
# Cloud Data Management Interface Profile: Basic Storage Service

## Version 1.0e

"Publication of this Working Draft for review and comment has been approved by the Cloud Storage Technical Working Group. This draft represents a "best effort" attempt by the Cloud Storage Technical Working Group to reach preliminary consensus, and it may be updated, replaced, or made obsolete at any time. This document should not be used as reference material or cited as other than a 'work in progress.' Suggestion for revision should be directed to http:/snia.org/feedback."

***Working Draft***

# Revision History

| Date | Version | By | Comments |
|---|---|---|---|
| 1/20/12 | 1.0a | Mark Carlson | Initial Creation |
| 1/30/13 | 1.0c | Mark Carlson, Oracle | Updated based on feedback during face to face meeting |
| 2/06/13 | 1.0d | Mark Carlson, Oracle | Updated based on feedback in TWG meeting this date |
| 2/08/13 | 1.0e | Marie McMinn | Includes minor edits. |

# Contents

Foreword ..... iv
Introduction ..... v
1 Scope ..... 1
2 Normative References ..... 1
3 Terms ..... 1
4 Conventions ..... 1
5 Basic Storage Service ..... 2
5.1 Overview ..... 2
5.2 Capabilities ..... 2
5.3 Profile Operations ..... 2
5.3.1 Container Operations ..... 3
5.3.2 Data Object Operations ..... 4

## Tables

Table 1 - System-Wide Capabilities ..... 2
Table 2 - Container Capabilities ..... 2

# Foreword

**Abstract**

This document defines a profile of the CDMI interface for easily storing data (data path only) in a cloud storage environment.

**SNIA Website**

Current SNIA practice is to make updates and other information available through its website at http://www.snia.org.

**SNIA Address**

Storage Networking Industry Association, 425 Market Street, Suite #1020, San Francisco, CA 94105, U.S.A.

# Introduction

This document is intended for application developers who are implementing or using cloud storage. It documents how to manage the data stored in a unified storage system.

This document is organized as follows:

| | |
|---|---|
| 1 - Scope | Defines the scope of this document |
| 2 - Normative References | Lists the normative references for this document |
| 3 - Terms | Provides terminology used in this document |
| 4 - Conventions | Describes the conventions used in presenting the interfaces and the typographical conventions used in this document |
| 5 - Basic Storage Service | Provides the normative standard of the profile with examples of typical usage |

# 1 Scope

This profile documents how to store and access the data stored in a cloud storage system. It applies to application developers who are implementing or using cloud storage.

# 2 Normative References

The following documents, in whole or in part, are normatively referenced in this document and are indispensable for its application. For dated references, only the edition cited applies. For undated references, the latest edition of the referenced document (including any amendments) applies.

CDMI, Cloud Data Management Interface

# 3 Terms

See CDMI.

# 4 Conventions

See CDMI.

# 5 Basic Storage Service

## 5.1 Overview

CDMI can be used in situations where the CDMI control path (data management through data system metadata) is minimally supported and the REST/HTTP data path is used to store and retrieve data.

This profile is intended to allow CDMI to be used as a simple data path interface to store and retrieve data for offerings such as this. This profile attempts to document the most basic interface that could be used as a building block in offerings that have more advanced functionality. Any offering can implement more of the CDMI features than are documented here, but this profile is as basic as it gets for using CDMI as a data path.

## 5.2 Capabilities

A CDMI implementation conforming to this profile might implement the system-wide and container capabilities described in Table 1 and Table 2, respectively.

- The mandatory capabilities shall be implemented to conform to this profile.

The system-wide capabilities are listed in Table 1 (see 12.1.1 of CDMI).

**Table 1 - System-Wide Capabilities**

| Capability | Description | Requirement |
|---|---|---|
| cdmi_dataobjects | If present and "true", this capability indicates that the cloud storage system supports the CDMI data path. | Mandatory |

The container capabilities are listed in Table 2 (see Section 12.1.5 of CDMI).

**Table 2 - Container Capabilities**

| Capability | Description | Requirement |
|---|---|---|
| cdmi_create_container | If present and "true", this capability indicates that the cloud storage system supports creating containers in this container. | Mandatory for a root container; optional for children of a root container |
| cdmi_delete_container | If present and "true", this capability indicates that the container can be deleted. | Mandatory for children of a root container |

## 5.3 Profile Operations

Because this profile is intended for storage systems where the CDMI data path is supported, the primary operations revolve around storing, retrieving, and deleting data objects and containers.

### 5.3.1 Container Operations

An implementation that conforms to this profile shall support the container operations that are referenced in the following CDMI subclauses:

- 9.3 Create a Container Object using a Non-CDMI Content Type
- 9.4 Read a Container Object using CDMI Content Type[1]
- 9.8 Delete a Container Object using a Non-CDMI Content Type

#### 5.3.1.1 Listing Top-Level Containers

Some storage systems might only support the creation of top-level containers and restrict certain operations on those containers, such as the ability to create subcontainers. Some implementations may offer the capability of using the top-level container name as part of a virtual-hosted URL, where the container name is added to the beginning of the host name part of the service URL. This practice obviously restricts the namespace for top-level container names in some cases to a single namespace for the entire offering.

CDMI does not allow the "/" character to be used as part of an object name and, instead, envisions that subcontainers are created to organize objects and perhaps allow for different policies and data services to be purchased on this basis. CDMI containers have their own rich set of defined operations as a result.

The following example shows how to list the top-level containers in the system.

EXAMPLE 1 Perform a cdmi-container GET to the service's root URI:

```
GET / HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2

{
   "objectType" : "application/cdmi-container",
   "objectID" : "00007E7F0010CEC234AD9E3EBFE9531D",
   "objectName" : "/",
   "childrenrange" : "0-2",
   "children" : [
       "foo/",
       "bar/",
       "baz/"
   ]
}
```

This example shows that the three top-level containers are named **foo**, **bar,** and **baz**.

[1]The CDMI international standard currently does not define a non-CDMI container read operation.

#### 5.3.1.2 Deleting Containers

A container can be deleted, of course, and in CDMI, you do not need to delete all of the objects in a container beforehand. Deleting a container will recursively delete all subcontainers also. The following example shows a CDMI container DELETE operation.

EXAMPLE 2 Perform a DELETE on the bar container URI:

```
DELETE /bar/ HTTP/1.1
Host: cloud.example.com
```

The following shows the response.

```
HTTP/1.1 204 No Content
```

#### 5.3.1.3 Listing Objects in a Container

To list the objects in a container, just GET the container and select the children.

EXAMPLE 3 Perform a GET to the baz container URI:

```
GET /baz/?children HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2

{
  "children" : [
       "object1.jpg",
       "object2.pdf",
       "object3.txt"
   ]
}
```

This example shows that the container named baz has three data object children: object1.jpg, object2.pdf, and object2.txt.

### 5.3.2 Data Object Operations

An implementation that conforms to this profile shall support the following data object operations and the subclauses of the CDMI international standard where they are detailed:

- 8.3 Create a Data Object using a Non-CDMI Content Type
- 8.5 Read a Data Object using a Non-CDMI Content Type
- 8.7 Update a Data Object using a Non-CDMI Content Type
- 8.9 Delete a Data Object using a Non-CDMI Content Type

#### 5.3.2.1 Uploading a Data Object

Creating an object with CDMI is as simple as sending it to any existing container location with an HTTP PUT:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain;charset=utf-8
Content-Length: 37

This is the Value of this Data Object
```

The following shows the response.

```
HTTP/1.1 201 Created
```

#### 5.3.2.2 Retrieving a Data Object

An object can be retrieved with CDMI by simply using HTTP GET:

```
GET /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: text/plain;charset=utf-8
Content-Length: 37

This is the Value of this Data Object
```

#### 5.3.2.3 Updating a Data Object

Updating an object with CDMI is as simple as sending the new value to the existing object location with an HTTP PUT:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain;charset=utf-8
Content-Length: 41

This is the new Value of this Data Object
```

The following shows the response.

```
HTTP/1.1 204 No Content
```

#### 5.3.2.4 Deleting a Data Object

Deleting an object with CDMI is as simple as sending an HTTP DELETE:

```
DELETE /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
```

The following shows the response.

```
HTTP/1.1 204 No Content
```