SNIA CDMI™ (Cloud Data Management Interface)

Multiple Representations CDMI Extension

Version 0.0.1

ABSTRACT: This CDMI Extension is intended for CDMI 2.0 developers who are considering a standardized way to add multiple representations to CDMI objects. When multiple compatible implementations are demonstrated and approved by the Technical Working Group, this extension will be incorporated into the CDMI standard.

Working Draft

February 6, 2026

USAGE

DISCLAIMER

The information contained in this publication is subject to change without notice. SNIA makes no warranty of any kind with regard to this specification, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. SNIA shall not be liable for errors contained herein or for incidental or consequential damages in connection with the furnishing, performance, or use of this specification.

Suggestions for revisions should be directed to https://www.snia.org/feedback/.

Clause 1

Object Value Representations

1.1 Overview

CDMI objects may contain a single value, which is analgous to the contents of a file. This value is stored in the `value` field of an object. In contrast to the value of a data object, alternate data streams and extended attributes are stored as metadata of the file under the “cdmi_ads” and “cdmi_xattr” metadata containers.

Some data access protocols, such as HTTP, permit the negotiation and selection between multiple representations of the value of a given resource. For example, an image object may have a JPEG representation and a PNG representation.

A CDMI-enabled storage system may optionally implement support for multiple representations. Support for multiple representations is indicated to CDMI-enabled clients by the presence of the `"cdmi_multiple_representations"` system-wide capability.

When this capability is present, CDMI defines two server-generated data system metadata items that allow for discovery of object value representations:

- The `"cdmi_representations"` storage system metadata item allows for the discovery of which representations are supported by the server for the value of the object.
- The `"cdmi_representation_default_provided"` data system metadata item allows for the discovery of which default representation will be used when no specific representations are specified by the client.

Regardless of the presence of the `"cdmi_multiple_representations"` capability, CDMI clients may specify which representation should be the default representation available to clients:

- The `"cdmi_representation_default"` data system metadata item specifies the preferred representation when no specific representations are specified by the client. This data system metadata item shall be preserved by and across CDMI servers.

1.1.1 Multiple Representations Metadata

The `"cdmi_representations"` metadata JSON object contains one or more representation objects. Each representation object has a key identifying the representation, and has a value that is a JSON object containing zero or more CDMI metadata items. For example, if the size of a representation is known, the “cdmi_size” metadata item shall be included in the corresponding representation object.

```
"cdmi_representations": {
  "<RMT>": {
    "metadata_name": "metadata value"
  }
}
```

For each representation object, the key “<RMT>” is set to the representation media type as defined in RFC 6838, and is percent-encoded as defined in RFC 3986.

Custom representation media types should be registered according to the process described in RFC 6838. Custom representation types not registered according to RFC 6838 shall have a type-name defined in the IANA media type

registries, and shall only use the characters a-z, A-Z and 0-9 for the subtype-name and the optional structured syntax suffix.

Each representation object may contain zero or more CDMI metadata items as defined in CDMI 2.0 Clause 16: metadata. When present, representation metadata replaces the metadata with the same name in the "metadata" field when accessing that representation. Representation metadata shall not recursively include the “cdmi_representations” storage system metadata.

Example Representations Metadata

An example with two representations shown below, one with a known size, and one with an ACL:

```
"metadata": {
  "cdmi_representations": {
    "image%2Fjpeg": {
      "cdmi_size": "35253"
    },
    "image%2Fpng": {
      "cdmi_acl": [
        {
         "acetype": "ALLOW",
         "identifier": "OWNER@",
         "aceflags": ""
         "acemask": "ALL_PERMS"
        }
      ]
    }
  }
}
```

1.2 Examples

EXAMPLE 1: Discover which representations and associated metadata are available for the value of a data object named “myImage”

HTTP request:

```
GET /myImage?metadata HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-object

{
  "metadata" : {
    "cdmi_size": "35253",
    "cdmi_representation_default": "image/jpeg",
    "cdmi_representation_default_provided": "image/jpeg",
    "custom_md": "ABC",
    "cdmi_representations": {
      "image%2Fjpeg": {
        "cdmi_size": "35253",
        "custom_md": "ABC"
      },
      "image%2Fpng": {
        "cdmi_size": "63442",
        "custom_md": "DEF"
      }
    }
  }
}
```

EXAMPLE 2: Access the default representation (JPEG) of the data object using HTTP

HTTP request:

```
GET /myImage HTTP/1.1
Host: cloud.example.com
Accept: */*
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: image/jpeg

<Binary JPEG data>
```

EXAMPLE 3: Access a specific discovered representation (PNG) of the data object using HTTP

HTTP request:

```
GET /myImage HTTP/1.1
Host: cloud.example.com
Accept: image/png
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: image/png

<Binary PNG data>
```

EXAMPLE 4: Discover the default representation for the value of a data object

HTTP request:

```
GET /myImage?metadata/cdmi_representation_default HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-object

{
  "metadata" : {
    "cdmi_representation_default": "image/jpeg"
  }
}
```

EXAMPLE 5: Modify the default representation for the value of a data object

HTTP request:

```
PATCH /myImage?metadata/cdmi_representation_default HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-object

{
  "metadata" : {
    "cdmi_representation_default": "image/png"
  }
}
```

EXAMPLE 6: See how metadata changes when the default representation changes:

HTTP request:

```
GET /myImage?metadata HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-object

{
  "metadata" : {
    "cdmi_size": "63442",
    "cdmi_representation_default": "image/png",
    "cdmi_representation_default_provided": "image/png",
    "custom_md": "DEF",
    "cdmi_representations": {
      "image%2Fjpeg" {
        "cdmi_size": "35253",
        "custom_md": "ABC"
      },
      "image%2Fpng" {
        "cdmi_size": "63442",
        "custom_md": "DEF"
      }
    }
  }
}
```

1.3 Multiple representations metadata validation

A CDMI server that supports multiple representations shall validate the server-generated “cdmi_representations” storage system metadata item against the following schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "http://snia.org/cdmi/3.0/cdmi-representations-schema.json#",
  "definitions": {
    "metadata_cdmi_representations_SSMD": {
      "type": "object",
      "properties": {
        "cdmi_representations": {
          "type": "object",
          "patternProperties": {
            "^([a-zA-Z0-9]){1,64}\%2[fF]([a-zA-Z0-9]){1,64}(\+([a-zA-Z0-9]){1,64})?$": {
              "type": "object",
              "properties": {
                "anyOf": [ { "$ref": "#/definitions/metadata_DSMDP" },
                           { "$ref": "#/definitions/metadata_SSMD" },
                           { "$ref": "#/definitions/metadata_UMD" }
                ]
              },
              "additionalProperties": false
            }
          },
          "minProperties": 1,
          "additionalProperties": false
        }
      }
    }
  }
}
```

Additional Validation Restrictions:

- Metadata items within a representations media type JSON object shall not recursively contain a cdmi_representations storage system metadata item.

Clause 2

Instructions to the editor

2.1 Overview

In addition to adding the preceding clause, incorporating this extension into the CDMI 2.0 specification requires the following changes to be made to the specification document.

2.2 Cloud storage system-wide capabilities

Add an entry to the end of the table starting on line 135 of cdmi_advanced/cdmi_capability_object.txt, as follows:

Table 2.1: System-wide capabilities

Capability name	Type	Definition
`cdmi_representations`	JSON string	If present and "`true`", indicates that the CDMI server supports multiple representations.

Add an entry to the end of the table starting on line 204 of cdmi_advanced/cdmi_metadata.txt, as follows:

Table 2.2: Storage system metadata

Metadata name	Type	Description	Requirement

`cdmi_representations`	JSON object	Contains zero or more JSON objects, one for each supported representation for the object, container or queue. The format for the name of each contained representation JSON object shall be an RFC 6838 media type. Each contained representation JSON object in turn contains CDMI storage system metadata, provided data system metadata, and user metadata specific to the representation. This representation metadata overrides the metadata for object, container or queue.
		An empty object indicates that representations exist, but no representations are available to the user.
		Example JSON object structure:
		`/metadata/`
		`/metadata/cdmi_representations/`
		`/metadata/cdmi_representations/image%2Fjpeg/`
		`/metadata/cdmi_representations/image%2Fjpeg/cdmi_size`
		`/metadata/cdmi_representations/image%2Fpng/`
		`/metadata/cdmi_representations/image%2Fpng/cdmi_size`
		Representation JSON objects shall only be returned to a client if the client has `CDMI_ACE_READ_METADATA` access permission for the corresponding representation.
			Optional

Add an entry to the end of the table starting on line 525 of cdmi_advanced/cdmi_metadata.txt, as follows:

Table 2.3: Data system metadata

Metadata name	Type	Description	Requirement
`cdmi_representation_default`	JSON string	If this data system metadata item is present and not an empty string, it indicates that the client is requesting a specific representation be provided when a representation is not specifically requested when using an access protocol that supports multiple representations. The default representation shall be specified as an RFC 6838 media type without percent encoding. • If `cdmi_representation_default` is not present, the default representation provided shall be determined by the CDMI server. • If `cdmi_representation_default` is present, and the requested default representation is both supported by the CDMI server and is authorized for the user, the default representation provided shall be the representation specified in this metadata item. Example JSON string value: `image/jpeg`	Optional

Add an entry to the end of the table starting on line 629 of cdmi_advanced/cdmi_metadata.txt, as follows:

Table 2.4: Provided values of data system metadata

Metadata name	Type	Description	Requirement
`cdmi_representation_default_provided`	JSON string	Indicates the default representation being provided for the object, container or queue. Example JSON string value: `image/jpeg`	Optional