

Internet Engineering Task Force
Internet-Draft
Intended status: Informational
Expires: July 11, 2011

K. Sankar, Ed.
Cisco
A. Jones
SNIA
January 7, 2011

Cloud Data Management Interface (CDMI) Media Types draft-cdmi-mediatypes-07

Abstract

This document describes several Internet media types defined for the Cloud Data Management Interface (CDMI) by the Storage Networking Industry Association (SNIA). The media types are:

- o application/cdmi-domain
- o application/cdmi-capability
- o application/cdmi-container
- o application/cdmi-object and
- o application/cdmi-queue

Status of this Memo

This Internet-Draft is submitted in full conformance with the provisions of [BCP 78](#) and [BCP 79](#).

Internet-Drafts are working documents of the Internet Engineering Task Force (IETF). Note that other groups may also distribute working documents as Internet-Drafts. The list of current Internet-Drafts is at <http://datatracker.ietf.org/drafts/current/>.

Internet-Drafts are draft documents valid for a maximum of six months and may be updated, replaced, or obsoleted by other documents at any time. It is inappropriate to use Internet-Drafts as reference material or to cite them other than as "work in progress."

This Internet-Draft will expire on July 11, 2011.

Copyright Notice

Copyright (c) 2011 IETF Trust and the persons identified as the document authors. All rights reserved.

This document is subject to [BCP 78](#) and the IETF Trust's Legal Provisions Relating to IETF Documents (<http://trustee.ietf.org/license-info>) in effect on the date of publication of this document. Please review these documents carefully, as they describe your rights and restrictions with respect to this document. Code Components extracted from this document must include Simplified BSD License text as described in Section 4.e of the Trust Legal Provisions and are provided without warranty as described in the Simplified BSD License.

Table of Contents

1.	Introduction	3
1.1.	Requirements Language	3
2.	Cloud Data Management Domain and its Relevance	3
3.	Processing Guidelines	4
3.1.	Type : application/cdmi-object	4
3.2.	Type : application/cdmi-container	4
3.3.	Type : application/cdmi-domain	5
3.4.	Type : application/cdmi-capability	5
3.5.	Type : application/cdmi-queue	5
4.	Transport Considerations	5
5.	IANA Considerations	6
5.1.	Media Type application/cdmi-capability	6
5.2.	Media Type application/cdmi-container	7
5.3.	Media Type application/cdmi-domain	8
5.4.	Media Type application/cdmi-object	9
5.5.	Media Type application/cdmi-queue	9
6.	Security Considerations	10
6.1.	Confidentiality and Integrity	11
6.2.	Access Control	11
6.3.	Audit	11
6.4.	JSON Security Considerations	11
6.5.	Executable/active content	11
7.	Acknowledgements	11
8.	References	12
8.1.	Normative References	12
8.2.	Informative References	12
	Authors' Addresses	12

1. Introduction

The Cloud Data Management Interface (CDMI) [[CDMI-1](#)] developed by the Storage Networking Industry Association (SNIA) is the functional interface that applications will use to create, retrieve, update and delete data elements from the cloud. As part of this interface, the client will be able to discover the capabilities of the cloud storage offering and use this interface to manage containers and the data that is placed in them. In addition, metadata can be set on containers and their contained data elements through this interface.

1.1. Requirements Language

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 [RFC 2119](#) [[RFC2119](#)].

2. Cloud Data Management Domain and its Relevance

A storage cloud is a storage service hosted either on-premise or off-premise, across a network. An important part of the cloud model, in general, is the concept of a pool of resources that are drawn from, on demand, in small increments (smaller than what one would typically purchase by buying equipment). By abstracting data storage behind a set of service interfaces and delivering it on demand, a wide range of actual offerings and implementations are possible. The only type of storage that is excluded from this definition is that which is delivered, not based on demand, but on fixed capacity increments.

The CDMI defines a set of functional interfaces (data paths) and management interfaces (control paths) to create, retrieve, update, and delete data elements from a storage cloud. Another important concept in this standard is that of metadata. When managing large amounts of data with differing requirements, metadata is a convenient mechanism to express those requirements in such a way that underlying data services can differentiate their treatment of the data to meet those requirements. CDMI also defines an extensible metadata system for storage clouds.

As part of the CDMI interface, the client will be able to discover the capabilities of the cloud storage offering and to use this interface to manage containers and the data that is placed in them. In addition, system metadata can be added to containers and their contained data elements through this interface.

The hierarchy that CDMI defines is as follows:

- o The basic element of storage is an object
- o Objects are stored in a container hierarchy
- o CDMI also defines an object, called a queue, which has special properties for in-order, first in, first-out creation and fetching of queue objects, similar to a container of data objects.
- o A cloud offering can also support domains, which allow administrative ownership to be associated with stored objects. Domains can also be hierarchical, allowing for corporate domains with multiple children domains for departments or individuals. The domain concept is also used to map Access Control Lists (ACLs) to principals as well as to aggregate usage data that is used to bill, meter, and monitor cloud usage. (Note: The CDMI "domain" defined here is not a DNS domain name as specified in RFCs 1023 and 1024.)
- o Finally, a capabilities resource and associated URI [[RFC3986](#)] allows a client to discover the capabilities of the offering and its implementation of CDMI.

3. Processing Guidelines

This section summarizes the processing of each media type. This document provides only the essential information. The CDMI specification [[CDMI-1](#)] which has more details and appropriate examples, is the final authority on the processing aspects.

3.1. Type : application/cdmi-object

A CDMI object is the basic storage element in a CDMI system and is analogous to a file within a filesystem. The object is represented in the CDMI interface in JSON format [[RFC4627](#)] (see the JSON web site at [[JSON-1](#)] for general information about JSON). Each data object has a set of well-defined fields that includes a single value and optional metadata. The implementations are free to store the data in any form they choose, but the application/cdmi-object SHOULD be represented in the CDMI interface as defined in [section 8](#) of the CDMI specification [[CDMI-1](#)].

3.2. Type : application/cdmi-container

A container objects is the the fundamental grouping of stored data within CDMI and is analogous to a directory within a filesystem. Each container has zero or more child objects and a set of well-defined fields that includes standardized and custom metadata. A

container can include other containers similar to sub-directories in a filesystem. The implementations are free to represent the container in any form they choose, but the application/cdmi-container SHOULD be represented in the CDMI interface as defined in [section 9](#) of the CDMI specification [[CDMI-1](#)].

[3.3.](#) Type : application/cdmi-domain

Domain objects represent the concept of administrative ownership of stored data within a CDMI storage system. A CDMI offering may include a hierarchy of domains that provide access to domain-related information within a CDMI context. This domain hierarchy is a series of CDMI objects that correspond to parent and child domains, with each domain corresponding to logical groupings of objects that are to be managed together. [Section 10](#) of the CDMI specification [[CDMI-1](#)] details the information content, representation and processing on domain objects.

[3.4.](#) Type : application/cdmi-capability

Capability objects form a special class of container object that allows a CDMI client to discover what subset of the CDMI standard is implemented by a CDMI provider. For each URI in a CDMI system, the set of interactions that the system is capable of performing against that URI is described by the presence of named "capabilities". Each capability present for a given URI indicates what functionality the cloud storage system will allow against that URI. Capabilities are always static. [Section 12](#) of the CDMI specification [[CDMI-1](#)] details the representation and processing of capability objects.

[3.5.](#) Type : application/cdmi-queue

Queues are a special class of container object and are used to provide first-in, first-out access when storing and retrieving data. A queue writer PUTs objects to the queue, and a queue reader GETs objects from the queue, acknowledging the receipt of the last object that it received. Queuing provides a simple mechanism for one or more writers to send data to a single reader in a reliable way. If queues are supported by the cloud storage system, cloud clients create the queue objects by using the same mechanism used to create data objects. [Section 11](#) of the CDMI specification [[CDMI-1](#)] details the operations and processing of queue objects.

[4.](#) Transport Considerations

The CDMI operates over HTTP [[RFC2616](#)] and does not make sense outside the HTTP realm. We do not expect the CDMI to operate over other

protocols, nor to use a transport protocol, such as TCP ([RFC 793](#)), directly.

5. IANA Considerations

This memo includes a request to IANA to register the following media types :

- o application/cdmi-domain
- o application/cdmi-capability
- o application/cdmi-container
- o application/cdmi-object and
- o application/cdmi-queue

5.1. Media Type application/cdmi-capability

Type name: application

Subtype name: cdmi-capability

Required parameters: none

Optional parameters: none

Encoding considerations: Assumes that the representation is always UTF-8 as defined in [[RFC3629](#)] and 8bit as defined in [[RFC4288](#)].

Security considerations: See Section 6 of [draft-cdmi-mediatypes-07](#)

Interoperability considerations: none

Published specification: [draft-cdmi-mediatypes-07](#)

Applications that use this media type: Implementations of the Cloud Data Management Interface (CDMI) defined by the Storage Networking Industry Association (SNIA)

Additional information:

Magic number(s): n/a

File extension(s): .cdmia

Macintosh file type code(s): TEXT

Person and email address to contact for further information: Arnold Jones
arnold.jones@snia.org

Intended usage: COMMON

Restrictions on usage: none

Author: SNIA Cloud Storage Initiative cloudtwg@snia.org

Change controller: SNIA Cloud Storage Initiative cloudtwg@snia.org

5.2. Media Type application/cdmi-container

Type name: application

Subtype name: cdmi-container

Required parameters: none

Optional parameters: none

Encoding considerations: Assumes that the representation is always UTF-8 as defined in [[RFC3629](#)] and 8bit as defined in [[RFC4288](#)].

Security considerations: See Section 6 of [draft-cdmi-mediatypes-07](#)

Interoperability considerations: none

Published specification: [draft-cdmi-mediatypes-07](#)

Applications that use this media type: Implementations of the Cloud Data Management Interface (CDMI) defined by the Storage Networking Industry Association (SNIA)

Additional information:

Magic number(s): n/a

File extension(s): .cdmic

Macintosh file type code(s): TEXT

Person and email address to contact for further information: Arnold Jones
arnold.jones@snia.org

Intended usage: COMMON

Restrictions on usage: none

Author: SNIA Cloud Storage Initiative cloudtwg@snia.org

Change controller: SNIA Cloud Storage Initiative cloudtwg@snia.org

5.3. Media Type application/cdmi-domain

Type name: application

Subtype name: cdmi-domain

Required parameters: none

Optional parameters: none

Encoding considerations: Assumes that the representation is always UTF-8 as defined in [[RFC3629](#)] and 8bit as defined in [[RFC4288](#)].

Security considerations: See Section 6 of [draft-cdmi-mediatypes-07](#)

Interoperability considerations: none

Published specification: [draft-cdmi-mediatypes-07](#)

Applications that use this media type: Implementations of the Cloud Data Management Interface (CDMI) defined by the Storage Networking Industry Association (SNIA)

Additional information:

Magic number(s): n/a

File extension(s): .cdmid

Macintosh file type code(s): TEXT

Person and email address to contact for further information: Arnold Jones arnold.jones@snia.org

Intended usage: COMMON

Restrictions on usage: none

Author: SNIA Cloud Storage Initiative cloudtwg@snia.org

Change controller: SNIA Cloud Storage Initiative cloudtwg@snia.org

5.4. Media Type application/cdmi-object

Type name: application

Subtype name: cdmi-object

Required parameters: none

Optional parameters: none

Encoding considerations: Assumes that the representation is always UTF-8 as defined in [[RFC3629](#)] and 8bit as defined in [[RFC4288](#)].

Security considerations: See Section 6 of [draft-cdmi-mediatypes-07](#)

Interoperability considerations: none

Published specification: [draft-cdmi-mediatypes-07](#)

Applications that use this media type: Implementations of the Cloud Data Management Interface (CDMI) defined by the Storage Networking Industry Association (SNIA)

Additional information:

Magic number(s): n/a

File extension(s): .cdmio

Macintosh file type code(s): TEXT

Person and email address to contact for further information: Arnold Jones arnold.jones@snia.org

Intended usage: COMMON

Restrictions on usage: none

Author: SNIA Cloud Storage Initiative cloudtwg@snia.org

Change controller: SNIA Cloud Storage Initiative cloudtwg@snia.org

5.5. Media Type application/cdmi-queue

Type name: application

Subtype name: cdmi-queue

Required parameters: none

Optional parameters: none

Encoding considerations: Assumes that the representation is always UTF-8 as defined in [[RFC3629](#)] and 8bit as defined in [[RFC4288](#)].

Security considerations: See Section 6 of [draft-cdmi-mediatypes-07](#)

Interoperability considerations: none

Published specification: [draft-cdmi-mediatypes-07](#)

Applications that use this media type: Implementations of the Cloud Data Management Interface (CDMI) defined by the Storage Networking Industry Association (SNIA)

Additional information:

 Magic number(s): n/a

 File extension(s): .cdmiq

 Macintosh file type code(s): TEXT

Person and email address to contact for further information: Arnold Jones arnold.jones@snia.org

Intended usage: COMMON

Restrictions on usage: none

Author: SNIA Cloud Storage Initiative cloudtwg@snia.org

Change controller: SNIA Cloud Storage Initiative cloudtwg@snia.org

6. Security Considerations

This section was developed with [RFC 3552](#) [[RFC3552](#)] as guide. CDMI is an application interface and the relevant security considerations include confidentiality, integrity, access control and audit. Transport and end point security artifacts like DDoS are orthogonal and domains like non-repudiation are left to the application that employs this interface.

[6.1.](#) Confidentiality and Integrity

The confidentiality and integrity of the CDMI exchanges are determined by the application that uses the interface. CDMI does not contain any specific mechanisms, and relies on transport mechanisms such as TLS (see [https \[RFC2818\]](https://tools.ietf.org/html/rfc2818)) for confidentiality and integrity of the messages across the network.

[6.2.](#) Access Control

The access control of the CDMI end point URLs are beyond this specification. If required, applications should use appropriate URL authentication and authorization techniques, such as URI routers for different classes of users and restrict access based on URI origin.

For fine-grained control of the CDMI objects, the CDMI specification [[CDMI-1](#)] contains the Access Control Lists (ACL) and Access Control Entries (ACE). These are described fully in [section 16.1](#) of the CDMI specification [[CDMI-1](#)].

[6.3.](#) Audit

The CDMI specification [[CDMI-1](#)] has defined a set of metadata fields, as explained in [Section 16.3](#), to facilitate the incorporation of access and other audit markers. The CDMI metadata system is extensible and the implementations can add more metadata as required by the security posture of the domain.

[6.4.](#) JSON Security Considerations

JSON related security considerations, described in [RFC 4627](#) [[RFC4627](#)] apply.

[6.5.](#) Executable/active content

The CDMI interface does not include any directives for active content.

[7.](#) Acknowledgements

The authors wish to acknowledge the guidance and wisdom from Mark Carlson and Peter Saint-Andre, comments from Patrick Faltstrom, and the SNIA CDMI cloud twg for all the insightful discussions and ideas.

[8.](#) References

8.1. Normative References

- [CDMI-1] SNIA, "Cloud data management Interface Version 1.0", 2010, <http://www.snia.org/tech_activities/standards/curr_standards/cdmi>.
- [JSON-1] json.org, "JavaScript Object Notation", 2006, <<http://www.json.org/>>.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", [BCP 14](#), [RFC 2119](#), March 1997.
- [RFC2616] Fielding, R., Gettys, J., Mogul, J., Frystyk, H., Masinter, L., Leach, P., and T. Berners-Lee, "Hypertext Transfer Protocol -- HTTP/1.1", [RFC 2616](#), May 2000.
- [RFC3629] Yergeau, F., "UTF-8, a transformation format of ISO 10646", STD 63, [RFC 3629](#), November 2003.
- [RFC3986] Berners-Lee, T., "Uniform Resource Identifier (URI): Generic Syntax", [RFC 3986](#), January 2005.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", [BCP 13](#), [RFC 4288](#), December 2005.
- [RFC4627] Crockford, D., "The application/json Media Type for JavaScript Object Notation (JSON)", [RFC 4627](#), July 2006.

8.2. Informative References

- [RFC2818] Rescorla, E., "HTTP Over TLS", [RFC 2818](#), May 2000.
- [RFC3552] Rescorla, E. and B. Korver, "Guidelines for Writing RFC Text on Security Considerations", [BCP 72](#), [RFC 3552](#), July 2003.

Authors' Addresses

Krishna Sankar (editor)
Cisco
170 W.Tasman Drive
San Jose, CA 95134
USA

Phone: (408) 853 8475
Email: ksankar@cisco.com

Arnold Jones
SNIA
4410 ArrowsWest Drive
Colorado Springs, CO 80907
USA

Phone: (407) 574 7273
Email: arnold.jones@snia.org