Intelligence Community and Department of Defense Technical Specification

REST Service Encoding Specification for Content Discovery and Retrieval: Manage

Version 1

09 May 2014

Table of Contents

List of Figures

List of Tables

Chapter 1 - Introduction

1.1 - Service Overview

The Manage Component, as defined by the IC / DoD Content Discovery and Retrieval (CDR) Specification Framework (CDR-SF)[6], serves as the primary mechanism to manage CDR Resources, where a CDR Resource is defined as one explicitly created and used to support CDR functions. A Saved Search is an example of a CDR Resource; in particular, a Saved Search is used by the Query Management Execute (QM-Execute) function.

This specification defines requirements and provides guidance for the realization of the CDR Manage Component, hereafter termed the Manage Service in this document, as a RESTful [1] web service. The content of this specification describes the Manage Service's behavior, interface and other aspects in detail, providing enough information for Manage Service providers and consumers to create and use CDR- conformant Manage Services. Specific uses of the Manage Service, such as to create, read, update, delete, and search for Saved Searches, will be elaborated as profiles in the corresponding documents for those uses. The Manage Service interfaces may be implemented to act on any CDR Resource type but actual use will always be in the context of a particular identified resource type.

The Manage Service provides a coordinated set of functions that enables service consumers to create, read, update, delete, and search for instances of any defined type of CDR Resources. The resource model presented in Figure 1 provides an overview of the information that supports Manage functionality. The CDR Resource type corresponding to specific uses of Manage is associated with a uniform resource identifier (URI) (see Table 2), where the Web-accessible resource accessed through that URI will identify the structure and semantics of the CDR Resource type designed for that use. For example, Query Management (QM) defines the Saved Search type as the CDR Resource relevant to that use.

For all uses, the CDR Resource Description shown on the left-hand-side of Figure 1 comprises the characteristic description metadata that aids in the discovery of CDR Resource instances. Some of this description, e.g., the date the resource was created in CDR Resource Collection, will be generated as part of the resource creation or update. Other description, such as a link to applicable policies, will be supplied by someone with responsibility for the resource. It is anticipated that a basic description vocabulary appropriate for any CDR Resource will contain a general set of properties (e.g., last modification date) while the description vocabulary associated with a particular resource type will add additional properties (e.g., query language for a Saved Search instance).

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

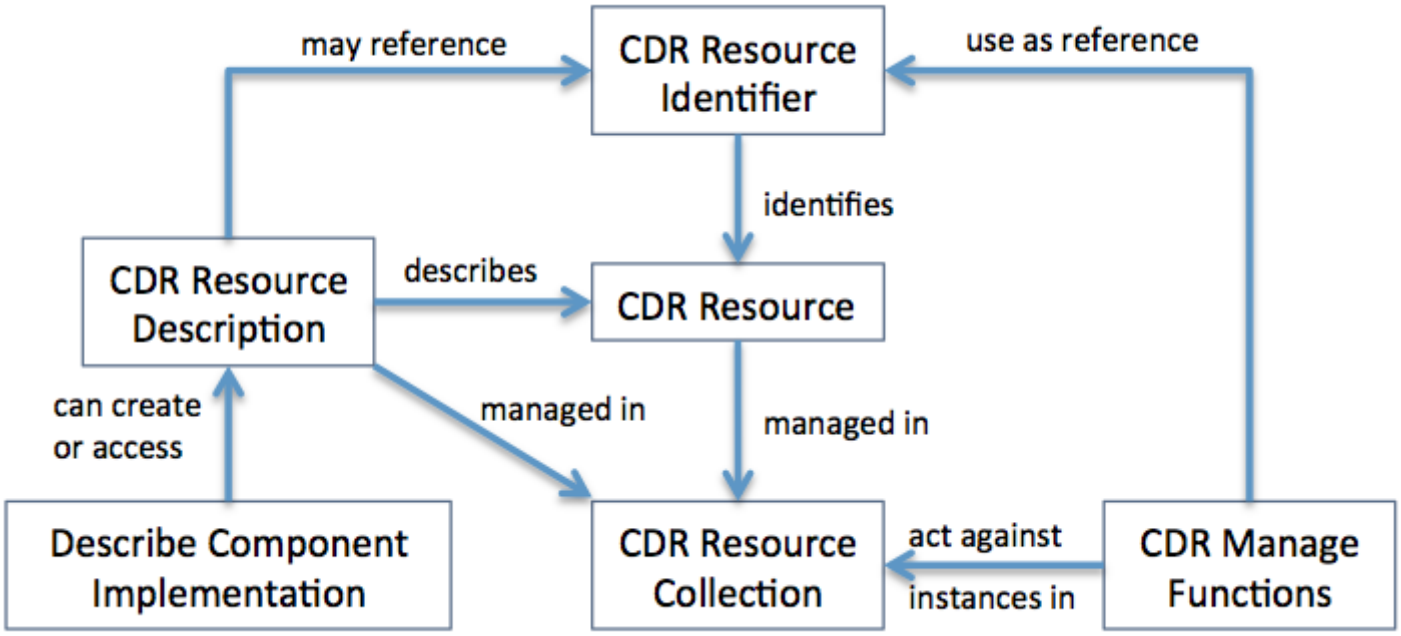

Figure 1 : CDR Resource Model

The ability to save and retrieve resource instances over time will require implementers to adopt a persistence mechanism, which this document refers to as a CDR Resource Collection. The implementation of the CDR Resource Collection is not in the scope of this document but the logical construct of a CDR Resource Collection MUST NOT be interpreted as requiring a separate physical implementation for each CDR Resource type.

This document specifies the standard interfaces to the functionality provided by the Manage Service. The CDR Manage specifications define the functions:

- M-Create
- M-Read
- M-Update
- M-Delete
- M-Search

The M- prefix is used to emphasize that these functions are defined for managing CDR Resources. While these functions may have wider applicability, defining such applicability is beyond the current scope.

1.2 - Scope

This specification is limited to the interactions that occur between an Initiating Consumer and the Manage Service as described in the CDR Reference Architecture (CDR-RA) [1] and CDR Specification Framework (CDR-SF) [6].

This specification provides the description of the Manage Service Behavior in terms of the message exchange patterns necessary to enable service consumers to create, read, update, delete, search for CDR Resources.

The scope of this specification is limited as follows:

- Versioning of the managed CDR resources is not defined
- A CDR resource update is a full replacement of the target resource. Partial update of the CDR resource 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.

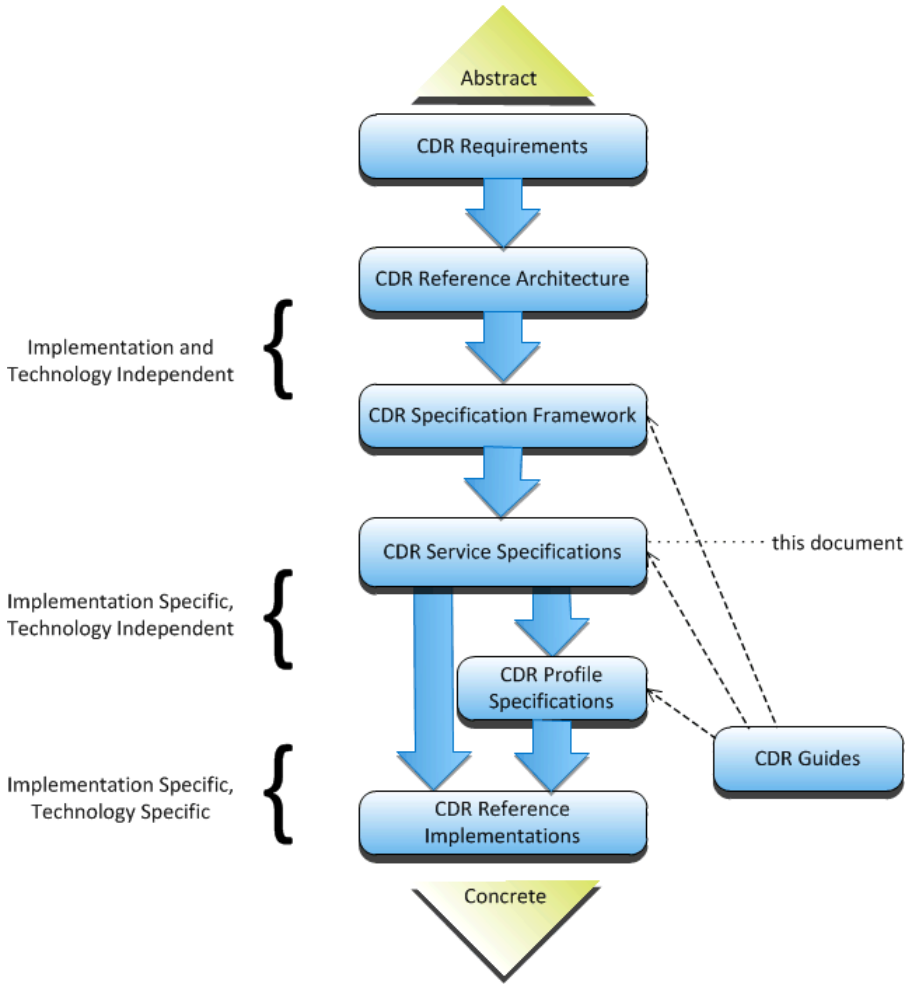

Figure 2 : CDR Architecture Documents

As illustrated in Figure 2 , the CDR-SF [6] 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., Manage) and, for each service, separate specifications to address Representational State Transfer (REST) and SOAP implementations.

This document is a specification for implementing the CDR Manage Service as a REST Web Service. It is intended to parallel the corresponding SOAP specification, the IC / DoD SOAP Interface Encoding Specification for CDR Manage[7], 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[10]
- 500 Series:
  - Intelligence Community Directive (ICD) 500, Director Of National Intelligence Chief Information Officer[11]
  - Joint IC/DoD Memorandum, IC and DoD Commitment to an Interoperable Service-Based Environment (13 Jul 07)[18]

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.[14] 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 brace, generally represented in the form {name}, is meant to be replaced with an actual value determined at run-time. An optional parameter in a URL template is one whose name is followed by ?, e.g., {name?} and 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 1 . The namespace prefix of any XML Qualified Name (QName) used in any example in this document should be interpreted using the information in Table 1 .

Table 1 - Namespaces

Prefix	URI	Description
cdrm	urn:cdr:manage:1	CDR Manage 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 [17]

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.

1.6 - Conformance

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

1.7 - CDR Resource Type and CDR Resource Description URIs

Table 2 and Table 3 list the Uniform Resource Identifiers (URIs) for CDR Resource Types and CDR Resource Description Vocabularies that are currently defined as part of or recognized by the CDR specification set. [2] These are accepted values for the resourceType parameter and descriptionVocabulary attribute defined in Section 3.1.2 - Input . Use of the term CDR Resource throughout this specification should be interpreted as referring to any of the resource types identified in Table 2 or subsequently defined as additional resource types.

Table 2 and Table 3 reference CDR Resource Types and description vocabularies by name and specify a URI that uniquely identifies each CDR Resource type and vocabulary. The URI may be either Uniform Resource Locator (URL) or Uniform Resource Name (URN) . Each URL MUST link to a resource that defines the structure and semantics of a CDR Resource Type or Description Vocabulary. Each URN MUST be associated with a means, such as a Profile Specification or a CDR Guide, to retrieve the detailed definitions of the CDR Resource Type or description vocabulary.

[2]The Subscription and Saved Result Set resource types are notional in that the corresponding Manage profiles have been discussed but have not been explicitly defined.

Table 2 - CDR Resource Type URIs

Name	URI	Description
Saved Search QM v1	urn:cdr:resourceType:qmv1	Definition for Saved Search using the Atom format as defined in the QM v1.0 specifications
Saved Search OS	urn:cdr:resourceType:ssos:1.0	Definition for Saved Search using OpenSearch format
Saved Search XML	urn:cdr:resourceType:ssxml:1.0	Definition for Saved Search using XML to specify information as payload
Saved Search OS Broker	urn:cdr:resourceType:ssosb:1.0	Definition for Saved Search using OpenSearch format for brokered search
Saved Search XML Broker	urn:cdr:resourceType:ssxmlb:1.0	Definition for Saved Search using XML to specify information as payload for brokered search
Subscription	urn:cdr:resourceType:sub:1.0	Definition for Subscription (notional)
Saved Result Set	urn:cdr:resourceType:srs:1.0	Definition for Saved Result Set (notional)

Table 3 - CDR Resource 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
Subscription	urn:cdr:resourceVocab:sub:1.0	Definition for Subscription vocabulary
Saved Result Set	urn:cdr:resourceVocab:srs:1.0	Definition for Saved Result Set vocabulary

Additional acceptable values for resource types and description vocabularies MAY be defined in the future and 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 Manage, to include appropriate access to CDR Resource 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, Manage behavior is realized through five activities – create, read, update, delete and search – and is accessed through the use of the M-Create, M-Read, M-Update, M-Delete and M-Search interfaces.

2.1 - M-Create

The M-Create function is used to insert a new CDR Resource instance into the CDR Resource Collection. The create process will assign a unique CDR Resource Identifier through which other Manage Functions will refer to the resource instance for subsequent activities. M-Create may also create a CDR Resource Description.

2.2 - M-Read

The M-Read function is used to retrieve a CDR Resource from the CDR Resource Collection. It refers to the CDR Resource through its CDR Resource Identifier. M-Read may also be used to retrieve the CDR Resource Description.

2.3 - M-Update

The M-Update function is used to change a CDR Resource being managed through the CDR Resource Collection. It refers to the CDR Resource through its CDR Resource Identifier. M-Update replaces the existing CDR Resource and the CDR Resource Description with the input provided and does not support partial updates.

2.4 - M-Delete

The M-Delete function is used to remove a CDR Resource instance and its description from the CDR Resource Collection. It refers to the CDR Resource through its CDR Resource Identifier. The M-Delete section of CDR-SF [1] discusses considerations when deleting a CDR Resource. These considerations include but are not limited to whether:

- the CDR Resource instance is logically marked as deleted or physically removed from the CDR Resource Collection;
- the CDR Resource instance can be undeleted;
- the associated description of the CDR Resource instance is deleted or marked as representing a deleted instance;
- the deleted CDR Resource instance can be included in search results of M-Search and under which conditions may it appear;
- the consumer is asked to verify the delete request before it is carried out.

[1] The CDR-SF [6] has yet to be updated to include the Manage Component. The material referred to here will draw from the QM-Delete section of QM v1.0.[3]

The choice of such M-Delete behavior may be configured using M-Delete Properties but such definition of M-Delete Properties is outside the current scope of this specification.

2.5 - M-Search

The M-Search function enables a prospective consumer to interrogate the CDR Resource Collection to determine if a suitable CDR Resource has already been created and is being managed. The search may be based on anything searchable in the CDR Resource or the CDR Resource Description. Generic search terms can be used to provide functionality that responds with a ‘list’ of the contents of the CDR Resource Collection. This capability SHOULD leverage CDR 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 Manage Service and what is expected in terms of a consumer interacting with the service.

3.1 - M-Create Function

A Manage Service MUST implement the M-Create Function.

3.1.1 - Preconditions

The following preconditions MUST be satisfied if 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 the M-Create Function implementation.

2. A CDR Resource collection exists and is available.

3.1.2 - Input

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

3.1.2.1 - HTTP Method

The M-Create function MUST use the HTTP POST method.

3.1.2.2 - URI Template

The URI used to access the M-Create function MUST conform to the following: [2]

```
http://{anyAuthority}/{anyHierarchy}/CDRresource?
resourceType={CDRresourceType}&{MCreateProperties}
```

where

{anyAuthority} and {anyHierarchy} are unconstrained in this specification beyond what is discussed in the syntax section of Uniform Resource Identifier (URI): Generic Syntax[16]. However, the URL path MUST end with /CDRresource before the “?”.

{CDRresourceType} – REQUIRED – MUST be a URI for which the content indicated by the URI defines the structure and semantics of a recognized CDR Resource. (See Section 1.7 - CDR Resource Type and CDR Resource Description URIs .)

[1] The Manage Service is intended to conform as described by the Manage Component section of the CDR-SF. [6]
[2] For example, http://www.cdr.org/templates/examples/CDRresource?resourceType=urn:cdr:resourceType:ssxml &output=all is a conforming URL .

{MCreateProperties} – OPTIONAL – Parameters through which the Manage consumer may specify and configure optional behavior supported by the M-Create function implementation

http://{anyAuthority}/{anyHierarchy}/CDRresource identifies the CDR Resource Collection that acts as the management point for a set of CDR Resource instances. The CDR Resource Collection (or whatever implements the collection entity) implements the CDR Manage interfaces, and these interfaces are the means to access and use the results of CDR Resource management.

M-Create properties {MCreateProperties} provide a means for both configuration and extensibility. Table 4 defines properties related to the cdrm: namespace.

Table 4 - M-Create Properties Defined in cdrm: namespace

M-Create Properties Definition	Support
output The value controls whether the resource or its description is included in the M-Create output or the output of other Manage functions. The description comprises both those description elements generated as part of the resource creation or update and those provided as input during M-Create or M-Update use; see discussion of description on Section 1.1 - Service Overview . Valid values are as follows: • resource: only the resource is included in the output • description: only the description is included in the output • all: both the resource and the description are included in the output • none: neither the resource nor the description is included in the output The default value is none.	MAY be supported by service. MAY be included by consumer in input if supported by service.

Definition and use of M-Create properties MAY be supported by an implementation; if supported, the properties MAY be included by the consumer in the input. An implementation SHOULD ignore properties it does not support. Additional values enabling more selective output or additional M-Create properties may be defined in future versions of this specification.

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.
- Unless overridden by an implementation specific Content-Type, the default Content-Type value SHOULD be ‘application/xml’.

3.1.2.4 - HTTP Message Body

The body of the M-Create request MUST contain a document that corresponds to the value of resourceType, as defined in Section 3.1.2.2 - URI Template . The profiles that elaborate specific uses of the Manage functions will provide additional details regarding the structure and semantics of the message body content.

It is RECOMMENDED [3] that the message body for the M-Create request be an XML document with a <cdrm:CDRresource> root element. The <cdrm:CDRresource> element and its child elements and associated attributes are defined in Table 5 . Unless specifically noted, the remainder of the message body definition that follows will use the <cdrm:CDRresource> structure.

Table 5 - Elements and Attributes of M-Create Request Message Body

Element/Attribute Name and Description	Support
/cdrm:CDRresource An XML element that serves as a wrapper for the information defining the resource being created and the associated description for the resource.	MUST be supported by service. MUST be included by consumer in input.
/cdrm:CDRresource/{CDR_Resource} Information corresponding to the structure and semantics of the specific CDR Resource indicated by the value of {CDRresourceType}(per Section 3.1.2.2 - URI Template).	MUST be supported by service. MUST be included by consumer in input.
/cdrm:CDRresource/description An XML element that serves as a wrapper for the information describing the resource being created.	MAY be supported by service. MAY be included by consumer in input if supported by service.
/cdrm:CDRresource/description/@descriptionVocabulary A URI that identifies a description vocabulary desired by the consumer. If provided, the value MUST be from those included in Table 3 or otherwise defined per Section 1.7 - CDR Resource Type and CDR Resource Description URIs . If not provided, the Service will use its default vocabulary.	MAY be supported by service. MAY be included by consumer in input if supported by service.

[3]The structure that follows is RECOMMENDED and NOT REQUIRED in order to provide a mechanism for compatibility with the <atom:entry> structure defined by version 1.0 of the Query Management specifications. The use of <atom:entry> instead of <cdrm:CDRresource> as the root element is elaborated in the Query Management v1.0 specifications.[3]

Element/Attribute Name and Description	Support
/cdrm:CDRresource/description/{description} Information describing the CDR Resource. The structure and semantics of the description MUST conform to the vocabulary identified in /cdrm:CDRresource/description/ @descriptionVocabulary.	MAY be supported by service. MAY be included by consumer in input if supported by service.

The value assigned to {CDRresourceType} as defined in Section 3.1.2.2 - URI Template identifies the Resource Type, as discussed in Section 1.1 - Service Overview and specified in Section 1.7 - CDR Resource Type and CDR Resource Description URIs . The message body MUST be validated against the Resource Type indicated. A fault MUST result if the validation fails.

If a value is provided for @descriptionVocabulary in the M-Create request, the indicated CDR Resource Description Vocabulary MUST be used to validate the description input against the Description Vocabulary indicated. If a Description Vocabulary is not provided, the Manage Service MAY validate against a default vocabulary. A fault MUST result if the validation fails.

3.1.2.5 - M-Create Request Example

Figure 4 shows an example with several { } fields 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. CDR specifications that define profiles for use of Manage functions contain examples with substitutions appropriate to that use.

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

results in

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

Figure 4 : General M-Create Request Example

3.1.3 - Output

3.1.3.1 - HTTP Status Code

If the POST is successful, the service MUST respond with a '201 Created' status code. For requests that result in an error, a HTTP error code MUST be returned. Fault codes are discussed in Section 3.1.5 - Fault Conditions .

3.1.3.2 - HTTP Message Header

- The Header SHOULD include Content-Type
- The Header SHOULD include Content-Encoding

3.1.3.3 - HTTP Message Body

The body of the HTTP response message MUST consist of an identifier for the CDR Resource that was created and MAY include a copy of the resource and/or its description as directed in Table 6 . The CDR Resource Identifier MUST take the form

```
http://{anyAuthority}/{anyHierarchy}/CDRresource/{CDRresourceID}
```

where

{anyAuthority} and {anyHierarchy} are the values as assigned for Section 3.1.2.2 - URI Template .

{CDRresourceID} is the unique part of the identifier assigned by the CDR Resource Collection identified as http://{anyAuthority}/{anyHierarchy}/CDRresource. The value for CDRresourceID is unconstrained in this specification except that it MUST be valid as defined by Uniform Resource Identifier (URI): Generic Syntax[16].

The CDR Resource Identifier is generated for use with the CDR Resource Collection; however, the CDR Resource Identifier MUST be able to support use in other Manage Service functions (M-Read, M-Update, M-Delete, M-Search) and any functions defined by specifications defined as profiles of the Manage Service.

Table 6 defines the XML elements and attributes that are RECOMMENDED [4] to form the M-Create response payload. Unless specifically noted, the remainder of the message body definition that follows will use the <cdrm:MCreateResponse> structure.

[4]The structure that follows is RECOMMENDED and NOT REQUIRED in order to provide a mechanism for compatibility with the <atom:entry> structure defined by version 1.0 of the Query Management specifications. The use of <atom:entry> instead of <cdrm:MCreateResponse> as the root element is elaborated in the Query Management v1.0 specifications.[3]

Table 6 - Elements and Attributes of M-Create Response Message Body

Element/Attribute Name and Description	Support
/cdrm:MResponse An XML element that serves as a wrapper for the response information.	MUST be supported by service.
/cdrm:MResponse/id Identifier for created CDR Resource. The identifier MUST conform to the pattern described above for the CDR Resource Identifier.	MUST be supported by service.
/cdrm:MResponse/{CDRresource} Copy of created resource that corresponds to the input generated per Table 5 .Inclusion of the resource in the output MUST correspond to the output control defined in Table 4 . If output control is not supported by Manage Service implementation or not provided on input, then the service MUST respond using its defined default.	MAY be supported by service.
/cdrm:MResponse/description/{description} Copy of description of created resource that corresponds to the input generated per Table 5 and other description generated by the M-Create implementation. Inclusion of the description in the output MUST correspond to the output control defined in Table 4 . If output control is not supported by Manage Service implementation or not provided on input, then the service MUST respond using its defined default.	MAY be supported by service.

3.1.3.4 - M-Create Response Example

Figure 5 shows an example that corresponds to the M-Create request example shown in Figure 4 . It includes several { } fields for which resource-specific substitutions are needed. CDR specifications that define profiles for use of Manage functions contain examples with substitutions appropriate to that use.

```
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>
   {CDRresource}
   <cdrm:description>
      {description}
   </cdrm:description>
</cdrm:MResponse>
```

Figure 5 : General M-Create Response Example

The example in Figure 5 includes the Location header. This header is OPTIONAL per the HTTP specification.[15] However, if the header is included in the output from the use of the M-Create Function, its value SHOULD be identical to the value of <cdrm:id>. If the values differ, the <cdrm:id> value MUST be used for all Manage functions. Possible uses of the Location header value are beyond the scope of this specification.

3.1.4 - Post-conditions

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

1. The CDR Resource is available for M-Read, M-Update, M-Delete, and M-Search and it is identifiable by the CDR Resource Identifier.

2. The use of this function has been audited according to applicable policy. [5]

3.1.5 - Fault Conditions

Table 7 lists common fault conditions that the M-Create function of a Manage Service implementation SHOULD handle, and Table 7 indicates the HTTP status codes that SHOULD be referenced.

Related specifications, such as profiles using the Manage specification, MAY create additional Fault Conditions, as necessary.

Table 7 - M-Create Fault Conditions and HTTP Responses

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

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

CDR Fault	Fault Description	HTTP Status	HTTP Description
Unsupported Resource Type	The Manage Service does not support the indicated resource type.	400	Bad Request
Unsupported Manage Properties	The Manage Service does not support one or more of the Manage Properties.	400	Bad Request
Unsupported Manage Properties Value	The Manage Service does not support one or more values associated with a Manage Property.	400	Bad Request
Unsupported CDR Resource Syntax	The CDR Resource does not conform to the value for Resource Type.	400	Bad Request
Unsupported Description Vocabulary	The Manage Service does not support the indicated description vocabulary.	400	Bad Request
Unsupported Description Vocabulary Syntax	The description does not conform to the value for Description Vocabulary.	400	Bad Request
Service Execution Fault	The Manage Service encounters an error during execution.	500	Internal Service Error

From Section 2.1 - M-Create : M-Create may also create a CDR Resource Description. The Resource Type, as discussed in Section 1.1 - Service Overview and specified in Section 1.7 - CDR Resource Type and CDR Resource Description URIs , MUST be used to validate the remainder of the input against the Resource Type indicated. If provided in the M-Create request, the CDR Resource Description Vocabulary MUST be used to validate the description input against the Description Vocabulary indicated. If a Description Vocabulary is not provided, the Manage Service MAY validate against a default vocabulary. A fault, as indicated in Table 7 , MUST result if either validation fails.

3.2 - M-Read Function

A Manage Service MUST implement the M-Read Function.

3.2.1 - Preconditions

The following preconditions MUST be satisfied if 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 the M-Read Function implementation.

2. The CDR Resource can be retrieved through reference to its CDR Resource Identifier for the purposes of M-Read.

3.2.2 - Input

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

3.2.2.1 - HTTP Method

The M-Read function MUST use the HTTP GET method.

3.2.2.2 - URI Template

The URI used to access the M-Read function MUST conform to the following: [6]

```
http://{anyAuthority}/{anyHierarchy}/CDRresource/{CDRresourceID}?
{MReadProperties}
```

where

{anyAuthority} and {anyHierarchy} – as defined in Section 3.1.2.2 - URI Template .

{CDRresourceID} – REQUIRED – as defined in Section 3.1.3.3 - HTTP Message Body .

{MReadProperties} – OPTIONAL – Parameters through which the Manage consumer may specify and configure optional behavior supported by the M-Read function implementation.

M-Read properties {MReadProperties} provide a means for both configuration and extensibility. The output parameter as defined in Table 4 MAY be used as an M-Read property. Definition and use of M-Read properties MAY be supported by an implementation; if supported, the properties MAY be included by the consumer in the input. An implementation SHOULD ignore properties it does not support. Additional values enabling more selective output or additional M-Read properties may be defined in future versions of this specification.

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/xml’.

3.2.2.4 - HTTP Message Body

There is no request message body for this function.

[6]For example, http://www.cdr.org/templates/examples/CDRresource/1234?output=all is a conforming URL .

3.2.2.5 - M-Read Request Example

Figure 6 shows an example using the CDR Resource Identifier returned in Figure 5 . Note that the example is independent of the CDR Resource type.

```
http://CDR.org/CDRresource/1234?output=all

results in

GET /CDRresource/1234?output=all HTTP/1.1
Host: CDR.org
```

Figure 6 : General M-Read Request Example

3.2.3 - Output

3.2.3.1 - HTTP Status Code

If the GET is successful, the service MUST respond with a '200 OK' Status Code. For requests that result in an error, a HTTP Error code MUST be returned. Fault codes are discussed in Section 3.2.5 - Fault Conditions .

3.2.3.2 - HTTP Message Header

- The Header SHOULD include Content-Type
- The Header SHOULD include Content-Encoding

3.2.3.3 - HTTP Message Body

The body of the HTTP response message MUST correspond to that defined in Section 3.1.3.3 - HTTP Message Body .

3.2.3.4 - M-Read Response Example

Figure 7 shows an example that corresponds to the M-Read request example shown in Figure 6 . It includes several { } fields for which resource-specific substitutions are needed. CDR specifications that define profiles for use of Manage functions contain examples with substitutions appropriate to that use.

```
HTTP/1.1 200 OK
 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>
   {CDRresource}
   <cdrm:description>
      {description}
   </cdrm:description>
</cdrm:MResponse>
```

Figure 7 : General M-Read Response Example

3.2.4 - Post-conditions

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

1. The CDR Resource is not affected by M-Read.

2. The use of this function has been audited according to applicable policy. [7]

3.2.5 - Fault Conditions

Table 8 lists common fault conditions that the M-Read function of a Manage Service implementation SHOULD handle, and Table 8 indicates the HTTP status codes that SHOULD be referenced.

Related specifications, such as profiles using the Manage specification, MAY create additional Fault Conditions, as necessary.

Table 8 - M-Read Fault Conditions and HTTP Responses

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 Manage Service cannot retrieve a CDR Resource instance corresponding to the supplied identifier.	404	Not Found
Unsupported Manage Properties	The Manage Service does not support one or more of the Manage Properties.	400	Bad Request

[7]The use of this function may be audited according to applicable policy regardless to the success or failure of the function.

CDR Fault	Fault Description	HTTP Status	HTTP Description
Unsupported Manage Properties Value	The Manage Service does not support one or more values associated with a Manage Property.	400	Bad Request
Service Execution Fault	The Manage Service encounters an error during execution.	500	Internal Service Error

3.3 - M-Update Function

A Manage Service MUST implement the M-Update Function.

The M-Update function allows a Consumer Component to change an existing CDR Resource instance. The CDR Resource ID uniquely identifies the CDR Resource instance 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 CDR Resource instance. The CDR Resource ID will remain the same; the CDR Resource Type will remain the same. It MAY be necessary to retrieve the CDR Resource instance prior to performing the update.

3.3.1 - Preconditions

The following preconditions MUST be satisfied if 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 the M-Update Function implementation.

2. The CDR Resource may be accessed through reference to CDR Resource Identifier for purposes of M-Update.

3.3.2 - Input

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

3.3.2.1 - HTTP Method

The M-Update function MUST use the HTTP PUT method.

3.3.2.2 - URI Template

The URI used to access the M-Update function MUST conform to the following: [8]

```
http://{anyAuthority}/{anyHierarchy}/CDRresource/{CDRresourceID}?
{MUpdateProperties}
```

where

[8]For example, http://www.cdr.org/templates/examples/CDRresource?output=all is a conforming URL .

{anyAuthority} and {anyHierarchy} – as defined in Section 3.1.2.2 - URI Template .

{CDRresourceID} – REQUIRED – as defined in Section 3.1.3.3 - HTTP Message Body .

{MUpdateProperties} – OPTIONAL – Parameters through which the Manage consumer may specify and configure optional behavior supported by the M-Update function implementation.

Note that resourceType={CDRresourceType} is not included as a parameter in the URI template because M-Update is intended to revise the content of a CDR Resource and not the type of a resource that was created.

M-Update properties {MUpdateProperties} provide a means for both configuration and extensibility. The output parameter as defined in Table 4 MAY be used as an M-Update property. Definition and use of M-Update properties MAY be supported by an implementation; if supported, the properties MAY be included by the consumer in the input. An implementation SHOULD ignore properties it does not support. Additional values enabling more selective output or additional M-Update properties may be defined in future versions of this specification.

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/xml’.

3.3.2.4 - HTTP Message Body

The body of the M-Update request MUST contain a complete representation of the CDR Resource as defined in Section 3.1.2.4 - HTTP Message Body .

3.3.2.5 - M-Update Request Example

Figure 8 shows an example [9] using the CDR Resource Identifier returned in Figure 5 and with several { } fields 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. CDR specifications that define profiles for use of Manage functions contain examples with substitutions appropriate to that use.

[9]The example assumes the use of <cdrm:CDRresource> as the root element. The use of <atom:entry> instead of <cdrm:CDRresource> as the root element is elaborated in the Query Management specifications.[3] [8]

```
http://CDR.org/CDRresource/1234?output=all
<?xml version="1.0" encoding="UTF-8"?>
<cdrm:CDRresource xmlns:cdrm="...">
   {CDRresource}
   <cdrm:description descriptionVocabulary="{descriptionVocabulary}">
      {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 8 : General M-Update Request Example

3.3.3 - Output

3.3.3.1 - HTTP Status Code

If the PUT is successful, the service MUST respond with a '200 OK' status code. For requests that result in an error, a HTTP error code MUST be returned. Fault codes are discussed in Section 3.3.5 - Fault conditions.

3.3.3.2 - HTTP Message Header

- The Header SHOULD include Content-Type
- The Header SHOULD include Content-Encoding

3.3.3.3 - HTTP Message Body

The body of the HTTP response message MUST correspond to that defined in Section 3.1.3.3 - HTTP Message Body; the content of the response payload (e.g., specific search parameters) MUST correspond to the content of the M-Update request that elicited this response.

3.3.3.4 - M-Update Response Example

Figure 9 shows an example that corresponds to the M-Update example shown in Figure 8. It includes several { } fields for which resource-specific substitutions are needed. CDR specifications that define profiles for use of Manage functions contain examples with substitutions appropriate to that use.

```
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>
   {CDRresource}
   <cdrm:description>
      {description}
   </cdrm:description>
</cdrm:MResponse>
```

Figure 9 : General M-Update Response Example

3.3.4 - Post-conditions

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

1. The CDR Resource reflects specified updates.

2. The CDR Resource is accessible by the CDR Resource Identifier.

3. The use of this function has been audited according to applicable policy. [10]

3.3.5 - Fault conditions

Table 9 lists common fault conditions that the M-Update function of a Manage Service implementation SHOULD handle, and Table 9 indicates the HTTP status codes that SHOULD be referenced.

Related specifications, such as profiles using the Manage specification, MAY create additional Fault Conditions, as necessary.

Table 9 - M-Update Fault Conditions and HTTP Responses

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 Manage Service cannot retrieve a CDR Resource instance corresponding to the supplied identifier.	404	Not Found

[10]The use of this function may be audited according to applicable policy regardless to the success or failure of the function.

CDR Fault	Fault Description	HTTP Status	HTTP Description
Unsupported Manage Properties	The Manage Service does not support one or more of the Manage Properties.	400	Bad Request
Unsupported Manage Properties Value	The Manage Service does not support one or more values associated with a Manage Property.	400	Bad Request
Unsupported CDR Resource Syntax	The CDR Resource does not conform to the value for Resource Type.	400	Bad Request
Unsupported Description Vocabulary	The Manage Service does not support the indicated description vocabulary.	400	Bad Request
Unsupported Description Vocabulary Syntax	The description does not conform to the value for Description Vocabulary.	400	Bad Request
Service Execution Fault	The Manage Service encounters an error during execution.	500	Internal Service Error

As noted in Section 3.1.5 - Fault Conditions with respect to the M-Create function, Section 2.1 - M-Create states that M-Update may also create a CDR Resource Description. The Resource Type, as discussed in Section 1.1 - Service Overview and specified in Section 1.7 - CDR Resource Type and CDR Resource Description URIs , MUST be used to validate the remainder of the input against the Resource Type indicated. If provided in the M-Update request, the CDR Resource Description Vocabulary MUST be used to validate the description input against the Description Vocabulary indicated. If a Description Vocabulary is not provided, the Manage Service MAY validate against a default vocabulary. A fault, as indicated in Table 9 , MUST result if either validation fails.

3.4 - M-Delete Function

A Manage Service MUST implement the M-Delete Function.

The M-Delete function removes a CDR Resource instance and its description from the CDR Resource collection managed by the Manage Component implementation. Section 2.4 - M-Delete includes a discussion of the design considerations related to the M-Delete function.

3.4.1 - Preconditions

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

1. The requester is authenticated and authorized according to applicable policy requirements for the M-Delete Function implementation.

2. The CDR Resource can be accessed through reference to its CDR Resource Identifier for purposes of M-Delete.

3.4.2 - Input

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

3.4.2.1 - HTTP Method

The M-Delete function MUST use the HTTP DELETE method.

3.4.2.2 - URI Template

The URI used to access the M-Delete function MUST conform to the following: [11]

```
http://{anyAuthority}/{anyHierarchy}/CDRresource/{CDRresourceID}?
{MDeleteProperties}
```

where

{anyAuthority} and {anyHierarchy} – as defined in Section 3.1.2.2 - URI Template .

{CDRresourceID} – REQUIRED – as defined in Section 3.1.3.3 - HTTP Message Body .

{MDeleteProperties} – OPTIONAL – Parameters through which the Manage consumer may specify and configure optional behavior supported by the M-Delete function implementation.

M-Delete properties {MDeleteProperties} provide a means for both configuration and extensibility, and specific properties may be defined in future versions of this specification. Definition and use of M-Delete properties MAY be supported by an implementation; if supported, the properties MAY be included by the consumer in the input. An implementation SHOULD ignore properties it does not support. Additional values enabling more selective output or additional M-Delete properties may be defined in future versions of this specification.

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 - M-Delete Request Example

Figure 10 shows an example using the CDR Resource Identifier returned in Figure 5 . Note that the example is independent of the CDR Resource type.

[11] For example, http://www.cdr.org/templates/examples/CDRresource/1234 is a conforming URL .

```
http://CDR.org/CDRresource/1234?delete=logical

results in

DELETE /CDRresource/1234?delete=logical HTTP/1.1
Host: CDR.org
```

Figure 10 : General M-Delete Request Example

The delete=logical parameter is included to illustrate a potential M-Delete property that is consistent with considerations as discussed in Section 2.4 - M-Delete , but definition of this parameter is outside the scope of this specification.

3.4.3 - Output

3.4.3.1 - HTTP Status Code

If the DELETE is successful, the service MUST respond with a '204 OK No Content' Status Code. For requests that result in an error, a HTTP Error code MUST be returned. Fault codes are discussed in Section 3.4.5 - Fault Conditions .

3.4.3.2 - HTTP Message Header

- The Header SHOULD include Content-Type
- The Header SHOULD include Content-Encoding

3.4.3.3 - HTTP Message Body

There is no response message body for this function.

3.4.3.4 - M-Delete Response Example

Figure 11 shows an example that corresponds to the M-Delete request example shown in Figure 10 .

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

Figure 11 : General M-Delete Response Example

3.4.4 - Post-conditions

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

1. The CDR Resource instance is no longer accessible by Manage functions.

2. The use of this function has been audited according to applicable policy. [12]

3.4.5 - Fault Conditions

Table 10 lists common fault conditions that the M-Delete function of a Manage Service implementation SHOULD handle, and Table 10 indicates the HTTP status codes that SHOULD be referenced.

Related specifications, such as profiles using the Manage specification, MAY create additional Fault Conditions, as necessary.

Table 10 - M-Delete Fault Conditions and HTTP Responses

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 Manage Service cannot retrieve a CDR Resource instance corresponding to the supplied identifier.	404	Not Found
Unsupported Manage Properties	The Manage Service does not support one or more of the Manage Properties.	400	Bad Request
Unsupported Manage Properties Value	The Manage Service does not support one or more values associated with a Manage Property.	400	Bad Request
Service Execution Fault	The Manage Service encounters an error during execution.	500	Internal Service Error

3.5 - M-Search Function

A Manage Service SHOULD implement the M-Search Function as defined in this section. This function is RECOMMENDED and NOT REQUIRED because at an early stage (or for a limited implementation) there may not be a sufficient number of CDR Resource instances to merit an associated search capability.

The M-Search function provides the capability of searching the CDR Resource Collection, which is the repository of one or more collections of CDR Resources. This can be viewed as listing all or some subset of the contents of the CDR Resource Collection. M-Search MUST be compliant with CDR Search Interface as specified in CDR-RS [4] or CDR-SS. [9] The search terms will be those appropriate to searching for the CDR Resource(s) of interest, and SHOULD correspond to a description vocabulary as described in the Describe service specifications CDR-SD [5]and CDR-RD. [2]

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

The relevant preconditions, inputs, outputs, post-condition, and faults are as specified in CDR-RS [4] or CDR-SS. [9]

Appendix A Feature Summary

Table 11 summarizes major features by version for RM 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, ICD s and other policy documents have a variety of effective dates.

Table 11 - 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. RM Feature Comparison

Table 12 - RM Feature Comparison

RM Feature Comparison		
Required date	**Feature**	**V1**
	CRUD (Create, Read, Update, Delete)	F
	Search	F

Appendix B Change History

Table 13 summarizes the version identifier history for this DES .

Table 13 - DES Version Identifier History

Doc Revision	Revision Date	Revisions
V1	14 March 2014	Initial Release based on Query Management. For details of changes, see Section B.1 - Changes Based on Query Management

B.1 - Changes Based on Query Management

This section summarizes the significant changes that were made from **IC-DoD Representational State Transfer (REST) Interface Encoding Specification for CDR Query Management, V1.0** to this document. These changes, shown in Table 14 , were primarily made to generalize the Query Management functionality to any identified CDR Resource, as defined, and to reformulate QM as a profile of the more general Manage Component. In addition, changes are introduced to harmonize the content of this specification with that contained in the other specifications produced by the CDR IPT, and to incorporate feedback on the specification from pilot implementations and conformance efforts.

Table 14 - Summary of Changes from QM v1.0

QM v1.0	Manage v1	Rationale for Change
Formulated in terms of managing saved search resource	Formulated in terms of general CDR resource	Other CDR resources identified; additional experience provided basis for generalization
Saved search model introduced	CDR Resource model generalized and revised	Generalize and improve clarity
Scope limited to saved searches; variations among saved search formats included as conditionals	Section added to define CDR resource, identify known resources (including separate saved search constructs), describe specifying new resources	Generalize and create basis for mapping to current and future work
Interfaces defined in terms of /atom:entry	Required use of Atom removed	Consensus that requiring use of Atom was overly restrictive; some Atom use not completely consistent with Atom specification

QM v1.0	Manage v1	Rationale for Change
Saved search instance identified separately by both atom:id and HTTP Location header	CDR resource instance identified by cdrm:id and relationship specified to HTTP Location header	Replace questionable use of atom:id with consistent format not dependent on Atom
URI template limited to saved searches	URI template modified and resource type added to Create function	Generalize for any CDR resource
No Query Management properties defined	output property defined for use across Manage functions	Provide flexibility, including minimum output for use of limited bandwidth
Faults in section on HTTP Status Codes	Added separate fault sections	Consistency with newer specifications.
QM-Search defined with intent to be consistent with CDR Search	M-Search directly references applicable CDR Search specs	Avoid chance for inconsistency
Mapping to Specification Framework done in each function	Mappings to Specification Framework collected in appendix	Consistency with newer specifications

Appendix C Mapping to Specification Framework

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

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 generally substitutes the following:

- Manage (in the context of the Manage Component) for Query Management (in the context of the Query Management as represented by v1.0 of the QM specifications)
- CDR Resource for Saved Search, including such uses as CDR Resource ID for Saved Search ID
- No substitution for the QM-Execute Function because this is a unique addition for Query Management

Table 15 and Table 16 map the CDR-SF inputs and outputs, respectively, to parameters, elements, and attributes defined in this specification. Common parameters are used across the Manage Functions and, as such, the mappings are not repeated for individual functions.

Table 15 - Mapping to CDR Specification Framework Input Variables

Specification Framework Variables	REST Manage Specification
CDR Resource Content	/cdrm:CDRresource/{CDR_Resource}
Resource Type	resourceType
Manage Properties	{MCreateProperties} {MReadProperties} {MUpdateProperties} {MDeleteProperties}
CDR Resource ID	CDR Resource Identifier
Search Function Inputs	as defined for CDR Search and CDR Brokered Search

Table 16 - Mapping to CDR Specification Framework Output Variables

Specification Framework Variables	REST Manage Specification
CDR Resource ID	CDR Resource Identifier
CDR Resource Return	/cdrm:MResponse/{CDRresource}
CDR Resource Description	/cdrm:MResponse/description/{description}
[Delete] Confirmation	(not defined)

Appendix D Glossary

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

CDR	Content Discovery and Retrieval
CDR-RA	Content Discovery & Retrieval - Reference Architecture
CDR-RD	Content Discovery & Retrieval - REST Describe
CDR-RS	Content Discovery & Retrieval - REST Search
CDR-SD	Content Discovery & Retrieval - SOAP Describe
CDR-SF	Content Discovery & Retrieval - Specification Framework
CDR-SS	Content Discovery & Retrieval - SOAP Search
CIO	Chief Information Officer
CVE	Controlled Vocabulary Enumeration
DES	Data Encoding Specification
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
IETF	Internet Engineering Task Force
IPT	Integrated Project Team
ISOO	Information Security Oversight Office
OCIO	Office of the Intelligence Community Chief Information Officer
OS	Open Search
QM	Query Management
REST	Representational State Transfer

RFC	Request for Comments
URI	Uniform Resource Identifier
URL	Uniform Resource Locator
URN	Uniform Resource Name
WWW	World Wide Web
XML	Extensible Markup Language
XPath	XML Path Language
XSL	Extensible Stylesheet Language

Appendix E 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-RD
Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *REST Interface Specification for Content Discovery and Retrieval: Describe (CDR-RD)*.
Available online Intelink-U at: http://purl.org/IC/Standards/CDR-RD
Available online at: http://purl.org/IC/Standards/public

[3] 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

[4] 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

[5] CDR-SD
Intelligence Community/Department of Defense Content Discovery & Retrieval Integrated Project Team. *SOAP Interface Specification for Content Discovery and Retrieval: Describe (CDR-SD)*.
Available online Intelink-U at: http://purl.org/IC/Standards/CDR-SD
Available online at: http://purl.org/IC/Standards/public

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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

[13] 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

[14] 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

[15] IETF-RFC 2616
Internet Engineering Task Force. *Hypertext Transfer Protocol -- HTTP/1.1*. June 1999.
Available online at: http://www.ietf.org/rfc/rfc2616.txt

[16] IETF-RFC 3986

Internet Engineering Task Force. *Uniform Resource Identifier (URI): Generic Syntax*. January 2005.
Available online at: http://tools.ietf.org/html/rfc3986

[17] IETF-RFC 4287
M. Nottingham, R. Sayre. *The Atom Syndication Format*. December 2005.
Available online at: http://www.ietf.org/rfc/rfc4287.txt

[18] 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 F 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 G 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.[13]