Table 10. Specification Framework To Service Specification Mapping .....	28
Table 11. Specification Framework To Service Specification Mapping .....	29
Table 12. Specification Framework To Service Specification Mapping .....	31
Table 13. Specification Framework To Service Specification Mapping .....	<b>ERROR!</b>
<b>BOOKMARK NOT DEFINED.</b>	
Table 14. Specification Framework To Service Specification Mapping .....	35
Table 15. Specification Framework To Service Specification Mapping .....	<b>ERROR!</b>
<b>BOOKMARK NOT DEFINED.</b>	

# 1. Introduction

## 1.1 Service Overview

The Query Management (QM) Component, as defined by the “IC/DoD Content Discovery and Retrieval (CDR) Specification Framework”<sup>1</sup>, is a CDR Component that manages Saved Searches<sup>2</sup> and may initiate search requests based on Saved Searches.

This document presents details for the realization of the Content Discovery and Retrieval (CDR) Query Management Component, hereafter termed the Query Management (QM) Service in this document, as a web service using the Representational State Transfer REST style binding. It describes the external service interfaces and the internal activities that provide service behaviors so that service providers and consumers can create and use CDR-compliant Query Management Services.

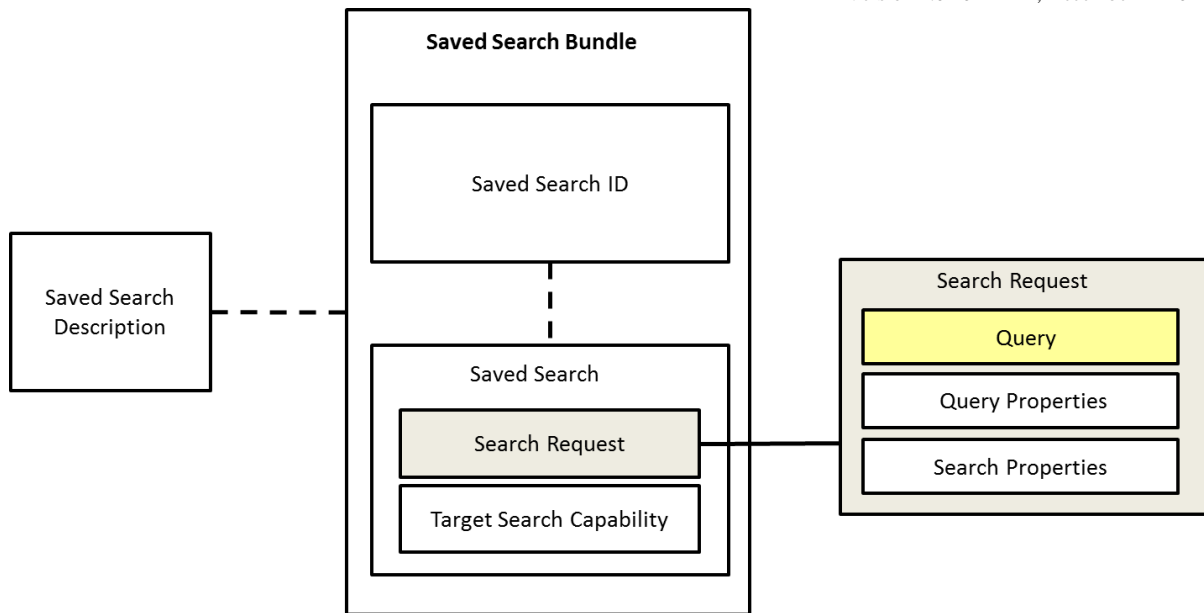
The Query Management Service provides a coordinated set of functions that enable service consumers to create, read, update, delete, search for, and execute Saved Searches. The resource model presented in Figure 1 provides an overview of the information that supports Query Management functionality. The rectangle in the middle of the figure shows the bundle of information contained within a Saved Search. The Saved Search information includes a Search Request and the target search capability (where the search is to be executed). Each Search Bundle has an associated Saved Search ID that is used to reference the Saved Search Bundle within a QM Collection.

The rectangle on the right-hand-side of Figure 1 emphasizes that the Search Request is consistent with the definition published in the CDR Search Specification. A CDR Search Request consists of the Query that contains the search criteria expressed in a documented format, along with property sets that can be used to provide more information about the query as well as the search itself. The Saved Search Description shown on the left-hand-side of Figure 1 shows the characteristic description metadata that aids in the discovery of Saved Searches.

---

<sup>1</sup> IC-DoD Content Discovery and Retrieval Specification Framework V1.0, DRAFT, 9 May 2011.

<sup>2</sup> Refer to the ‘Common Definitions of Terms related to Search and Query’ in the IC-DOD Content Discovery & Retrieval Specification Framework V1.0 for a consistent set of definitions.



**Figure 1. Query Management Resource Model**

This specification addresses the representation, management, and use of the Saved Search information bundle. Bundle information is represented within an Atom Entry Document<sup>3</sup>. The Entry serves as the container for the metadata and data associated with a Saved Search. Each 'atom entry' has an associated 'atom entry ID' which is used to reference a Saved Search Bundle and the information it contains. The Query Management functions for creating, reading, updating, and deleting Saved Search information are based on the protocol operations specified in the Atom Publishing Protocol (AtomPub).<sup>4</sup>

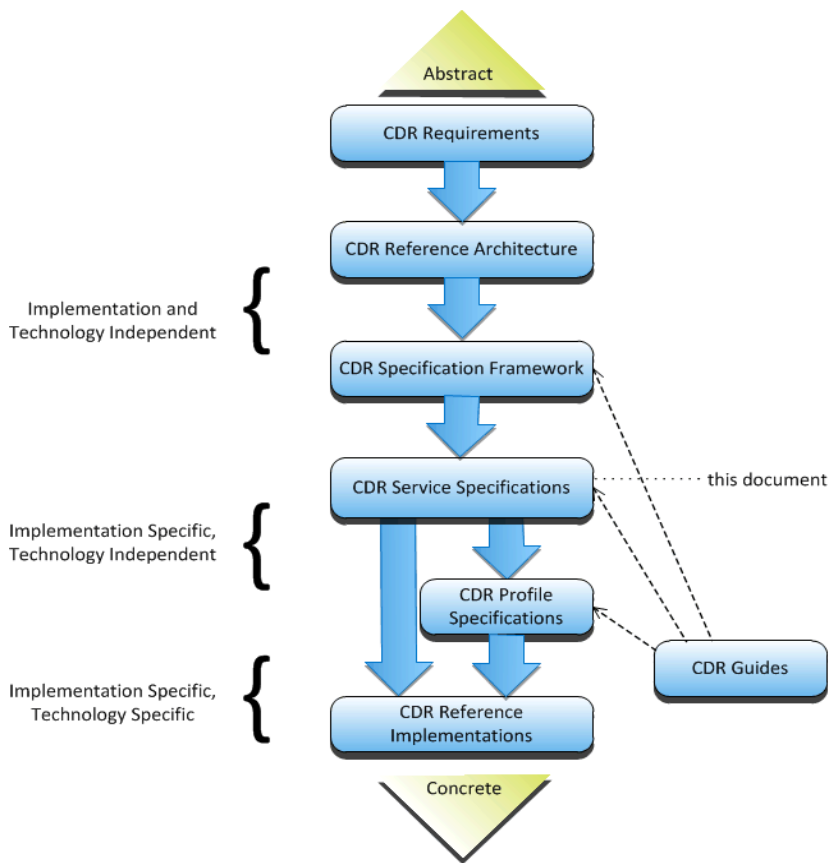
The ability to save and retrieve queries over time will require implementers to adopt a persistence mechanism which this document refers to as a QM Collection. Like an Atom Collection, the QM Collection contains a set of resources that can be retrieved in all or in part. The implementation of the QM Collection is not in the scope of this document. This document specifies the standard, implementation-independent interfaces to the functionality provided by the QM Service.

<sup>3</sup> Atom Syndication Format, <http://www.ietf.org/rfc/rfc4287.txt>.

<sup>4</sup> Atom Publishing Protocol, <http://www.ietf.org/rfc/rfc5023.txt>.

## 1.2 Relationship to Other CDR Architecture Elements

The CDR Architecture prescribes an abstract-to-concrete model for the development of architecture elements and guidance for CDR. 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 content discovery and retrieval. The following graphic in Figure 2, discussed in detail within the CDR Reference Architecture (RA),<sup>5</sup> illustrates this model.



**Figure 2. CDR Architecture Model**

<sup>5</sup> IC/DoD Content Discovery & Retrieval Reference Architecture V1.0.



As illustrated in Figure 2, the CDR Specification Framework derives from the Reference Architecture (RA) and describes behavior in terms of the capabilities, components, and usage patterns defined in the RA. The Specification Framework allows multiple Service Specifications to provide consistent interfaces, both in terms of the structure and semantics of the exchanged information.

This document provides guidance for implementing the CDR Query Management Component using the RESTful architecture style. It is intended to provide minimal requirements for implementing Query Management.

### 1.3 Scope

This specification is limited to the interactions that occur between an Initiating Consumer and the Query Management Service as described in the CDR Reference Architecture and CDR Specification Framework. QM Service Behavior is described in terms of the message exchange patterns necessary to enable service consumers to create, read, update, delete, search for, and execute Saved Searches. This specification does not address operational security use considerations as to how access control for this data is managed. These will be addressed via separate security specifications or through future versions of the specification.

### 1.4 Notational Convention

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 the IETF RFC 2119<sup>6</sup>. When these words are not capitalized, they are meant in their natural-language sense.

When describing concrete eXtensible Markup Language (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 attribute wildcard (`<xs:anyAttribute>`).

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

---

<sup>6</sup> Key words for use in RFCs to Indicate Requirement Levels, <http://www.ietf.org/rfc/rfc2119.txt>.

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

**Figure 3. Example Notation Convention**

## 1.5 Conformance

This specification defines the interface to a Query Management Service. All RESTful Query Management implementations **MUST** support all of the mandatory terms in this specification.

## 1.6 Namespaces

The following table represents only those XML namespaces that are directly leveraged in this document.

Table 1 contains namespaces used throughout this document in parameter definitions and examples.

**Table 1. Namespaces**

Prefix	URI	Description
cdrqm	<a href="urn:cdr:querymanagement:1.0">urn:cdr:querymanagement:1.0</a>	CDR Query Management at the indicated version
cdrs	<a href="urn:cdr:search:2.0">urn:cdr:search:2.0</a>	CDR Search
atom	<a href="http://www.w3.org/2005/Atom">http://www.w3.org/2005/Atom</a>	Atom Syndication

Many of the examples will include an entry such as <atom:entry xmlns...> to indicate that the full XML would include the appropriate namespace declarations but the full declarations have not been included as part of the example for brevity and ease of maintaining this specification. Any use of namespaces included in Table 1 should be interpreted as defined in Table 1. The use of elements from the atom namespace is consistent with the Atom Syndication Format.

## 2. Service Behavior

QM Service Behavior is implemented through the six fundamental functions that together define service behavior. The six functions listed below provide QM service consumers with a coordinated set of capabilities that support managing and using both searches and search related information. The first four functions facilitate the Saved Search lifecycle that is associated with persistent storage. This specification mandates the generic interfaces that a QM consumer would utilize to manage and use a Saved Search:

1. Create – The Create function is used to insert a new Saved Search into the QM Collection.
2. Read – The Read function is used to retrieve a Saved Search from the QM Collection.
3. Update – The Update function is used to change a Saved Search in the QM Collection.
4. Delete – The Delete function is used to logically remove a Saved Search from the QM Collection.
5. Execute – The Execute function enables a QM consumer to execute (run) a Saved Search at the location specified by the Target Search Capability. To process an execute request, the Query Management Service retrieves the Target Search Capability (the location of the Search Service) and the Search Request from the Saved Search. This information is then used to initiate the Search. This capability SHOULD leverage the CDR Search approach to effect the execution of a Saved Search.
6. Search – The Search function enables a prospective consumer to interrogate the QM Collection on the basis of anything to be found in the Saved Search Bundle or the Saved Search Description to determine if a suitable search has already been created. Generic search terms can be used to provide functionality that provides a ‘list’ of the contents of the QM Collection. This capability SHOULD leverage the CDR Search discovery approach.

The QM functions can be implemented using a request/response message pattern. For example, the QM-Create message request supplies information pertinent to the information to be persisted and the QM service will respond with the resource reference and status information.

### 3. Service Interfaces

This section describes the optional, required, and recommended functions that have been specified for the Query Management Service. The description of each function includes:

- A textual description of the function.
- Pre-conditions: A condition that must exist or be established before the service function can be invoked. The preconditions **MUST** be satisfied if the requested QM function is to correctly process input and generate results and post-conditions as described.
- Request Related Information:
  - The Hypertext Transfer Protocol (HTTP) method
  - The Uniform Resource Identifier (URI) template along with the variables used as URI parameters
  - HTTP Message Header
  - The HTTP Message Body (if any) which is used to carry the entity-body associated with the request
  - A Request Message Example that demonstrates how the pieces fit together
  - A mapping that shows how the QM *input* interface items map to the variables defined as input to the corresponding QM function in the CDR-IPT Specification Framework
- Response Related Information:
  - HTTP Status Code(s)<sup>7</sup> which show the results (status) of the HTTP Message Request
  - HTTP Message Body (if any)
  - A Response Message Example that illustrates possible outcomes that result from a function request
  - A mapping that shows how the QM *output* interface items map to the variables defined as output to the corresponding QM function in the CDR-IPT Specification Framework
- Post-conditions: The information state after the requested service has successfully executed.

---

<sup>7</sup> HTTP status codes are based on values from the HTTP Status Code Registry maintained by IANA – [www.iana.org/assignments/http-status-codes](http://www.iana.org/assignments/http-status-codes).

This version of the CDR Query Management Service uses elements, concepts, and constraints from the Atom Syndication Format (Atom). As mentioned in the introduction, atom provides a convenient and applicable format for a Saved Search resource representation. The atom specification provides the normative guidance for usage of atom elements within this specification. Where applicable, this QM Specification provides *additional* information and/or constraints necessary for QM implementations. This specification uses the conceptual model for CRUD-like operations as published in the Atom Publishing Protocol (AtomPub) specification. The use of Atom to express Saved Search information in a common payload enables a degree of REST/SOAP interoperability.

The following table summarizes the service functions defined for Query Management.

**Table 2. Query Management Service Function Requirements**

Function Name	Requirement Level
QM-Create	REQUIRED
QM-Read	REQUIRED
QM-Update	REQUIRED
QM-Delete	REQUIRED
QM-Execute	REQUIRED
QM-Search	RECOMMENDED

The following table provides an overview of the Query Management functions.

**Table 3. Summary of REST Query Management Functions**

QM Function	HTTP Method	URI Template
Create	POST	/savedSearches?{QMCreateProperties}
Read	GET	/savedSearches/{SavedSearchID}?{QMReadProperties}
Update	PUT	/savedSearch/{SavedSearchID}?{QMUpdateProperties}
Delete	DELETE	/savedSearches/{SavedSearchID}?{QMDeleteProperties}
Execute	GET	/savedSearches/{SavedSearchID}/SearchResults?{SearchProperties} {QMExecuteProperties}
Search	GET	/savedSearches?{SearchRequest}{QMSearchProperties}

### 3.1 QM-Create Function

The QM-Create function enables the consumer to define and save a Search Request that can later be accessed and managed through reference to a Saved Search ID.

#### 3.1.1 Preconditions

There are no specific preconditions for this function, beyond the existence of the relevant service that provides the function.

#### 3.1.2 Request

##### 3.1.2.1 HTTP Method

The Create function **MUST** use the HTTP POST method.

##### 3.1.2.2 URI Template

/savedSearches?{QMCreateProperties}

**{QMCreateProperties}** – OPTIONAL – Parameters through which the QM consumer may specify and configure optional behavior supported by the QM Create function implementation.

### 3.1.2.3 HTTP Message Header

- The Header **MUST** include the Host request-header field.
- The Header **SHOULD** include Content-Type and Content-Length.
- The Header **MUST** include a Location field to reflect that a new resource has been created.
- Unless overridden by an implementation specific Content-Type, the default Content-Type value **SHOULD** be 'application/atom+xml; type=entry'.

### 3.1.2.4 HTTP Message Body

The Body of the POST method **MUST** contain an Atom Entry Document, where the atom:content will contain the information to be captured as a Saved Search.

/atom:entry<sup>8</sup> – REQUIRED

/atom:entry/atom:id – REQUIRED

#### QM Recommendation:

- The atom id serves as the Saved Search ID and is used to uniquely identify a Saved Search. The value returned by QM-Create is permanent for the life of a Saved Search resource and is the value for all uses of the Saved Search ID.
- The user may provide a user-defined identifier be used in creating the atom:id, but the QM-Create function implementation is not obligated to honor the user's ID request. Given atom:id is a required input, a value of "urn-defaultID" **SHOULD** be used if the user does not wish to provide a user-defined identifier.

/atom:entry/atom:title – REQUIRED

/atom:entry/atom:summary – OPTIONAL

/atom:entry/atom:author – REQUIRED

#### QM Recommendation:

---

<sup>8</sup> QM Guidance is provided to identify use of Atom constructs in the QM context. The QM Guidance is meant to augment but not conflict with Atom definitions available via the referenced atom format specification, IETF RFC 4287.

- If available, the name element SHOULD be populated with the name of the organization/program/agency/person authoring the Saved Search.
- If available, the “uri” element SHOULD refer to a resource providing information about the named author.
- If available, the “email” element SHOULD refer to the email address for the named author.

Example:

```
<atom:author>
  <atom:name>DNI IC SOA Team</atom:name>
  <atom:uri>http://intelink.gov</atom:uri>
  <atom:email>some.address@dni.gov</atom:email>
</atom:author>
```

**/atom:entry/atom:updated** – REQUIRED

#### QM Recommendation:

- The initial value of atom:updated SHOULD be the date the Saved Search was created.
- The value of atom:updated SHOULD be revised whenever a QM-Update request is successfully processed.

Example:

```
<atom:updated>2003-12-13T18:30:02Z</atom:updated>
```

**/atom:entry/{SavedSearchDescription}** – OPTIONAL – is a collection of description elements that are defined by one or more industry, government, or other organizations to further describe a Saved Search.

Example of {SavedSearchDescription}:

```
<pol:AllocationPolicy xmlns:pol = "http://policy.gov">
  JointIC-DoDPolicyDefaults
</pol:AllocationPolicy>
```



**/atom:entry/atom:content/cdrqm:SavedSearch<sup>9</sup>** – REQUIRED – A search request and related information that is managed by implementations of the Query Management Component. A cdrqm:SavedSearch MUST contain either a cdrqm:SavedSearchURL OR a cdrs:SearchRequest and cdrqm:TargetSearchCapability.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL** – OPTIONAL (CONDITIONAL) – URL containing all information necessary by a search consumer to initiate a search. The URL location must reference a valid OpenSearch Search Component or Brokered Search Component implementation and the search request MUST be composed of Search Function inputs as defined by the IC/DoD Content Discovery & Retrieval OpenSearch specification.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest** – OPTIONAL (CONDITIONAL) – the information sent by a search consumer that initiates a search. The search request MUST be composed of Search Function inputs as defined by one of the IC/DoD Content Discovery & Retrieval Search specifications.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability** – OPTIONAL (CONDITIONAL) - Reference to the Search Component or Brokered Search Component implementation that is to process the search request. The reference MUST provide, or be able to be transformed to, an address through which a Consumer Component can later initiate a search by sending the search request to that address. A cdrqm:SavedSearch MUST contain either a cdrqm:SavedSearchURL OR a cdrs:SearchRequest and cdrqm:TargetSearchCapability.

---

<sup>9</sup> The normative definitions for search-related terms in the cdrqm namespace may be found in section 2.3 of the CDR Specification Framework.

### 3.1.2.5 Create Request - Message Example

```
POST /savedSearches/?timeout=60 HTTP/1.1
Host: CDR.gov
Content-Type: application/atom+xml; type=entry
Content-Length: nnn
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:...>
  <atom:id>urn:defaultID</atom:id>
  <atom:title>Mohammad Atta Keyword Search</atom:title>
  <atom:summary>Search for all occurrences of the named terrorist</atom:summary>
  <atom:author>
    <atom:name>John Smith</atom:name>
  </atom:author>
  <atom:updated>2011-07-13T18:30:02Z</atom:updated>
  <pol:AllocationPolicy xmlns:pol = "http://policy.gov">
    JointIC-DoDPolicyDefaults
  </pol:AllocationPolicy>
  <atom:content type="application/xml">
    <cdrqm:SavedSearch>
      <cdrqm:SavedSearchURL>
        http://CDR.gov/?searchTerms=Mohammad+Atta& amp;
        startIndex=1& amp;totalResults=100;format=atom
      </cdrqm:SavedSearchURL>
    </cdrqm:SavedSearch>
  </atom:content>
</atom:entry>
```

**Figure 4. Create Request**

**3.1.2.6 Mapping to CDR Specification Framework Input Variables****Table 4. Specification Framework to Service Specification Mapping**

Specification Framework Variables	REST Specification Element
Search Request	/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL /atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest
Target Search Capability	/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL /atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability
QM Properties	{ QMCreateProperties }
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:summary /atom:entry/atom:author /atom:entry/atom:updated /atom:entry/{ SavedSearchDescription }

**3.1.3 Response****3.1.3.1 HTTP Status Code(s)<sup>10</sup>**

- If the POST is successful, the service will respond with a ‘201 Created’ Status Code and a Location header that contains the IRI<sup>11</sup> of the newly created Saved Search Resource.
- Fault Conditions
  - ‘400 Bad Request’ (e.g. client sends a malformed request to the service)
  - ‘500 Internal Server Error’<sup>12</sup>

---

<sup>10</sup> Any other appropriate HTTP fault condition may be sent as described in IETF 2616.

<sup>11</sup> Internationalized Resource Identifier, IETF RFC-3987.

<sup>12</sup> 500 Internal Server Error as a catchall error response from the ordering service isn’t very descriptive. Service Implementers MAY need to develop more robust fault handling (e.g., 503 Service Unavailable and a Retry-After header indicating that the server is temporarily too busy to process the request).

### 3.1.3.2 HTTP Message Body

#### QM Recommendation

- OPTIONAL elements supplied in the QM-Create Request Message SHOULD be included in the QM-Create Response Message.

**/atom:entry** – REQUIRED

**/atom:entry/atom:id** – REQUIRED - As defined in Section 3.1.2.4.

#### QM Recommendation

- The atom:id element conveys a permanent, universally unique identifier for an entry.
- The content of an atom:id element MUST be created in a way that assures uniqueness.
- Instances of atom:id elements can be compared to determine whether an entry or feed is the same as one seen before.

Where previously defined, the following atom: and cdrqm: items primarily reflect values as provided in the HTTP request.

**/atom:entry/atom:title** – REQUIRED

**/atom:entry/atom:summary** – OPTIONAL

**/atom:entry/atom:updated** – REQUIRED

**/atom:entry/{SavedSearchDescription}** – OPTIONAL

**/atom:entry/atom:content/cdrqm:SavedSearch** – REQUIRED – As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL** – OPTIONAL (CONDITIONAL) - As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest** – OPTIONAL (CONDITIONAL) - As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability –**  
OPTIONAL (CONDITIONAL) - As defined in Section 3.1.2.4.

### 3.1.3.3 Create Response - Message Example

```

HTTP/1.1 201 Created
Content-Length: nnn
Content-Type: application/atom+xml; type=entry
Location: http://cdr.gov/savedSearches/1234
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:... >
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1</atom:id>
  <atom:title>Mohammad Atta Keyword Search</atom:title>
  <atom:summary>Search for all occurrences of the named terrorist</atom:summary>
  <atom:author>
    <atom:name>John Smith</atom:name>
  </atom:author>
  <atom:updated>2011-07-13T18:30:02Z</atom:updated>
  <pol:AllocationPolicy xmlns:pol = "http://policy.gov">
    JointIC-DoDPolicyDefaults
  </pol:AllocationPolicy>
  <atom:content type="application/xml">
    <cdrqm:SavedSearch>
      <cdrqm:SavedSearchURL>
        http://CDR.gov/?searchTerms=Mohammad+Atta&
        startIndex=1&totalResults=100;format=atom
      </cdrqm:SavedSearchURL>
    </cdrqm:SavedSearch>
  </atom:content>
</atom:entry>

```

**Figure 5. Create Response**

**3.1.3.4 Mapping to CDR Specification Framework Output Variables****Table 5. Specification Framework to Service Specification Mapping**

Specification Framework Variables	REST Specification Element
Saved Search ID	/atom:entry/atom:id
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry/{ SavedSearchDescription }

**3.1.4 Post-conditions**

1. The Saved Search Resource is available for QM-Read, QM-Update, QM-Delete, QM-Search, and QM-Execute and it is identifiable by the Saved Search ID.
2. The Create function has been audited according to applicable policy.<sup>13</sup>

**3.2 QM-Read Function**

The QM-Read function uses the Saved Search ID to get/retrieve a Saved Search Bundle from the QM Collection. Since there is a one-to-one correspondence between the Saved Search Bundle (expressed as an atom:entry) and a Saved Search (encapsulated in the atom:content element) the unique atom:id that is associated with the bundle uniquely identifies a Saved Search.

**3.2.1 Preconditions**

1. The Saved Search referenced by the Saved Search ID must exist under the management of the service.

---

<sup>13</sup> The Create function may be audited according to applicable policy regardless to the success or failure of the function.

### 3.2.2 Request

#### 3.2.2.1 HTTP Method

The Read function **MUST** use the HTTP GET method.

#### 3.2.2.2 URI Template

/savedSearches/{SavedSearchID}?{QMReadProperties}

**{SavedSearchID}** - REQUIRED - The value returned by QM-Create that represents the Saved Search resource.

**{QMReadProperties}** - OPTIONAL - Parameters through which the QM consumer may specify and configure optional behavior supported by the QM Read function implementation.

#### 3.2.2.3 HTTP Message Header

- The Header **MUST** include the Host request-header field.
- The Header **SHOULD** include Content-Type and Content-Length.
- Unless overridden by an implementation specific Content-Type, the default Content-Type value **SHOULD** be 'application/atom+xml; type=entry'.

#### 3.2.2.4 HTTP Message Body

There is no request message body for this function.

#### 3.2.2.5 Read Request - Message Example

GET /savedSearches/1234?timeOut=22 HTTP/1.1 Host: CDR.gov
--

**Figure 6. Read Request**

**3.2.2.6 Mapping to CDR Specification Framework Input Variables****Table 6. Specification Framework to Service Specification Mapping**

Input Name	REST Specification Element
Saved Search ID	{SavedSearchID}
QM Properties	{QMReadProperties}

**3.2.3 Response****3.2.3.1 HTTP Status Code(s)**

- If the GET request is successful, the service will respond with a '200 OK' Status Code and a representation of the state of the Saved Search Bundle.
- Fault Conditions
  - '404 Not Found'
  - '500 Internal Server Error'

**3.2.3.2 HTTP Message Body****/atom:entry** – REQUIRED**/atom:entry/atom:id** – REQUIRED**/atom:entry/atom:title** – REQUIRED**/atom:entry/atom:summary** – OPTIONAL**/atom:entry/atom:author** – REQUIRED**/atom:entry/atom:updated** – REQUIRED**/atom:entry/{SavedSearchDescription}** – OPTIONAL – As defined in Section 3.1.4.2.**/atom:entry/atom:content/cdrqm:SavedSearch** – REQUIRED – As defined in Section 3.1.2.4.



**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL** – OPTIONAL  
(CONDITIONAL) – As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest** – OPTIONAL  
(CONDITIONAL) – As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability** –  
OPTIONAL (CONDITIONAL) – As defined in Section 3.1.2.4.

**/atom:entry/atom:link** - OPTIONAL – As defined in Section 3.1.4.2.

### 3.2.3.3 Read Response - Message Example

```

HTTP/1.1 200 OK
Content-Length: nnn
Content-Type: application/atom+xml; type=entry
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:... >
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1</atom:id>
  <atom:title>Mohammad Atta Keyword Search</atom:title>
  <atom:summary>Search for all occurrences of the named terrorist</atom:summary>
  <atom:author>
    <atom:name>John Smith</atom:name>
  </atom:author>
  <atom:updated>2011-07-13T18:30:02Z</atom:updated>
  <pol:AllocationPolicy xmlns:pol = "http://policy.gov">
    JointIC-DoDPolicyDefaults
  </pol:AllocationPolicy>
  <atom:content type="application/xml">
    <cdrqm:SavedSearch>
      <cdrqm:SavedSearchURL>
        http://CDR.gov/?searchTerms=Mohammad+Atta&
        startIndex=1&totalResults=100;format=atom
      </cdrqm:SavedSearchURL>
    </cdrqm:SavedSearch>
  </atom:content>
</atom:entry>

```

**Figure 7. Read Response****3.2.3.4 Mapping to CDR Specification Framework Output Variables****Table 7. Specification Framework to Service Specification Mapping**

Output Name	REST Specification Element
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry/{ SavedSearchDescription }

**3.2.4 Post-conditions**

1. The Saved Search Bundle is not affected by Read.
2. The Read function has been audited according to applicable policy.<sup>14</sup>

**3.3 QM-Update Function**

The QM-Update function allows a Consumer Component to change a Saved Search. The Saved Search ID uniquely identifies the Saved Search to be modified. Partial updates are not allowed; therefore, the QM-Update request **MUST** send a complete resource representation that is used to replace the corresponding Saved Search. The Saved Search ID will remain the same. It **MAY** be necessary to retrieve the Saved Search prior to performing the update.

---

<sup>14</sup> The Read function may be audited according to applicable policy regardless to the success or failure of the function.

### 3.3.1 Preconditions

1. The Saved Search referenced by the Saved Search ID must exist under the management of the service.

### 3.3.2 Request

#### 3.3.2.1 HTTP Method

The Update function MUST use the HTTP PUT method.

#### 3.3.2.2 URI Template

/savedSearches/{SavedSearchID}?{QMUpdateProperties}

**{SavedSearchID}** – The value returned by QM-Create that represents the Saved Search resource.

**{QMUpdateProperties}** – OPTIONAL - Parameters through which the QM consumer may specify and configure optional behavior supported by the QM Update function implementation.

#### 3.3.2.3 HTTP Message Header

- The Header MUST include the Host request-header field.
- The Header SHOULD include Content-Type and Content-Length.
- Unless overridden by an implementation specific Content-Type, the default Content-Type value SHOULD be 'application/atom+xml; type=entry'.

#### 3.3.2.4 HTTP Message Body

The Update Message Body MUST contain a complete representation of the Saved Search Bundle.

**/atom:entry** – REQUIRED

**/atom:entry/atom:id** – REQUIRED

**/atom:entry/atom:title** – REQUIRED

**/atom:entry/atom:summary** – OPTIONAL

**/atom:entry/atom:author – REQUIRED**

**/atom:entry/atom:updated – REQUIRED**

**/atom:entry/atom:content/cdrqm:SavedSearch – REQUIRED – As defined in Section 3.1.2.4.**

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL – OPTIONAL (CONDITIONAL) – As defined in Section 3.1.2.4.**

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest – OPTIONAL (CONDITIONAL) – As defined in Section 3.1.2.4.**

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability – OPTIONAL (CONDITIONAL) – As defined in Section 3.1.2.4.**

### **3.3.2.5 Update Request – Message Example**

Note the atom title, summary, and the cdrs search request have been updated in this example.

```

PUT/savedSearches/1234?timeOut=90&totalResults=100 HTTP/1.1
Host: CDRServices.gov
Content-Type: application/atom+xml; type=entry
Content-Length: nnn
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:... >
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1</atom:id>
  <atom:title>Terrorist Name Search</atom:title>
  <atom:summary>Search for all occurrences of the named terrorist and the named acquaintance</atom:summary>
  <atom:author>
    <atom:name>John Smith</atom:name>
  </atom:author>
  <atom:updated>2011-07-22T10:30:02Z</atom:updated>
  <pol:AllocationPolicy xmlns:pol = "http://policy.gov">
    JointIC-DoDPolicyDefaults
  </pol:AllocationPolicy>
  <atom:content type="application/xml">
    <cdrqm:SavedSearch>
      <cdrqm:SavedSearchURL>
        http://CDR.gov/?searchTerms=Mohammad+Atta&
        startIndex=1&totalResults=100;format=atom
      </cdrqm:SavedSearchURL>
    </cdrqm:SavedSearch>
  </atom:content>
</atom:entry>

```

Figure 8. Update Request

**3.3.2.6 Mapping to CDR Specification Framework Input Variables****Table 8. Specification Framework to Service Specification Mapping**

Input Name	REST Specification Element
Saved Search ID	{SavedSearchID}
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
QM Properties	{QMUpdateProperties}

**3.3.3 Response****3.3.3.1 HTTP Status Code(s)**

- If the PUT request is successful, the service MUST return a '200 OK' followed by the resource representation in the HTTP response body or a '204 OK No Content' with no data in the HTTP response body.
- Fault Conditions
  - 404 Not Found
  - 409 Conflict, where the request has failed because of incompatible state
  - 500 Internal Server Error

**3.3.3.2 HTTP Message Body**

**/atom:entry** – REQUIRED – As defined in Section 3.1.3.2.

**/atom:entry/atom:id** – REQUIRED – As defined in Section 3.1.3.2.

**/atom:entry/atom:title** – REQUIRED – As defined in Section 3.1.3.2.

**/atom:entry/atom:summary** – OPTIONAL – As defined in Section 3.1.3.2.

**/atom:entry/atom:author** – REQUIRED – As defined in Section 3.1.3.2.

**/atom:entry/atom:updated** – REQUIRED – As defined in Section 3.1.3.2.

**/atom:entry/{SavedSearchDescription}** – OPTIONAL – As defined in Section 3.1.3.2.

**/atom:entry/atom:content/cdrqm:SavedSearch** – REQUIRED – As defined in Section 3.1.3.2.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:SavedSearchURL – OPTIONAL**  
(CONDITIONAL) – As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrs:SearchRequest – OPTIONAL**  
(CONDITIONAL) – As defined in Section 3.1.2.4.

**/atom:entry/atom:content/cdrqm:SavedSearch/cdrqm:TargetSearchCapability -**  
OPTIONAL (CONDITIONAL) – As defined in Section 3.1.2.4.

### 3.3.3.3 Update Response – Message Example

```
HTTP/1.1 200 OK
Content-Length: nnn
Content-Type: application/atom+xml; type=entry
<?xml version="1.0" encoding="UTF-8"?>
<atom:entry xmlns:... >
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1</atom:id>
  <atom:title>Terrorist Name Search</atom:title>
  <atom:summary>Search for all occurrences of the named terrorist and the named acquaintance</atom:summary>
  <atom:author>
    <atom:name>John Smith</atom:name>
  </atom:author>
  <atom:updated>2011-07-22T10:30:02Z</atom:updated>
  <pol:AllocationPolicy xmlns:pol = "http://policy.gov">
    JointIC-DoDPolicyDefaults
  </pol:AllocationPolicy>
  <atom:content type="application/xml">
    <cdrqm:SavedSearch>
      <cdrqm:SavedSearchURL>
        http://CDR.gov/?searchTerms=Mohammad+Atta&startIndex=1&totalResults=100;format=atom
      </cdrqm:SavedSearchURL>
    </cdrqm:SavedSearch>
  </atom:content>
</atom:entry>
```

**Figure 9. Update Response**

**3.3.3.4 Mapping to CDR Specification Framework Output Variables****Table 9. Specification Framework to Service Specification Mapping**

Output Name	REST Specification Element
Saved Search ID	/atom:entry/atom:id
Saved Search	/atom:entry/atom:content/cdrqm:SavedSearch
Saved Search Description	/atom:entry/atom:title /atom:entry/atom:author /atom:entry/atom:summary /atom:entry/atom:updated /atom:entry/{SavedSearchDescription}

**3.3.4 Post-conditions**

1. Saved Search Bundle reflects specified updates.
2. Saved Search Bundle is accessible by the Saved Search ID.
3. The Update function has been audited according to applicable policy.<sup>15</sup>

**3.4 QM-Delete Function**

The QM-Delete function removes a Saved Search resource from the Saved Search collection managed by the QM Component. The CDR Specification Framework includes a discussion of the design considerations related to the QM-Delete function.

**3.4.1 Preconditions**

1. The Saved Search referenced by the Saved Search ID must exist under the management of the service.

---

<sup>15</sup> The Update function may be audited according to applicable policy regardless to the success or failure of the function.



### 3.4.2 Request

#### 3.4.2.1 HTTP Method

The Delete function **MUST** use the HTTP DELETE method.

#### 3.4.2.2 URI Template

/savedSearches/{SavedSearchID}?{QMDeleteProperties}

{SavedSearchID} – The value returned by QM-Create that represents the Saved Search resource.

{QMDeleteProperties} – OPTIONAL – Parameters through which the QM consumer may specify and configure optional behavior supported by the QM Delete function implementation.

#### 3.4.2.3 HTTP Message Header

- The Header **MUST** include the Host request-header field.

#### 3.4.2.4 HTTP Message Body

There is no request message body for this function.

#### 3.4.2.5 Delete Request - Message Example

DELETE /savedSearches/1234?delete=logical HTTP/1.1 Host: CDRServices.gov
---

**Figure 10. Delete Request**

#### 3.4.2.6 Mapping to CDR Specification Framework Input Variables

**Table 10. Specification Framework to Service Specification Mapping**

Input Name	REST Specification Element
Saved Search ID	{SavedSearchID}
QM Properties	{QMDeleteProperties}

### 3.4.3 Response

#### 3.4.3.1 HTTP Status Code(s)

- If the Delete request is successful, the service MUST return a '204 OK No Content' with no data in the HTTP response body.
- Fault conditions:
  - '404 Not Found'
  - '405' Method Not Allowed'
  - '503 Service Unavailable'

#### 3.4.3.2 HTTP Message Body

There is no response message body for this function.

#### 3.4.3.3 Delete Response - Message Example

HTTP/1.1 204 OK No Content Date: Fri, 29 Jul 2011 08:30:03
---

**Figure 11. Delete Response**

#### 3.4.3.4 Mapping to CDR Specification Framework Output Variables

**Table 11. Specification Framework to Service Specification Mapping**

Input Name	REST Specification Element
Confirmation	HTTP Status

### 3.4.4 Post-conditions

1. Saved Search Bundle is no longer accessible by QM functions.

2. The Delete function has been audited according to applicable policy.<sup>16</sup>

### 3.5 QM-Execute Function

The QM-Execute function leverages a CDR Search to submit a Search Request to a specified location, where both the Search Request and the location are contained within a Saved Search that is managed in a QM Collection. In particular, QM-Execute uses a modified CDR Search Interface as specified in the “IC-DoD Content Discovery & Retrieval Search Service Specification for OpenSearch Implementations 1.1”, with the Query being replaced by a Saved Search ID. When the request message is received by the Query Management Service, it uses the Saved Search ID to retrieve the corresponding Saved Search from the QM Collection and sends the Search Request to the location specified in the query or in the Target Search Capability. In accordance with the CDR Search Specification, the Search Service builds a set of items, called Search Results, which match the Query. The Search Results support HTML and Atom response formats. Search Properties specified in the QM-Execute request supersede the property values contained in the Saved Search.

#### 3.5.1 Preconditions

1. The Saved Search referenced by the Saved Search ID must exist under the management of the service.

#### 3.5.2 Request

##### 3.5.2.1 HTTP Method

The Execute function MUST use the HTTP GET method.

##### 3.5.2.2 URI Template

```
/SavedSearches/{SavedSearchID}/SearchResults?{SearchProperties}{QMExecuteProperties}
```

---

<sup>16</sup> The Delete function may be audited according to applicable policy regardless to the success or failure of the function.

**{SavedSearchID}** – The value returned by QM-Create that represents the Saved Search resource.

**{SearchProperties}** – OPTIONAL – Parameters through which the QM consumer may specify and configure optional behavior supported by the referenced Search Component or Brokered Search Component.

**{QMExecuteProperties}** – OPTIONAL – Parameters through which the QM consumer may specify and configure optional behavior supported by the QM Execute function implementation.

### 3.5.2.3 HTTP Message Header

- The Header MUST include the Host request-header field.

### 3.5.2.4 HTTP Message Body

There is no request message body for this function.

### 3.5.2.5 Execute Request - Message Example

```
GET/savedSearches/1234/ResultSet?startIndex=20&timeOut=25 HTTP/1.1
Host: CDR.gov
```

**Figure 12. Execute Request**

Note that in Figure 12 the startIndex of the search request has been specified. This value will override the startIndex specified in the Saved Search.

### 3.5.2.6 Mapping to CDR Specification Framework Input Variables

**Table 12. Specification Framework to Service Specification Mapping**

Input Name	REST Specification Element
Modified Search Function Inputs	{SearchProperties}
Saved Search ID	{SavedSearchID}
QM Properties	{QMSearchProperties}

### 3.5.3 Response

*The Output and Post-conditions are those as specified in the “IC-DoD Content Discovery & Retrieval Search Service Specification for REST Implementations 2.0” Search function.*

### 3.5.3.1 Execute Response - Message Example

```

<atom:entry>          <atom:entry>

HTTP/1.1 200 OK
Content-Length: nnn
Content-Type: application/atom+xml; type=feed

<?xml version="1.0" encoding="UTF-8"?>
<atom:feed xmlns="http://www.w3.org/2005/Atom"
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <atom:title>CDR.gov Search: Terrorist Name Search</atom:title>
  <atom:link href="http://CDR.gov/savedSearches/1234"/>
  <atom:updated>2011-07-22T11:15:00Z</atom:updated>
  <atom:author>
    <name>Bill Smith</name>
  </atom:author>
  <!-- NOTE: this is the *feed* id -->
  <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1</atom:id>
  <opensearch:totalResults>4000</opensearch:totalResults>
  <opensearch:startIndex>20</opensearch:startIndex>
  <opensearch:itemsPerPage>10</opensearch:itemsPerPage>
  <opensearch:Query role="request" searchTerms="Mohammad+Atta+AND+Perry+Noid" startPage="1" />
  <atom:link rel="alternate" href="http://badguy.gov/Atta+Noid?pw=3" type="text/html"/>
  <atom:link rel="self" href="http://reallybadguy.gov/Mohammad+Atta+Noid?pw=3&format=atom"
    type="application/atom+xml"/>
  <atom:link rel="last" href="http://example.com/New+York+History?pw=42299&format=atom"
    type="application/atom+xml"/>
  <atom:link rel="search" type="application/opensearchdescription+xml"
    href="http://example.com/opensearchdescription.xml"/>
  <atom:entry>
    <atom:title>Dirty Rotten Scoundrels</atom:title>
    <atom:link href="http://agency.watchlist.gov/badGuy.html"/>
    <atom:id>urn:uuid:1225c695-cfb8-4ebb-aaaa-80da344efa6a</atom:id>
    <atom:updated>2011-07-01T18:30:02Z</atom:updated>
    <atom:content type="text">
      ... This list contains the list of individuals ....
      known associates ... activities ... hostile to the United States
      ... Be on the lookout...
    </atom:content>
  </atom:entry>
</atom:feed>

```

Figure 13. Execute Response

## 3.6 QM-Search Function

The QM-Search function provides the capability of listing or searching the QM Collection, which is the repository of Saved Searches. QM-Search MUST be compliant with CDR Search Interface as Specified in the “IC-DoD Content Discovery & Retrieval Search Service Specification for OpenSearch Implementations 1.1”. As described in the CDR Search specification, an OpenSearch Query (including corresponding search attributes) is sent via HTTP to a content collection’s OpenSearch compliant service. The Search Service builds a set of items, called Search Results, which match the Query. The Search Results support HTML and Atom response formats. It is the Atom Feed content that enables QM-Search to list Saved Searches.

### 3.6.1 Preconditions

No additional preconditions exist beyond those defined for the CDR Search Service.

### 3.6.2 Request

#### 3.6.2.1 HTTP Method

The Search function MUST use the HTTP GET method.

#### 3.6.2.2 URI Template

/savedSearches?{SearchRequest}{QMSearchProperties}

**{SearchRequest}** – REQUIRED – Search Request query parameters as specified in the IC-DoD Content Discovery & Retrieval OpenSearch Specification.

**{QMSearchProperties}** – OPTIONAL – Parameters through which the QM consumer may specify and configure optional behavior supported by the QM Search function implementation.

The URI MUST reference the QM Collection Resource.

#### 3.6.2.3 HTTP Message Header

- The Header MUST include the Host request-header field.

### 3.6.2.4 HTTP Message Body

There is no request message body for this function.

### 3.6.2.5 Search Request - Message Example

```
GET/savedSearches?searchTerms=Mohammad+Atta&itemsPerPage=15&timeOut=120 HTTP/1.1  
Host: CDR/QM_Collection.gov
```

**Figure 14. Search Request**

### 3.6.2.6 Mapping to CDR Specification Framework Input Variables

**Table 13. Specification Framework to Service Specification Mapping**

Input Name	REST Specification Element
Search Request	{SearchRequest}

### 3.6.3 Response

*The Output and Post-conditions are those as specified in the “IC-DoD Content Discovery & Retrieval Search Service Specification for REST Implementations 2.0” Search function.*



**3.6.3.1 Search Response - Message Example**

```

HTTP/1.1 200 OK
Content-Length: nnn
Content-Type: application/atom+xml; type=feed
<?xml version="1.0" encoding="UTF-8"?>
<atom:feed xmlns:atom="http://www.w3.org/2005/Atom">
  xmlns:opensearch="http://a9.com/-/spec/opensearch/1.1/">
  <atom:title>CDR/QM_Search.gov Search: Terrorist Name Search</atom:title>
  <atom:link href="http://CDR.gov/Mohammad+Atta"/>
  <atom:updated>2011-07-22T11:15:00Z</atom:updated>
  <atom:author>
    <name>Search Team</name>
  </atom:author>
  <atom:entry>
    <atom:id>urn:uuid:60a76c80-d399-11d9-b93C-0003939e0af6::1</atom:id>
    <atom:title>Terrorist Name Search</atom:title>
    <atom:summary>Search for all occurrences of the named terrorist and the named acquaintance</atom:summary>
    <atom:author>
      <atom:name>John Smith</atom:name>
    </atom:author>
    <atom:updated>2011-07-22T10:30:02Z</atom:updated>
    <pol:AllocationPolicy xmlns:pol="http://policy.gov">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
    <atom:content type="application/xml">
      <cdrqm:SavedSearch>
        <cdrqm:SavedSearchURL>
          http://CDR.gov/?searchTerms=Mohammad+Atta&
            startIndex=1&totalResults=100;format=atom
        </cdrqm:SavedSearchURL>
      </cdrqm:SavedSearch>
    </atom:content>
  </atom:entry>
  <atom:entry>
    <atom:id>urn:uuid:ffff6c80-d399-11d9-b93C-0003939ddfa8::1</atom:id>
    <atom:title>Directed Name Search</atom:title>
    <atom:summary>This is a keyword search that can be used to search yy sources</atom:summary>
    <atom:author>
      <atom:name>Jane Doe</atom:name>
    </atom:author>
    <atom:updated>2011-015-22T10:30:02Z</atom:updated>
    <pol:AllocationPolicy xmlns:pol="http://policy.gov">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
    <atom:content type="application/xml">
      <cdrqm:SavedSearch>
        <cdrqm:SavedSearchURL>
          http://CDR.gov/?searchTerms=Mohammad+Atta&
            startIndex=1&totalResults=100;format=atom
        </cdrqm:SavedSearchURL>
      </cdrqm:SavedSearch>
    </atom:content>
  </atom:entry>
</atom:feed>

```

**Figure 15. QM Search Response**

## 4. References

This section includes additional references that may be used to provide further insight into the overall design concepts that serve to guide the CDR-IPT engineering efforts.

### 4.1 Content Discovery and Retrieval Specifications

The CDR Reference Architecture and Specification Framework provide essential background and context to service designers. This document was based on the following CDR Reference Architecture and Specification Framework document versions:

- “IC/DoD Content Discovery and Retrieval Reference Architecture Version 1.1”, 25 February 2011.
- “IC/DoD Content Discovery and Retrieval Specification Framework Version 1.0”, 9 May 2011.
- IC/DoD REST Interface Encoding Specification for CDR Search, Version 2.0, DRAFT”, 14 December, 2011

The most recent version of the documents can be found at the unclassified Intelink web site in the Information Access and Discovery focus area section.