



Intelligence Community and Department of Defense Technical Specification

REST Service Encoding Specification for Content Discovery and Retrieval: Query Management

Version 2

09 May 2014

Distribution Notice:

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

Table of Contents

Chapter 1 - Introduction	1
1.1 - Service Overview	1
1.2 - Scope	2
1.3 - Artifact Overview	2
1.4 - Enterprise Need	4
1.5 - Conventions	4
1.5.1 - Namespaces	5
1.6 - Conformance	5
1.7 - Saved Search Specified as the CDR Resource	5
1.8 - Security	6
Chapter 2 - Service Behavior	8
2.1 - QM Use of M-Create	8
2.2 - QM Use of M-Read	8
2.3 - QM Use of M-Update	8
2.4 - QM Use of M-Delete	8
2.5 - QM Use of M-Search	8
2.6 - QM-Execute	8
Chapter 3 - Service Interfaces	10
3.1 - QM Profile of M-Create Function	10
3.1.1 - Preconditions	10
3.1.2 - Input	10
3.1.3 - Output	11
3.1.4 - Post-conditions	12
3.1.5 - Fault Conditions	12
3.2 - QM Profile of M-Read Function	12
3.2.1 - Preconditions	12
3.2.2 - Input	13
3.2.3 - Output	13
3.2.4 - Post-conditions	13
3.2.5 - Fault Conditions	13
3.3 - QM Profile of M-Update Function	13
3.3.1 - Preconditions	14
3.3.2 - Input	14
3.3.3 - Output	15
3.3.4 - Post-conditions	16
3.3.5 - Fault conditions	16
3.4 - QM Profile of M-Delete Function	16
3.4.1 - Preconditions	16
3.4.2 - Input	17
3.4.3 - Output	17
3.4.4 - Post-conditions	17
3.4.5 - Fault Conditions	17
3.5 - QM Profile of M-Search Function	17
3.6 - QM-Execute Function	18
3.6.1 - Preconditions	18
3.6.2 - Input	18

3.6.2.1 - HTTP Method	18
3.6.2.2 - URI Template	18
3.6.2.3 - HTTP Message Header	19
3.6.2.4 - HTTP Message Body	19
3.6.2.5 - QM-Execute Request Example	19
3.6.3 - Output	20
3.6.4 - Post-conditions	20
3.6.5 - Fault Conditions	20
Appendix A - Feature Summary	21
A.1 - QM Feature Comparison	21
Appendix B - Change History	22
B.1 - V2 Change Summary	22
Appendix C - Saved Search Resource Types (Normative)	23
C.1 - SavedSearch QMv1 (urn:cdr:resourceType:qmv1)	23
C.2 - Saved Search OS (urn:cdr:resourceType:ssos:1.0)	23
C.3 - Saved Search XML (urn:cdr:resourceType:ssxml:1.0)	26
C.4 - Saved Search OS Broker (urn:cdr:resourceType:ssosb:1.0)	29
C.5 - Saved Search XML Broker (urn:cdr:resourceType:ssxmlb:1.0)	32
Appendix D - Mapping to Specification Framework	36
Appendix E - Glossary	37
Appendix F - Bibliography	39
Appendix G - Points of Contact	42
Appendix H - IC CIO Approval Memo	43

List of Figures

Figure 1 - Query Management Resource Model	2
Figure 2 - CDR Architecture Model	3
Figure 3 - Example Notation Convention	4
Figure 4 - QM Specialization of M-Create Request Example	11
Figure 5 - QM Specialization of M-Create Response Example	12
Figure 6 - QM Specialization of M-Update Request Example	15
Figure 7 - QM Specialization of M-Update Response Example	16
Figure 8 - QM-Execute Request Example	19
Figure 9 - Example of QM Use of Saved Search OS Resource Type in M-Create Request	24
Figure 10 - Example of QM Use of Saved Search OS Resource Type in M-Create Response	25
Figure 11 - Example of QM Use of Saved Search XML Resource Type in M-Create Request	27
Figure 12 - Example of QM Use of Saved Search XML Resource Type in M-Create Response	28
Figure 13 - Example of QM Use of Saved Search OS Broker Resource Type in M-Create Request	30
Figure 14 - Example of QM Use of Saved Search OS Broker Resource Type in M-Create Response	31
Figure 15 - Example of QM Use of Saved Search XML Broker Resource Type in M-Create Request	33
Figure 16 - Example of QM Use of Saved Search OS Broker Resource Type in M-Create Response	34

List of Tables

Table 1 - Relationship between Manage Component and QM Component	1
Table 2 - Namespaces	5
Table 3 - Saved Search Resource Type URIs	6
Table 4 - Saved Search Description Vocabulary URIs	6
Table 5 - List of QM-Execute Function Faults	20
Table 6 - Feature Summary Legend	21
Table 7 - QM Feature Comparison	21
Table 8 - DES Version Identifier History	22
Table 9 - Data Encoding Specification V2 Change Summary	22
Table 10 - Elements and Attributes of the Saved Search OS Resource Type	23
Table 11 - Elements and Attributes of the Saved Search XML Resource Type	26
Table 12 - Elements and Attributes of the Saved Search OS Broker Resource Type	29
Table 13 - Elements and Attributes of the Saved Search XML Broker Resource Type	32
Table 14 - Mapping to CDR Specification Framework Input Variables	36
Table 15 - Mapping to CDR Specification Framework Output Variables	36

Chapter 1 - Introduction

1.1 - Service Overview

The Query Management (QM) Component, as defined by the Intelligence Community (IC) and Department of Defense (DoD) Content Discovery and Retrieval (CDR) Specification Framework (CDR-SF), [7] serves as the primary mechanism to manage Saved Searches and to initiate search requests based on Saved Searches. ¹

This specification defines requirements and provides guidance for the realization of the CDR Query Management Component, hereafter termed the Query Management (QM) Service in this document, as a RESTful ² web service. It does this by defining a profile of the CDR Manage functions, in particular, specifying how the Manage functions are used when a Saved Search serves as the CDR Resource. The content of this specification describes the QM Service's behavior, interface and other aspects in detail, providing enough information for QM Service providers and consumers to create and use CDR- conformant QM Services.

In addition to the use of the CDR Manage functions, the QM Service provides a single recommended function that enables service consumers to execute Saved Searches. The resource model presented in [Figure 1](#) provides an overview of the information that supports Query Management functionality. It is consistent with the model discussed in the CDR Manage specifications, CDR-RM, [3] and CDR-SM, [8] but it is expressed in terms of a Saved Search as the target resource for Query Management. In particular, the relationship of the QM Component to the Manage Component is as shown in [Table 1](#) .

Table 1 - Relationship between Manage Component and QM Component

Manage Component	QM Component
CDR Resource	Saved Search
CDR Resource Identifier	Saved Search ID
CDR Resource Collection	QM Collection

Use of the Manage Component should be interpreted in terms of these substitutions. In addition, the Resource Type input for M-Create MUST indicate the structure and semantics defined for a Saved Search. The QM use of Manage functions is shown in [Chapter 3 - Service Interfaces](#) .

The Search Request 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](#) comprises the characteristic description metadata that aids in the discovery of Saved Searches. Some of this description e.g., the date the Saved Search was created in QM Collection, will be generated as

¹Refer to the Common Definitions of Terms related to Search and Query in the IC/ DoD CDR-SF [7] for a consistent set of definitions.

² REST is an architectural style that encapsulates the design principles of the World Wide Web (WWW).

part of the Saved Search creation. Other description, such as a link to applicable policies, will be supplied by someone with responsibility for the resource. The description vocabulary associated with Saved Searches is anticipated to be an extension of a basic description vocabulary appropriate for any CDR Resource. For example, the basic vocabulary may include the last modification date, while the QM description vocabulary may include the query language for a Saved Search instance.

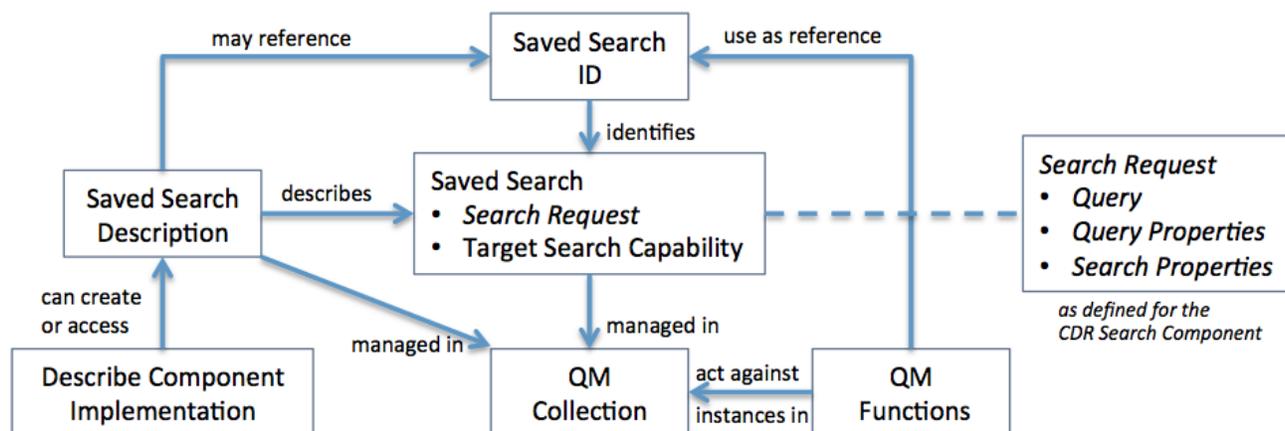


Figure 1 : Query Management Resource Model

The ability to save and retrieve saved searches over time will require implementers to adopt a persistence mechanism which this document refers to as a QM Collection. The implementation of the QM Collection is not in the scope of this document but this document MUST NOT be interpreted as precluding a single collection implementation managing other CDR Resources in addition to Saved Searches. This document specifies the standard interfaces to the functionality provided by the QM Service.

1.2 - Scope

This specification is limited to the interactions that occur between an Initiating Consumer and the QM Service as described in the CDR Reference Architecture (CDR-RA) [1] and CDR-SF. [7]

This specification provides the description of the QM Service Behavior in terms of the message exchange patterns necessary to enable service consumers to create, read, update, delete, search for, and execute Saved Searches.

The scope of this specification is limited as follows:

- Versioning of the managed Saved Searches is not defined
- A Saved Search update is a full replacement of the target resource. Partial update of the Saved Search is out of scope.

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 CDR Integrated Project

Team (IPT). The CDR-RA [1] prescribes an abstract-to-concrete model for the development of architecture elements and guidance for content discovery and retrieval. Each layer or tier of the model is intended to provide key aspects of the overall guidance to achieve the goals and objectives for joint DoD/ IC content discovery and retrieval. Figure 2, discussed in detail within the CDR-RA, [1] illustrates this model.

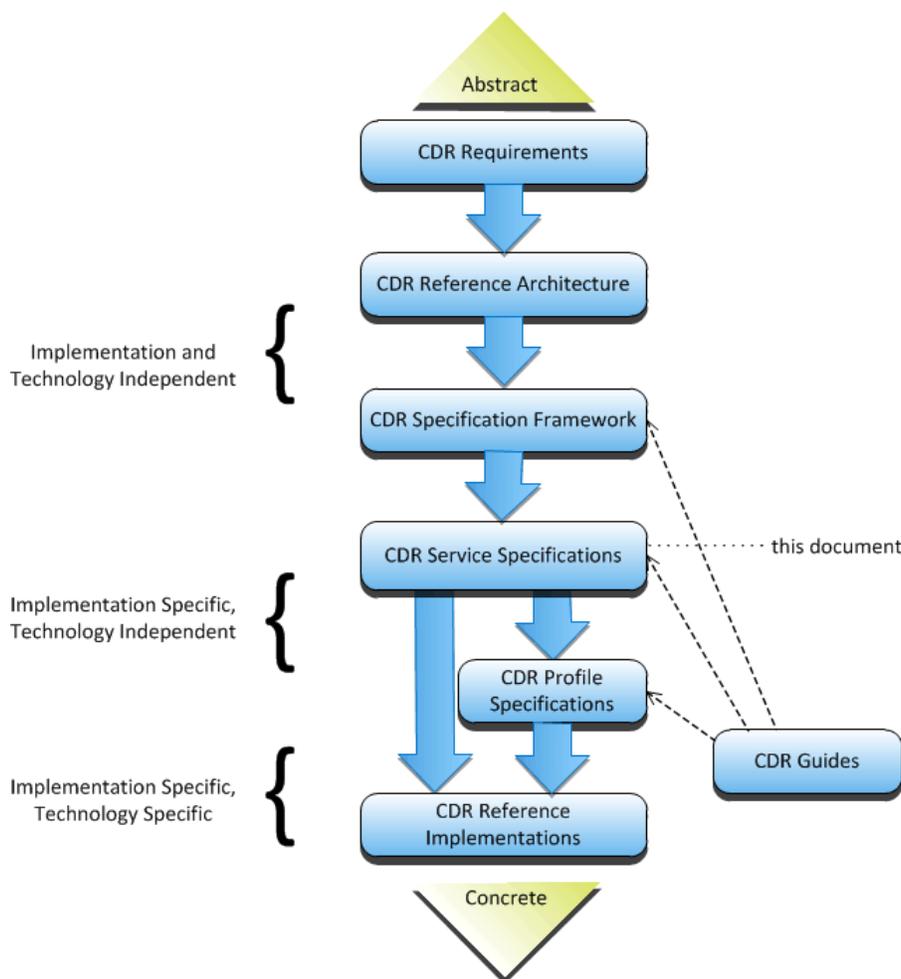


Figure 2 : CDR Architecture Model

As illustrated in Figure 2, the CDR-SF [7] derives from the CDR-RA and describes behavior in terms of the capabilities, components, and usage patterns defined in the CDR-RA. Multiple CDR Service Specifications are derived from the CDR-SF, with separate specifications associated with the components of the architecture (e.g., QM) and, for each service, separate specifications to address Representational State Transfer (REST) and SOAP implementations.

This document is a specification for implementing the CDR QM Service as a RESTful Web Service. It is intended to parallel the corresponding SOAP specification, the IC/ DoD SOAP Interface Encoding Specification for CDR Query Management,[10] 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 - Enterprise Need

Enterprise needs and requirements for this specification can be found in the following policies and implementation guidance:

- IC Information Technology Enterprise (IC ITE)
 - Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan^[12]
- 500 Series:
 - Intelligence Community Directive (ICD) 500, Director Of National Intelligence Chief Information Officer^[13]
 - Joint IC/DoD Memorandum, IC and DoD Commitment to an Interoperable Service-Based Environment (13 Jul 07)^[19]

1.5 - Conventions

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. ^[16] 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 attribute wildcard (<xs:anyAttribute>).

A parameter contained in curly braces, 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 blue border as shown in [Figure 3](#). These are meant to be illustrative and represent one way that the described syntax can be used.

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

Figure 3 : Example Notation Convention

Examples are typically provided or referenced for each function.

1.5.1 - Namespaces

Namespaces referenced in this document and the prefixes used to represent them are listed in [Table 2](#) . The namespace prefix of any XML Qualified Name (QName) used in any example in this document should be interpreted using the information in [Table 2](#) .

Table 2 - Namespaces

Prefix	URI	Description
cdrm	urn:cdr:manage:1	CDR Manage at the indicated version
cdrqm	urn:cdr:querymanagement:2	CDR Query Management at the indicated version
cdrs	urn:cdr:search:3.0	CDR Search at the indicated version
xs	http://www.w3.org/2001/XMLSchema	XML Schema
atom	http://www.w3.org/2005/Atom	Atom Syndication Format ^[18]

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 should be interpreted as defined in [Table 2](#) . The use of elements from the atom namespace is consistent with the Atom Syndication Format.

1.6 - Conformance

For an implementation to conform to this specification, it MUST adhere to all normative aspects of the specification. For the purposes of this document, normative and informative are defined as:

- *Normative*: considered to be prescriptive and necessary to conform to the standard.
- *Informative*: serving to instruct, enlighten or inform.

This specification defines an interface to a QM 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 Query Management specification, the implementation MUST adhere to all mandatory aspects of the specification.

For the indicated functions, this specification defines a QM profile for use of CDR Manage. The QM profile constrains the use of Manage functions but is not intended to conflict with conformance to the corresponding Manage specifications.

1.7 - Saved Search Specified as the CDR Resource

The CDR Manage Service^[3] defines the means to specify the specific resource type to be managed; for Query Management the CDR Resource Type MUST be associated with a value that defines the structure and semantics of a Saved Search. In addition, the CDR Resource Description MUST specify a description structure and semantics that is applicable for describing a Saved Search.

Table 3 - Saved Search Resource Type URIs^a

Name	URI	Description
Saved Search QMv1	urn:cdr:resourceType:qmv1	Definition for Saved Search using the Atom format as defined in the QM v1.0 specifications (See Appendix B)
Saved Search OS	urn:cdr:resourceType:ssos:1.0	Definition for Saved Search using OpenSearch format (See Appendix B)
Saved Search XML	urn:cdr:resourceType:ssxml:1.0	Definition for Saved Search using XML to specify information as payload (See Appendix B)
Saved Search OS Broker	urn:cdr:resourceType:ssosb:1.0	Definition for Saved Search using OpenSearch format for brokered search (See Appendix B)
Saved Search XML Broker	urn:cdr:resourceType:ssxmlb:1.0	Definition for Saved Search using XML to specify information as payload for brokered search (See Appendix B)
Saved Search JSON	urn:cdr:resourceType:ssjson:1.0	Definition for Saved Search using JSON format (notional)

^aThe Uniform Resource Identifier (URI) may be either Uniform Resource Locator (URL) or Uniform Resource Name (URN).

Table 4 - Saved Search Description Vocabulary URIs

Name	URI	Description
CDR Resource	urn:cdr:resourceVocab:res:1.0	Definition for basic CDR Resource vocabulary
Saved Search	urn:cdr:resourceVocab:ss:1.0	Definition for Saved Search vocabulary

[Table 3](#) and [Table 4](#) reference Saved Search resource types and description vocabularies relevant in describing Saved Search instances³ Additional acceptable values for Saved Search resources and description vocabularies MAY be defined in the future but such values MUST also be identified by Name and by a URI that is associated with detailed definitions of the new resource type or new description vocabulary. Additional acceptable values are anticipated to include new versions of those currently identified as part of or recognized by the CDR specification set.

1.8 - Security

This specification does not directly address security concerns. It will be necessary for any implementation of this specification to address security concerns relevant to the systems with which they interact and the corresponding governance bodies. Several aspects of Query

³The Saved Search JSON resource type is notional in that the corresponding Manage profiles have been discussed but have not been explicitly defined.

Management, to include appropriate access to Saved Search instances, should be addressed in the detailed security plan of an implementation, but are out of scope for this document.

Chapter 2 - Service Behavior

As defined in the CDR-SF, Query Management behavior is realized through six activities. Five activities are realized through the use of the CDR Manage Service and are accessed through the Manage create, read, update, delete, and search interfaces. Refer to the REST Manage specification^[3] for specifics of using these functions; this document contains details that constitute a profile of how the Manage functions are used in the context of QM and Saved Searches.

The sixth activity, executing a Saved Search, is unique to Query Management and accessed through the QM-Execute Function interface.

2.1 - QM Use of M-Create

The Manage M-Create function is used to insert a new Saved Search into the QM Collection. A Saved Search ID is returned if the M-Create is successful.

2.2 - QM Use of M-Read

The Manage M-Read function is used to retrieve a Saved Search from a QM Collection. It refers to the Saved Search through its Saved Search ID. M-Read may also be used to retrieve the Saved Search Description.

2.3 - QM Use of M-Update

The Manage M-Update function is used to change a Saved Search being managed through the QM Collection. It refers to the Saved Search through its Saved Search ID. M-Update replaces the existing Saved Search and the Saved Search Description with the input provided and does not support partial updates.

2.4 - QM Use of M-Delete

The Manage M-Delete function is used to remove a Saved Search from the QM Collection. It refers to the Saved Search through its Saved Search ID. The M-Delete section of CDR-SF and the CDR Manage specifications discuss considerations when deleting a CDR Resource, such as a Saved Search.

2.5 - QM Use of M-Search

The Manage M-Search function enables a prospective consumer to interrogate the QM Collection to determine if a suitable Saved Search has already been created and is being managed. As discussed in CDR-RM, this capability SHOULD leverage CDR Search.

2.6 - QM-Execute

The QM-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 QM Service retrieves the value of 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 CDR Search to execute the Saved Search.

Chapter 3 - Service Interfaces

The service interface contains the technical descriptions ¹ of the functions through which the consumer will interact with the service. Support for input and output parameters for each function is described in associated input and output tables in terms of what is expected of the QM Service and what is expected in terms of a consumer interacting with the service. The discussions herein of the CDR Manage functions provide a profile of the use of those functions in the context of a Saved Search acting as the CDR Resource for QM.

3.1 - QM Profile of M-Create Function

A QM Service MUST implement the M-Create Function as defined in this section. The relationship between Manage and QM is shown in [Table 1](#).

3.1.1 - Preconditions

The following preconditions MUST be satisfied if the QM use of the M-Create 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 QM use of the M-Create Function implementation.
2. A QM collection exists and is available.

3.1.2 - Input

The input is as specified for the Manage M-Create function with the following additional constraints:

1. The URI Template MUST conform with the following ²: `http://{anyAuthority}/{anyHierarchy}/CDRresource?resourceType={savedSearchType}&{MCreateProperties}`

where

`{savedSearchType}` – REQUIRED – MUST be a URI for which the content indicated by the URI defines the structure and semantics of a Saved Search; the default value SHOULD be `urn:cdr:resourceType:ssos`. `{savedSearchType}` replaces the more general `{CDRresourceType}`. See Appendix A for the definitions of select Saved Search resource types.

`{MCreateProperties}` MAY include additional parameters that support configuring optional behavior specific to use for Query Management.

2. `{CDRresource}` MUST be replaced by a saved search resource `{saved search resource}` as specified by `{savedSearchType}`.

¹The QM Service is intended to conform as described by the Query Management Component section of the CDR-SF. [\[7\]](#)

²For example, `http://www.cdr.org/templates/examples/CDRresource?resourceType=urn:cdr:resourceType:ssxml&output=all` is a conforming URL.

3. {descriptionVocabulary} MUST be replaced by a URI that indicates a vocabulary that adequately describes a saved search resource.
4. {description} MUST be replaced by description information {saved search description} that corresponds as specified to {descriptionVocabulary} or the specified default.

[Figure 4](#) shows an example of the QM specialization of an M-Create request. In this example, several { } fields appear for which resource-specific substitutions are needed. In addition, {XML payload} is the same as <cdrm:CDRresource> and its child elements as shown earlier in the example, but is used for brevity.

```

http://CDR.org /CDRresource?resourceType={savedSearchType}&output=all
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
  {saved search resource}
  <cdrm:description descriptionVocabulary="{descriptionVocabulary}">
    {saved search description}
  </cdrm:description>
</cdrm:CDRresource>

results in

POST /CDRresource?resourceType={savedSearchType} &output; =all HTTP/1.1
Host: CDR.org
Content-Type: application/xml
Content-Length: nnn
{XML payload}

```

Figure 4 : QM Specialization of M-Create Request Example

3.1.3 - Output

The response MUST conform to the Manage M-Create function with the appropriate inclusion of the following:

1. The {CDRresourceID} of the CDR Resource Identifier MUST be replaced with the {savedSearchID}. The {savedSearchID}. The {savedSearchID} MUST conform to the definition of the {CDRresourceID}; in addition, the QM Collection MAY impose additional constraints in the generation of the {savedSearchID}.
2. {CDRresource} MUST be replaced by a saved search resource {saved search resource} as specified by {savedSearchType}.
3. {description} MUST be replaced with the details reflecting both the information provided by the M-Create consumer, per [Section 3.1.2 - Input](#), and information that is automatically maintained by the QM Collection.

[Figure 5](#) shows an example of the M-Create response that corresponds to the M-Create request example shown in [Figure 4](#). It includes several { } fields for which resource-specific substitutions are needed.

```

HTTP/1.1 201 Created
Content-Length: nnn
Content-Type: application+xml
Location: http://CDR.org/CDRresource/1234
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:MResponse xmlns:cdrm="...">
  <cdrm:id> http://CDR.org/CDRresource/1234 </cdrm:id>
  {saved search resource}
  <cdrm:description>
    <{saved search description}>
  </cdrm:description>
</cdrm:MResponse>

```

Figure 5 : QM Specialization of M-Create Response Example

3.1.4 - Post-conditions

The following conditions MUST be met upon completion of M-Create:

1. The Saved Search Resource is available for M-Read, M-Update, M-Delete, M-Search, and QM-Execute and it is identifiable by the Saved Search ID.
2. The use of this function has been audited according to applicable policy.³

3.1.5 - Fault Conditions

Faults as defined in the CDR Manage specification for the M-Create function are also applicable to the QM use of M-Create. The corresponding QM term as identified in [Table 1](#) MAY be substituted for CDR Manage term used in the fault text.

3.2 - QM Profile of M-Read Function

A QM Service MUST implement the M-Read Function as defined in this section. The relationship between Manage and QM is shown in [Table 1](#).

3.2.1 - Preconditions

The following preconditions MUST be satisfied if the QM use of the M-Read 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 QM use of the M-Read Function implementation.
2. The Saved Search instance can be retrieved through reference to its Saved Search ID for purposes of M-Read.

³The Create function may be audited according to applicable policy regardless of the success or failure of the function.

3.2.2 - Input

The request is as specified for the Manage M-Read function with the following additional constraints:

1. The {CDRresourceID} MUST be replaced with the {savedSearchID} as defined in [Section 3.1.3 - Output](#).
2. {MReadProperties} MAY include additional parameters that support configuring optional behavior specific to use for Query Management.

The use of M-Read is otherwise not dependent on its use in the context of QM. The Manage M-Read request example is applicable as is.

3.2.3 - Output

The response MUST conform to the Manage M-Read function with the appropriate inclusion of the following:

1. {CDRresource} MUST be replaced by a saved search resource {saved search resource} as specified by {savedSearchType}.
2. {description} MUST be replaced with the details reflecting both the information provided by the M-Create consumer per [Section 3.1.2 - Input](#) and that automatically maintained by the QM Collection.

The use of M-Read is otherwise not dependent on its use in the context of QM. The Manage M-Read response example is applicable as is.

3.2.4 - Post-conditions

The following conditions MUST be met upon completion of M-Read:

1. The Saved Search is not affected by M-Read.
2. The use of this function has been audited according to applicable policy.⁴

3.2.5 - Fault Conditions

Faults as defined in the CDR Manage specification for the M-Read function are also applicable to the QM use of M-Read. The corresponding QM term as identified in [Table 1](#) MAY be substituted for CDR Manage term used in the fault text.

3.3 - QM Profile of M-Update Function

A QM Service MUST implement the M-Update Function as defined in this section. The relationship between Manage and QM is shown in [Table 1](#).

⁴The Read function may be audited according to applicable policy regardless to the success or failure of the function.

The M-Update function used for QM allows a Consumer Component to change an existing Saved Search. The Saved Search ID uniquely identifies the Saved Search to be modified. Partial updates are not allowed; therefore the M-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; the Saved Search type will remain the same. It MAY be necessary to retrieve the Saved Search prior to performing the update.

3.3.1 - Preconditions

The following preconditions MUST be satisfied if the QM use of the M-Update 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 QM use of the M-Update Function implementation.
2. The Saved Search instance can be accessed through reference to its Saved Search ID for purposes of M-Update.

3.3.2 - Input

The input is as specified for the Manage M-Update function with the following additional constraints:

1. The {CDRresourceID} MUST be replaced with the {savedSearchID} as defined in [Section 3.1.3 - Output](#).
2. {MUpdateProperties} MAY include additional parameters that support configuring optional behavior specific to use for Query Management.
3. {CDRresource} MUST be replaced by a saved search resource {saved search resource} as specified by {savedSearchType}.
4. {descriptionVocabulary} MUST be replaced by a URI that indicates a vocabulary that adequately describes a saved search resource.
5. {description} MUST be replaced by description information {saved search description} that corresponds as specified to {descriptionVocabulary} or the specified default.

[Figure 6](#) shows an example of the QM specialization of an M-Update request. In this example, several {} fields appear for which resource-specific substitutions are needed. In addition, {XML payload} is the same as <cdrm:CDRresource> and its child elements as shown earlier in the example, but is used for brevity.

```

http://CDR.org/CDRresource/1234&output=all
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
  {saved search resource}
  <cdrm:description descriptionVocabulary="{descriptionVocabulary}">
    {saved search description}
  </cdrm:description>
</cdrm:CDRresource>

results in

PUT /CDRresource/1234?output=all HTTP/1.1
Host: CDR.org
Content-Type: application/xml
Content-Length: nnn
{XML payload}

```

Figure 6 : QM Specialization of M-Update Request Example

3.3.3 - Output

The response MUST conform to the Manage M-Update function with the appropriate inclusion of the following:

1. The {CDRresourceID} of the CDR Resource Identifier MUST be replaced with the {savedSearchID} as defined in [Section 3.1.3 - Output](#).
2. {CDRresource} MUST be replaced by a saved search resource {saved search resource} as specified by {savedSearchType}.
3. {description} MUST be replaced with the details reflecting both the information provided by the M-Create consumer per [Section 3.1.2 - Input](#) and that automatically maintained by the QM Collection.

[Figure 7](#) shows an M-Update response example that corresponds to the M-Update request example shown in [Figure 6](#). It includes several { } fields for which resource-specific substitutions are needed.

```
HTTP/1.1 200 OK
  Content-Length: nnn
  Content-Type: application+xml
  Location: http://CDR.org/CDRresource/1234
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:MResponse xmlns:cdrm="...">
  <cdrm:id> http://CDR.org/CDRresource/1234 </cdrm:id>
  {saved search resource}
  <cdrm:description>
    {saved search description}
  </cdrm:description>
</cdrm:MResponse>
```

Figure 7 : QM Specialization of M-Update Response Example

3.3.4 - Post-conditions

The following conditions MUST be met upon completion of M-Update:

1. The Saved Search reflects specified updates.
2. The Saved Search is accessible by the Saved Search ID.
3. The use of this function has been audited according to applicable policy.⁵

3.3.5 - Fault conditions

Faults as defined in the CDR Manage specification for the M-Update function are also applicable to the QM use of M-Update. The corresponding QM term as identified in [Table 1](#) MAY be substituted for CDR Manage term used in the fault text.

3.4 - QM Profile of M-Delete Function

A QM Service MUST implement the M-Delete Function as defined in this section. The relationship between Manage and QM is shown in [Table 1](#).

The M-Delete function removes a Saved Search resource instance and its description from the Saved Search collection managed by the QM Component. [Section 2.4 - QM Use of M-Delete](#) references a discussion of the design considerations related to the M-Delete function.

3.4.1 - Preconditions

The following preconditions MUST be satisfied if the QM use of the M-Delete 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 QM use of the M-Delete Function implementation.

⁵The Update function may be audited according to applicable policy regardless to the success or failure of the function.

2. The Saved Search instance can be accessed through reference to its Saved Search ID for purposes of M-Delete.

3.4.2 - Input

The request is as specified for the Manage M-Create function with the following additional constraints:

1. The {CDRresourceID} MUST be replaced with the {savedSearchID} as defined in [Section 3.1.3 - Output](#).
2. {MDeleteProperties} MAY include additional parameters that support configuring optional behavior specific to use for Query Management.

The use of M-Delete is otherwise not dependent on its use in the context of QM. The Manage M-Delete request example is applicable as is.

3.4.3 - Output

The response MUST conform to the Manage M-Delete function, and the response example is applicable as is.

3.4.4 - Post-conditions

The following conditions MUST be met upon completion of M-Delete:

1. The Saved Search is no longer accessible by Manage and QM functions.
2. The use of this function has been audited according to applicable policy.⁶

3.4.5 - Fault Conditions

Faults as defined in the CDR Manage specification for the M-Delete function are also applicable to the QM use of M-Delete. The corresponding QM term as identified in [Table 1](#) MAY be substituted for CDR Manage term used in the fault text.

3.5 - QM Profile of M-Search Function

A QM Service SHOULD implement the M-Search Function as defined in this section. The relationship between Manage and QM is shown in [Table 1](#).

The M-Search function as applied to Query Management provides the capability of listing or searching the QM Collection, which is the repository of Saved Searches. M-Search MUST be compliant with CDR Search Interface as specified in CDR-RS, [\[5\]](#) CDR-SS, [\[11\]](#) CDR-RB, [\[2\]](#) or CDR-SB. [\[6\]](#) The search terms will be those appropriate to searching for Saved Searches, and SHOULD correspond to a description vocabulary as described in the Describe service specifications, CDR-RS, [\[5\]](#) CDR-SS, [\[11\]](#) CDR-RB, [\[2\]](#) or CDR-SB. [\[6\]](#)

⁶The Delete function may be audited according to applicable policy regardless to the success or failure of the function.

The relevant preconditions, inputs, outputs, post-condition, and faults are as specified in CDR-RS, [5] CDR-SS, [11] CDR-RB, [2] or CDR-SB. [6]

3.6 - QM-Execute Function

A QM Service SHOULD implement the QM-Execute Function as defined in this section. This function is RECOMMENDED and NOT REQUIRED. While including this function will enable an implementation to support a seamless, single request for choosing and executing an identified Saved Search, some implementations may choose to retrieve the Saved Search (using M-Read) and subsequently execute the search using the CDR Search function. The details of a multi-step execution are outside the scope of this specification.

The QM-Execute function leverages a CDR Search or CDR Brokered 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 modification of the CDR Search Interfaces as specified in the CDR-RS, [5] CDR-SS, [11] CDR-RB, [2] or CDR-SB. [6] with the Query expression or search terms being replaced by a Saved Search ID. The Saved Search Consumer identifies the saved search to be executed and invokes submitting the Search Request to the Target Search Capability.

Except as noted below, the QM-Execute Function MUST conform to the relevant preconditions, inputs, outputs, post-condition, and faults as specified in CDR-RS, [5] CDR-SS, [11] CDR-RB, [2] or CDR-SB. [6] depending on whether the REST (OpenSearch) or SOAP encodings are used. The following will specify the REST encoding for QM-Execute. This does not preclude a QM implementation from supporting both the SOAP and the REST encodings.

3.6.1 - Preconditions

In addition to the preconditions listed in CDR-RS, [5] CDR-SS, [11] CDR-RB, [2] and CDR-SB, [6] the following preconditions MUST be satisfied if the execute 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 QM-Execute Function implementation.
2. The Saved Search can be accessed through reference to its Saved Search ID for the purposes of QM-Execute.

3.6.2 - Input

The QM-Execute function is the use of an HTTP/HTTPS GET method, acting on a single information resource, as identified by a URL.

3.6.2.1 - HTTP Method

The Execute function MUST use the HTTP GET method.

3.6.2.2 - URI Template

The URI used to access the QM-Execute function MUST conform to the following: ⁷

```
http://{anyAuthority}/{anyHierarchy}/savedSearch/{savedSearchID}/  
searchResults?{SearchProperties}&{QMLExecuteProperties}
```

where

{anyAuthority} and {anyHierarchy} are unconstrained in this specification beyond what is discussed in the syntax section of Uniform Resource Identifier (URI): Generic Syntax.^[17] However, the URL path MUST end with /savedSearch /{savedSearchID}/searchResults before the “?”.

{savedSearchID} is defined in [Section 3.1.3 - Output](#).

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

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

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 - QM-Execute Request Example

[Figure 8](#) shows an example using the CDR Resource Identifier returned in response to the M-Create process.

```
http://CDR.org/savedSearch/1234/searchResults?startIndex=20&timeout=120  
  
results in  
  
GET /savedSearch/1234/searchResults?startIndex=20&timeout=120 HTTP/1.1  
Host: CDR.org
```

Figure 8 : QM-Execute Request Example

Note that in [Figure 8](#) the startIndex of the search request has been specified. This value will override the startIndex specified in the Saved Search.

⁷For example, <http://www.cdr.org/templates/examples/CDRresource/1234/SearchResults?timeout=120> is a conforming URL.

3.6.3 - Output

The response and corresponding output are as specified in CDR-RS, [\[5\]](#) CDR-SS, [\[11\]](#) CDR-RB, [\[2\]](#) and CDR-SB. [\[6\]](#)

3.6.4 - Post-conditions

The post-conditions are as specified in CDR-RS, [\[5\]](#) CDR-SS, [\[11\]](#) CDR-RB, [\[2\]](#) and CDR-SB. [\[6\]](#)

3.6.5 - Fault Conditions

An implementation of the QM-Execute function MAY provide any of the faults listed in [Table 5](#) as a CDR Fault to the consumer. In addition, faults may be returned that correspond to the search execution as defined in the REST encoding for CDR Search. [\[5\]](#)

Table 5 - List of QM-Execute Function Faults

CDR Fault	Fault Description	HTTP Status	HTTP Description
Unauthorized Access	The Consumer is either not authenticated or not authorized to perform the requested function.	403	Forbidden
Resource Instance Not Found	The QM Service cannot retrieve a CDR Resource instance corresponding to the supplied identifier.	404	Not Found
Unsupported QMExecute Properties	The QM Service does not support one or more of the QM Properties.	400	Bad Request
Unsupported QMExecute Properties Value	The QM Service does not support one or more values associated with a QM Property.	400	Bad Request
Service Execution Fault	The QM Service encounters an error during execution.	500	Internal Service Error

Appendix A Feature Summary

The following table summarizes major features by version for QM and all dependent specs. The "Required date" is the date when systems should support a feature based on the specified driver. Executive Orders, ISOO notices, ICDs and other policy documents have a variety of effective dates.

Table 6 - Feature Summary Legend

Key	Description
F	Full (able to comply and verified by spec to some degree)
P	Partial (Able to comply but not verifiable)
N	Non-compliance (Can't comply)
N/A	Not Applicable. Feature is no longer required.
Cell Colors represent the same information as the Key value	

A.1. QM Feature Comparison

Table 7 - QM Feature Comparison

QM Feature Comparison			
Required date	Feature	V1	V2
	Profile of CDR Manage		F

Appendix B Change History

[Table 8](#) summarizes the version identifier history for this DES.

Table 8 - DES Version Identifier History

Doc Revision	Revision Date	Revisions
1.0	August 2011	Initial Release
2.0	14 March 2014	Structured as a profile of CDR Manage. For details of changes, see Section B.1 - V2 Change Summary

B.1 - V2 Change Summary

Significant drivers for Version 2 include:

- Generalize QM functionality to any identified CDR resource

[Table 9](#) summarizes the changes made to V1 in developing V2.

Table 9 - Data Encoding Specification V2 Change Summary

Change	Artifacts changed	Compatibility Notes
Structured as profile of CDR Manage specification	QM Specification	Applied general reusable constructs
Saved search model as specialization of generalized and revised CDR Resource model	QM Specification	Consistent definition and use
Appendix added to define known saved search resource types	QM Specification	Clarity and consistent extensibility
QM-Execute function is RECOMMENDED	QM Specification	Some implementations preferred a two-step process of retrieving saved search and then executing
Updated specification with new format	QM Specification	

Appendix C Saved Search Resource Types (Normative)

This appendix is a normative definition of a number of Saved Search resource types. In particular, the following subsections define the replacement for {CDRresource}. The normative definitions for search-related terms can be found in section 2.3 of the CDR Specification Framework.^[7]

Additional Saved search resource types MAY be defined external to this specification.

C.1 - SavedSearch QMv1 (urn:cdr:resourceType:qmv1)

This Saved Search resource type is as specified in the Query Management V1.0 specifications.^[4] ^[9] HTTP message body and SOAP body definitions and examples in those documents MUST be used as the basis for structure and semantics for the corresponding functions in this specification. For example, the HTTP message body for the QM-Create Function MUST be used for the QM Profile of M-Create function in this specification.

C.2 - Saved Search OS (urn:cdr:resourceType:ssos:1.0)

This Saved Search resource type corresponds to the use of Opensearch {OS} and specifies the corresponding details of the constraints and substitutions to the generic {CDRresource}:

1. resourceType="urn:cdr:resourceType:ssos:1.0" MUST replace resourceType={savedSearchType} in the M-CreateURI Template.
2. /cdrm:CDRresource/cdrqm:savedSearch and its child elements as defined in [Table 10](#) MUST replace /cdrm:CDRresource/{CDR_Resource}.

Table 10 - Elements and Attributes of the Saved Search OS Resource Type

Element/Attribute Name and Description	Support
<p>/cdrm:CDRresource/cdrqm:savedSearch</p> <p>An XML element that replaces /cdrm:CDRresource/{CDR_Resource} and that serves as a wrapper for the information corresponding to the structure and semantics of the Saved Search being created.</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>
<p>/cdrm:CDRresource/cdrqm:SavedSearch/cdrqm:SavedSearchURL</p> <p>URL containing all information necessary for 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 REST Encoding Specification for Search.^[5]</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>

3. [Figure 9](#) provides an example of a M-Create request for a Saved Search OS resource type. {XML payload} is the same as <cdrm:CDRresource> and its child elements as shown earlier in the example, but is used for brevity.

```

http://CDR.org/CDRresource?resourceType=urn:cdr:resourceType:ssos:
1.0&output=all
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="...">
    <cdrs:SavedSearchURL>
      http://example.com/?q=watson+ibm&startIndex=31&count=10
    </cdrs:SavedSearchURL>
  </cdrqm:savedSearch>
  <cdrm:description descriptionVocabulary="urn:cdr:resourceVocab:ss">
    <pol:AllocationPolicy xmlns:pol=http://policy.org>
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:CDRresource>

results in

POST /CDRresource?resourceType=urn:cdr:resourceType:ssos:1.0&output=all
HTTP/1.1
  Host: CDR.org
  Content-Type: application/xml
  Content-Length: nnn
{XML payload}

```

Figure 9 : Example of QM Use of Saved Search OS Resource Type in M-Create Request

This example is also valid for the QM use of the M-Update function except that the HTTP method will be PUT rather than POST.

4. [Figure 10](#) provides a corresponding example of a M-Create response for a Saved Search OS resource type.

```

HTTP/1.1 201 Created
Content-Length: nnn
Content-Type: application+xml
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:MResponse xmlns:cdrm="...">
  <cdrm:id> http://CDR.org/CDRresource/1234 </cdrm:id>
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="...">
    <cdrs:SavedSearchURL>
      http://example.com/?q=watson+ibm&startIndex=31&count=10
    </cdrs:SavedSearchURL>
  </cdrqm:savedSearch>
  <cdrm:description xmlns:desc="urn:cdr:resourceVocab:ss">
    <cdrm:resourceType>
      urn:cdr:resourceType:ssos:1.0
    </cdrm:resourceType>
    <desc:summary> example summary </desc:summary>
    <desc:updated> 2011-07-13T18:30:02Z </desc:updated>
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:MResponse>

```

Figure 10 : Example of QM Use of Saved Search OS Resource Type in M-Create Response

This example is also valid for the QM use of the M-Update function except that the HTTP status code on successful completion is “200 OK” instead of “201 Created”.

The response example corresponds to the request example shown in [Figure 9](#) . As such, both the resource and its description are included in the response to conform to the output=all attribute in the request. In addition, <cdrm:description> contains a number of description properties for illustrative purposes:

- <cdrm:resourceType> is included to capture the CDR Resource type. It is expected that this information will be maintained as part of the description in the CDR Resource Collection, but implementation.
- Description properties in the desc: namespace are examples of the type of information that is not input by the M-Create consumer but will be maintained by the CDR Resource Collection to trace activity related to the CDR Resource. The details of the description vocabulary and implementation details of the CDR Resource Collection are outside the scope of this specification.
- <pol:AllocationPolicy xmlns:pol=http://policy.org"> is a notional entry demonstrating description information supplied on input. It does not represent a particular defined description property.

Including information in the CDR Resource Description does not preclude the information also being maintained as part of the resource instance itself. For example, an implementation may

save the CDR Resource type with the resource. The example as shown includes <cdrm:resourceType> as part of the description in anticipation that this will facilitate search.

C.3 - Saved Search XML (urn:cdr:resourceType:ssxml:1.0)

This Saved Search resource type corresponds to the XML encoding used with SOAP requests and specifies the corresponding details of the constraints and substitutions to the generic {CDRresource}. The examples below illustrate this format used in a non-SOAP context.

1. resourceType="urn:cdr:resourceType:ssxml:1.0" MUST replace resourceType={savedSearchType} in the M-Create URI Template.
2. /cdrm:CDRresource/cdrqm:savedSearch and its child elements as defined in [Table 11](#) MUST replace /cdrm:CDRresource/{CDR_Resource}.

Table 11 - Elements and Attributes of the Saved Search XML Resource Type

Element/Attribute Name and Description	Support
<p>/cdrm:CDRresource/cdrqm:savedSearch</p> <p>An XML element that replaces /cdrm:CDRresource/{CDR_Resource} and that serves as a wrapper for the information corresponding to the structure and semantics of the Saved Search being created.</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>
<p>/cdrm:CDRresource/cdrqm:SavedSearch/cdrs:SearchRequest</p> <p>The attributes and child element convey the information sent by a search consumer that initiates a search. The search request MUST be composed of Search Function inputs as defined by the IC / DoD SOAP Encoding Specification for Search.^[11]</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>
<p>/cdrm:CDRresource/cdrqm:SavedSearch/cdrqm:TargetSearchCapability</p> <p>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.</p>	<p>MAY be supported by service.</p> <p>MAY be included by consumer in input.</p>

3. [Figure 11](#) provides an example of a M-Create request for a Saved Search XML resource type. {XML payload} is the same as <cdrm:CDRresource> and its child elements as shown earlier in the example, but is used for brevity.

```

http://CDR.org/CDRresource?resourceType=urn:cdr:resourceType:ssxml:1.0
      &output=all
<?XML version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="...">
    <cdrs:SearchRequest startIndex="31" count="10"
      responseFormat="urn:cdr:1.0:resultset:atom-1.0">
      <cdrs:Expression queryLanguage="urn:cdr:queryLanguage:keyword">
        watson ibm
      </cdrs:Expression>
    </cdrs:SearchRequest>
    <cdrs:TargetSearchCapability>
      example.com
    </cdrs:TargetSearchCapability>
  </cdrqm:savedSearch>
  <cdrm:description descriptionVocabulary="urn:cdr:resourceVocab:ss">
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:CDRresource>

results in

POST /CDRresource?resourceType=urn:cdr:resourceType:ssxml:1.0&output=all HTTP/
1.1
  Host: CDR.org
  Content-Type: application/xml
  Content-Length: nnn
{XML payload}

```

Figure 11 : Example of QM Use of Saved Search XML Resource Type in M-Create Request

This example is also valid for the QM use of the M-Update function except that the HTTP method will be PUT rather than POST.

4. [Figure 12](#) provides a corresponding example of a M-Create response for a Saved Search XML resource type.

```

HTTP/1.1 201 Created
Content-Length: nnn
Content-Type: application+xml
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:MResponse xmlns:cdrm="...">
  <cdrm:id> http://CDR.org/CDRresource/1234 </cdrm:id>
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="...">
    <cdrs:SearchRequest startIndex="31" count="10"
      responseFormat="urn:cdr:1.0:resultset:atom-1.0">
      <cdrs:Expression queryLanguage="urn:cdr:queryLanguage:keyword">
        watson ibm
      </cdrs:Expression>
    </cdrs:SearchRequest>
    <cdrs:TargetSearchCapability>
      example.com
    </cdrs:TargetSearchCapability>
  </cdrqm:savedSearch>
  <cdrm:description xmlns:desc="urn:cdr:resourceVocab:ss">
    <cdrm:resourceType> urn:cdr:resourceType:ssxml </cdrm:resourceType>
    <desc:summary> example summary </desc:summary>
    <desc:updated> 2011-07-13T18:30:02Z </desc:updated>
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:MResponse>

```

Figure 12 : Example of QM Use of Saved Search XML Resource Type in M-Create Response

This example is also valid for the QM use of the M-Update function except that the HTTP status code on successful completion is “200 OK” instead of “201 Created”.

The response example corresponds to the request example shown in [Figure 11](#) . As such, both the resource and its description are included in the response to conform to the output=all attribute in the request. In addition, <cdrm:description> contains a number of description properties for illustrative purposes:

- <cdrm:resourceType> is included to capture the CDR Resource type. It is expected that this information will be maintained as part of the description in the CDR Resource Collection, but implementation details of the CDR Resource Collection are outside the scope of this specification.
- Description properties in the desc: namespace are examples of the type of information that is not input by the M-Create consumer but will be maintained by the CDR Resource Collection to trace activity related to the CDR Resource. The details of the description vocabulary and implementation details of the CDR Resource Collection are outside the scope of this specification.
- <pol:AllocationPolicy xmlns:pol=http://policy.org"> is a notional entry demonstrating description information supplied on input. It does not represent a particular defined description property.

Including information in the CDR Resource Description does not preclude the information also being maintained as part of the resource instance itself. For example, an implementation may save the CDR Resource type with the resource. The example as shown includes <cdrm:resourceType> as part of the description in anticipation that this will facilitate search.

C.4 - Saved Search OS Broker (urn:cdr:resourceType:ssosb:1.0)

This Saved Search resource type corresponds to the use of OpenSearch (OS) for requests to search brokers and specifies the details of the constraints and substitutions to the generic {CDRresource}.

1. resourceType="urn:cdr:resourceType:ssosb:1.0" MUST replace resourceType={savedSearchType} in the M-Create URI Template.
2. /cdrm:CDRresource/cdrqm:savedSearch and its child elements as defined in [Table 12](#) MUST replace /cdrm:CDRresource/{CDR_Resource}.

Table 12 - Elements and Attributes of the Saved Search OS Broker Resource Type

Element/Attribute Name and Description	Support
<p>/cdrm:CDRresource/cdrqm:savedSearch</p> <p>An XML element that replaces /cdrm:CDRresource/{CDR_Resource} and that serves as a wrapper for the information corresponding to the structure and semantics of the Saved Search being created.</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>
<p>/cdrm:CDRresource/cdrqm:SavedSearch/cdrqm:SavedSearchURL</p> <p>URL containing all information necessary for a search consumer to initiate a search. The URL location MUST reference a valid OpenSearch Search Component implementation and the search request MUST be composed of Search Function inputs as defined by the IC / DoD Content Discovery & Retrieval REST Interface Encoding Specification for Content Discovery and Retrieval Brokered Search.^[2]</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>

3. [Figure 13](#) provides an example of a M-Create request for a Saved Search OS Broker resource type. {XML payload} is the same as <cdrm:CDRresource> and its child elements as shown earlier in the example, but is used for brevity.

```

http://CDR.org/CDRresource?resourceType=urn:cdr:resourceType:ssosb:1.0
&output=all
<?XML version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="..." xmlns:cdrb="...">
    <cdrs:SavedSearchURL>
      http://example.com/?q=watson+ibm&startIndex=31&count=10&src=abc,def
    </cdrs:SavedSearchURL>
  </cdrqm:savedSearch>
  <cdrm:description descriptionVocabulary="urn:cdr:resourceVocab:ss">
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:CDRresource>

results in

POST /CDRresource?resourceType=urn:cdr:resourceType:ssosb:1.0&output=all HTTP/
1.1
Host: CDR.org
Content-Type: application/xml
Content-Length: nnn
{XML payload}

```

Figure 13 : Example of QM Use of Saved Search OS Broker Resource Type in M-Create Request

This example is also valid for the QM use of the M-Update function except that the HTTP method will be PUT rather than POST.

4. [Figure 14](#) provides a corresponding example of a M-Create response for a Saved Search OS Broker resource type.

```

HTTP/1.1 201 Created
Content-Length: nnn
Content-Type: application+xml
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:MResponse xmlns:cdrm="...">
  <cdrm:id> http://CDR.org/CDRresource/1234 </cdrm:id>
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="...">
    <cdrs:SavedSearchURL>
      http://example.com/?q=watson+ibm&startIndex=31&count=10&src=abc,def
    </cdrs:SavedSearchURL>
  </cdrqm:savedSearch>
  <cdrm:description xmlns:desc="urn:cdr:resourceVocab:ss">
    <cdrm:resourceType> urn:cdr:resourceType:ssosb:1.0 </cdrm:resourceType>
    <desc:summary> example summary </desc:summary>
    <desc:updated> 2011-07-13T18:30:02Z </desc:updated>
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:MResponse>

```

Figure 14 : Example of QM Use of Saved Search OS Broker Resource Type in M-Create Response

This example is also valid for the QM use of the M-Update function except that the HTTP status code on successful completion is “200 OK” instead of “201 Created”.

The response example corresponds to the request example shown in [Figure 13](#) . As such, both the resource and its description are included in the response to conform to the output=all attribute in the request. In addition, <cdrm:description> contains a number of description properties for illustrative purposes:

- <cdrm:resourceType> is included to capture the CDR Resource type. It is expected that this information will be maintained as part of the description in the CDR Resource Collection, but implementation details of the CDR Resource Collection are outside the scope of this specification.
- Description properties in the desc: namespace are examples of the type of information that is not input by the M-Create consumer but will be maintained by the CDR Resource Collection to trace activity related to the CDR Resource. The details of the description vocabulary and implementation details of the CDR Resource Collection are outside the scope of this specification.
- <pol:AllocationPolicy xmlns:pol=http://policy.org"> is a notional entry demonstrating description information supplied on input. It does not represent a particular defined description property.

Including information in the CDR Resource Description does not preclude the information also being maintained as part of the resource instance itself. For example, an implementation may

save the CDR Resource type with the resource. The example as shown includes <cdrm:resourceType> as part of the description in anticipation that this will facilitate search.

C.5 - Saved Search XML Broker (urn:cdr:resourceType:ssxmlb:1.0)

This Saved Search resource type corresponds to the XML encoding used with SOAP requests to search brokers and specifies the details of the constraints and substitutions to the generic {CDRresource}. The examples below illustrate this format used in a non-SOAP context.

1. resourceType="urn:cdr:resourceType:ssxmlb:1.0" MUST replace resourceType={savedSearchType} in the M-Create URI Template.
2. /cdrm:CDRresource/cdrqm:savedSearch and its child elements as defined in [Table 13](#) MUST replace /cdrm:CDRresource/{CDR_Resource}.

Table 13 - Elements and Attributes of the Saved Search XML Broker Resource Type

Element/Attribute Name and Description	Support
<p>/cdrm:CDRresource/cdrqm:savedSearch</p> <p>An XML element that replaces /cdrm:CDRresource/{CDR_Resource} and that serves as a wrapper for the information corresponding to the structure and semantics of the Saved Search being created.</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>
<p>/cdrm:CDRresource/cdrqm:SavedSearch/cdrs:SearchRequest</p> <p>The attributes and child element convey the information sent by a search consumer that initiates a search. The search request MUST be composed of Search Function inputs as defined by the IC / DoD Content Discovery & Retrieval SOAP Interface Encoding Specification for Content Discovery and Retrieval Brokered Search.^[6]</p>	<p>MUST be supported by service.</p> <p>MUST be included by consumer in input.</p>
<p>/cdrm:CDRresource/cdrqm:SavedSearch/cdrqm:TargetSearchCapability</p> <p>Reference to the 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.</p>	<p>MAY be supported by service.</p> <p>MAY be included by consumer in input.</p>

3. [Figure 15](#) provides an example of a M-Create request for a Saved Search XML Broker resource type. {XML payload} is the same as <cdrm:CDRresource> and its child elements as shown earlier in the example, but is used for brevity.

```

http://CDR.org/CDRresource?resourceType=urn:cdr:resourceType:ssxmlb:1.0
      &output=all
<?XML version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="..." xmlns:cdrb="...">
    <cdrs:SearchRequest startIndex="31" count="10" cdrb:routeTo="abc,def"
      responseFormat="urn:cdr:1.0:resultset:atom-1.0">
      <cdrs:Expression queryLanguage="urn:cdr:queryLanguage:keyword">
        watson ibm
      </cdrs:Expression>
    </cdrs:SearchRequest>
    <cdrs:TargetSearchCapability>
      example.com
    </cdrs:TargetSearchCapability>
  </cdrqm:savedSearch>
  <cdrm:description descriptionVocabulary="urn:cdr:resourceVocab:ss">
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:CDRresource>

results in

POST /CDRresource?resourceType=urn:cdr:resourceType:ssxmlb:1.0&output=all
HTTP/1.1
Host: CDR.org
Content-Type: application/xml
Content-Length: nnn
{XML payload}

```

Figure 15 : Example of QM Use of Saved Search XML Broker Resource Type in M-Create Request

This example is also valid for the QM use of the M-Update function except that the HTTP method will be PUT rather than POST:

4. [Figure 16](#) provides a corresponding example of a M-Create response for a Saved Search OS Broker resource type.

```

HTTP/1.1 201 Created
Content-Length: nnn
Content-Type: application+xml
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:MResponse xmlns:cdrm="...">
  <cdrm:id> http://CDR.org/CDRresource/1234 </cdrm:id>
  <cdrqm:savedSearch xmlns:cdrqm="..." xmlns:cdrs="...">
    <cdrs:SearchRequest startIndex="31" count="10" cdrb:routeTo="abc,def"
      responseFormat="urn:cdr:1.0:resultset:atom-1.0">
      <cdrs:Expression queryLanguage="urn:cdr:queryLanguage:keyword">
        watson ibm
      </cdrs:Expression>
    </cdrs:SearchRequest>
    <cdrs:TargetSearchCapability>
      example.com
    </cdrs:TargetSearchCapability>
  </cdrqm:savedSearch>
  <cdrm:description xmlns:desc="urn:cdr:resourceVocab:ss">
    <cdrm:resourceType> urn:cdr:resourceType:ssxmlb:1.0 </cdrm:resourceType>
    <desc:summary> example summary </desc:summary>
    <desc:updated> 2011-07-13T18:30:02Z </desc:updated>
    <pol:AllocationPolicy xmlns:pol=http://policy.org">
      JointIC-DoDPolicyDefaults
    </pol:AllocationPolicy>
  </cdrm:description>
</cdrm:MResponse>

```

Figure 16 : Example of QM Use of Saved Search OS Broker Resource Type in M-Create Response

This example is also valid for the QM use of the M-Update function except that the HTTP status code on successful completion is “200 OK” instead of “201 Created”.

The response example corresponds to the request example shown in [Figure 15](#). As such, both the resource and its description are included in the response to conform to the output=all attribute in the request. In addition, <cdrm:description> contains a number of description properties for illustrative purposes:

- <cdrm:resourceType> is included to capture the CDR Resource type. It is expected that this information will be maintained as part of the description in the CDR Resource Collection, but implementation details of the CDR Resource Collection are outside the scope of this specification.
- Description properties in the desc: namespace are examples of the type of information that is not input by the M-Create consumer but will be maintained by the CDR Resource Collection to trace activity related to the CDR Resource. The details of the description vocabulary and implementation details of the CDR Resource Collection are outside the scope of this specification.
- <pol:AllocationPolicy xmlns:pol=http://policy.org"> is a notional entry demonstrating description information supplied on input. It does not represent a particular defined description property.

Including information in the CDR Resource Description does not preclude the information also being maintained as part of the resource instance itself. For example, an implementation may save the CDR Resource type with the resource. The example as shown includes `<cdrm:resourceType>` as part of the description in anticipation that this will facilitate search.

Appendix D Mapping to Specification Framework

As a profile of the REST Manage specification,^[3] the appendix in that document showing mapping between the CDR Specification Framework and REST Manage is applicable to this specification. In this document, Query Management parameters and concepts then describe use of the mapped Manage constructs.

In addition, the QM-Execute function provides a capability specific to QM, and that requires a separate mapping to explicitly tie the QM-Execute items to the requirements of the CDR Specification Framework. At the time of the publication of this document, the CDR-SF has not been updated to reflect the specification of the Manage Component or the recasting of the Query Management Component as a profile of Manage. A draft of those changes indicates the QM-Execute section of the current CDR-SF will remain applicable and [Table 14](#) and [Table 15](#) show a mapping against the parameters defined there.

Table 14 - Mapping to CDR Specification Framework Input Variables

Specification Framework Variables	REST QM Specification
Modified Search Function Inputs	http://{anyAuthority}/{anyHierarchy}/savedSearch/{savedSearchID}/searchResults ? {SearchProperties}&{QMExecuteProperties}
Saved Search ID	{savedSearchID}
QM Properties	{QMExecuteProperties}

Table 15 - Mapping to CDR Specification Framework Output Variables

Specification Framework Variables	REST QM Specification
Search Results	The response and corresponding output are as specified in CDR-RS, ^[5] CDR-SS, ^[11] CDR-RB, ^[2] and CDR-SB. ^[6]

Appendix E Glossary

This appendix lists all the acronyms and abbreviations referenced in this encoding specification.

CDR	Content Discovery and Retrieval
CDR-SM	Content Discovery & Retrieval - SOAP Manage
CDR-RA	Content Discovery & Retrieval - Reference Architecture
CDR-RB	Content Discovery & Retrieval - REST Brokered Search
CDR-RM	Content Discovery & Retrieval - REST Manage
CDR-RS	Content Discovery & Retrieval - REST Search
CDR-SB	Content Discovery & Retrieval - Brokered Search
CDR-SF	Content Discovery & Retrieval - Specification Framework
CDR-SS	Content Discovery & Retrieval - SOAP Search
CIO	Chief Information Officer
CVE	Controlled Vocabulary Enumeration
DNI	Director of National Intelligence
DOD	Department of Defense
HTTP	Hypertext Transfer Protocol
IC	Intelligence Community
IC CIO	Intelligence Community Chief Information Officer
IC ITE	IC Information Technology Enterprise
ICD	Intelligence Community Directive
ICS	Intelligence Community Standard
IPT	Integrated Project Team
ISOO	Information Security Oversight Office
JSON	JavaScript Object Notation
OCIO	Office of the Intelligence Community Chief Information Officer
QM	Query Management
REST	Representational State Transfer

URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
XML	Extensible Markup Language
XSL	Extensible Stylesheet Language

Appendix F Bibliography

Bibliography

[1] CDR-RA

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *Content Discovery and Retrieval Integrated Project Team Reference Architecture (CDR-RA)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RA>

Available online at: <http://purl.org/IC/Standards/public>

[2] CDR-RB

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Data Encoding Specification for Content Discovery and Retrieval: Brokered Search (CDR-RB)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RB>

Available online at: <http://purl.org/IC/Standards/public>

[3] CDR-RM

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Interface Specification for Content Discovery and Retrieval: Manage (CDR-RM)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RM>

Available online at: <http://purl.org/IC/Standards/public>

[4] CDR-RQM V1

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Encoding Specification for CDR Query Management*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RQM>

Available online at: <http://purl.org/IC/Standards/public>

[5] CDR-RS

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Interface Encoding Specification for Content Discovery and Retrieval: Search (CDR-RS)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-RS>

Available online at: <http://purl.org/IC/Standards/public>

[6] CDR-SB

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Data Encoding Specification for Content Discovery and Retrieval: Brokered Search (CDR-SB)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SB>

Available online at: <http://purl.org/IC/Standards/public>

[7] CDR-SF

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *Data Encoding Specification for Content Discovery and Retrieval: Specification Framework (CDR-SF)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SF>

Available online at: <http://purl.org/IC/Standards/public>

[8] CDR-SM

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Interface Encoding Specification for Content Discovery and Retrieval: Manage (CDR-SM)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SM>

Available online at: <http://purl.org/IC/Standards/public>

[9] CDR-SQM V1

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Encoding Specification for CDR Query Management*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SQrM> [<http://purl.org/IC/Standards/CDR-SQM>]

Available online at: <http://purl.org/IC/Standards/public>

[10] CDR-SQM

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Interface Encoding Specification for Content Discovery and Retrieval: Query Management (CDR-SQM)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SQrM> [<http://purl.org/IC/Standards/CDR-SQM>]

Available online at: <http://purl.org/IC/Standards/public>

[11] CDR-SS

Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Interface Encoding Specification for Content Discovery and Retrieval: Search (CDR-SS)*.

Available online Intelink-U at: <http://purl.org/IC/Standards/CDR-SS>

Available online at: <http://purl.org/IC/Standards/public>

[12] IC ITE INC1 IMPL

Office of the Director of National Intelligence. *Intelligence Community Information Technology Enterprise (IC ITE) Increment 1 Implementation Plan*. July 2012.

Available online Intelink-TS at: <http://go.ic.gov/HvBHBmY>

[13] ICD 500

Office of the Director of National Intelligence. *Director of National Intelligence Chief Information Officer*. Intelligence Community Directive 500. 7 August 2008.

Available online Intelink-TS at: <http://go.ic.gov/enm8L9x>

Available online at: http://www.dni.gov/files/documents/ICD/ICD_500.pdf

[14] ICPG 710.1

Assistant Director of National Intelligence for . *Application of Dissemination Controls: Originator Control*. Intelligence Community Policy Guidance 710.1. 25 July 2012.

Available online Intelink-TS at: <http://go.ic.gov/yAqVQ0H>

[15] ICS 500-20

Director of National Intelligence Chief Information Officer. *Intelligence Community Enterprise Standards Compliance*. Intelligence Community Standard 500-20. 16 December 2010.

Available online Intelink-TS at: <http://go.ic.gov/QUDIJkZ>

Available online Intelink-U at: https://intelshare.intelink.gov/sites/odni/cio/ea/library/Data%20Specifications/500-21/500_20_signed_16DEC2010.pdf

[16] IETF-RFC 2119

Internet Engineering Task Force. *Key words for use in RFCs to Indicate Requirement Levels*. March 1997.

Available online at: <http://tools.ietf.org/html/rfc2119>

[17] IETF-RFC 3986

Internet Engineering Task Force. *Uniform Resource Identifier (URI): Generic Syntax*. January 2005.

Available online at: <http://tools.ietf.org/html/rfc3986>

[18] IETF-RFC 4287

M. Nottingham, R. Sayre. *The Atom Syndication Format*. December 2005.

Available online at: <http://www.ietf.org/rfc/rfc4287.txt>

[19] Joint IC/DoD Memorandum

Intelligence Community Chief Information Officer, and Department of Defence Chief Information Officer. *IC and DoD Commitment to an Interoperable Service-Based Environment*. 13 July 2007.

Available online at: <http://www.docstoc.com/docs/797594/Department-of-Defense-DoD-and-Intelligence-Community-IC-Commitment-to-an-Interoperable-Services-Based-Environment---Enterprise-Services>

Appendix G Points of Contact

The Intelligence Community Chief Information Officer (IC CIO) facilitates one or more collaboration and coordination forums charged with the adoption, modification, development, and governance of IC technical specifications of common concern. This technical specification was produced by the IC CIO and coordinated with these forums, approved by the IC CIO or a designated representative, and made available at DNI -sponsored web sites. Direct all inquiries about this IC technical specification to the IC CIO, an IC technical specification collaboration and coordination forum, or IC element representatives involved in those forums.

Public Website: <http://purl.org/ic/standards/public>

E-mail: ic-standards-support@intelink.gov [mailto:ic-standards-support@intelink.gov].

Appendix H IC CIO Approval Memo

An Office of the Intelligence Community Chief Information Officer (OCIO) Approval Memo should accompany this enterprise technical data specification bearing the signature of the Intelligence Community Chief Information Officer (IC CIO) or an IC CIO -designated official(s). If an OCIO Approval Memo is not accompanying this specification's version release package, then refer back to the authoritative web location(s) for this specification to see if a more complete package or a specification update is available.

Specification artifacts display a date representing the last time a version's artifacts as a whole were modified. This date most often represents the conclusion of the IC Element collaboration and coordination process. Once the IC Element coordination process is complete, the specification goes through an internal OCIO staffing and coordination process leading to signature of the OCIO Approval Memo. The signature date of the OCIO Approval Memo will be later than the last modified date shown on the specification artifacts by an indeterminable time period.

Upon signature of the OCIO Approval Memo, IC Elements may begin to use this specification version in order to address mission and business objectives. However, it is critical for IC Elements, prior to disseminating information encoded with this new specification version, to ensure that key enterprise services and consumers are prepared to accept this information. IC Elements should work with enterprise service providers and consumers to orchestrate an orderly implementation transition to this specification version in concert with mandatory and retirement usage decisions captured in the IC Enterprise Standards Baseline as defined in Intelligence Community Standard (ICS) 500-20.^[15]