Header-based Metadata Extension

Version 1.1a

"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
2012-11-12	1.0a	TWG	Document created at the Portland Face to Face meeting.
2012-11-14	1.0b	TWG	Edits at San Diego Face-to-Face meeting.
2013-03-20	1.0c	TWG	Edits at Austin Face-to-Face meeting.
2013-11-12	1.0d	TWG	Edits at Memphis Face-to-Face meeting.
2014-01-29	1.0e	TWG	Edits at San Jose SNIA Symposium
2014-05-15	1.1a	TWG	Edits at San Diego Face-to-Face meeting

Header-based Metadata Extension

Overview

The CDMI protocol enables CDMI-aware clients to store and retrieve structured metadata using JSON bodies, but does not permit HTTP-based clients to access this metadata. This extension extends CDMI metadata to permit HTTP header metadata to be stored and retrieved as a subset of CDMI metadata.

Due to limitations associated with HTTP headers, certain restrictions must be placed on metadata that is accessible via headers.

Modifications to the CDMI 1.1 spec:

1) In Clause 6.2.3, add a new row at end of table "Table 6 - Request Headers - Create a Data Object using HTTP"

Header	Type	Description	Requirement
x-*-meta-*	Header String	If the "cdmi_header_metadata" capability is present, for each request header matching the pattern "x-*-meta-*", a new user metadata item shall be created, with the metadata name set to the header field-name, and the metadata value set to the header field-value. If the number of headers, the length of any of the headers, or the total size of the headers exceeds the limits specified in RFC 2616, or specified by the cdmi_header_metadata_maxitems, cdmi_header_metadata_maxsize, or the cdmi_header_metadata_maxtotalsize capabilities, a 400 Bad Request shall be returned to the client.	Conditional

2) In Clause 6.2.8 "Examples", insert at end

EXAMPLE 2 PUT to the container URI the data object name, contents, and metadata:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain;charset=utf-8
X-CDMI-Meta-Colour: Yellow
X-Object-Meta-Shape: Square
Content-Length: 37

This is the Value of this Data Object
```

The following shows the response.

```
HTTP/1.1 201 Created
```

3) After Clause 6.2, add a new clause "6.3 Inspect a Data Object using HTTP"

6.3.1 Synopsis

To check for the presence of a data object, the following request shall be performed:

HEAD <root URI>/<ContainerName>/<DataObjectName>

Where:

- <root URI> is the path to the CDMI cloud.
- <ContainerName> is zero or more intermediate containers that already exist, with one slash (i.e., "/") between each pair of container names.
- <DataObjectName> is the name specified for the data object to be checked.

The object shall also also be able to be checked at <root URI>/cdmi_objectid/<objectID>.

6.3.2 Capabilities

The following capabilities describe the supported operations that may be performed when reading an existing data object:

- Support for the ability to read the metadata of an existing data object is indicated by the presence of the cdmi_read_metadata capability in the specified object.

6.3.3 Request Headers

Request headers may be provided as per RFC 2616.

6.3.4 Request Message Body

A request message body shall not be provided.

6.3.5 Response Headers

The HTTP response headers for checking for the presence of a data object using HTTP are shown in Table 14.

Table 8 - Response Headers - Inspect a Data Object using HTTP

Header	Type	Description	Requirement
Content-Type	Header String	The content type returned shall be the mimetype field in the data object.	Mandatory
Location	Header String	The server shall respond with the URI that the reference redirects to if the object is a reference.	Conditional
x-*-meta-*	Header String	If the "cdmi_header_metadata" capability is present, for each user metadata item in the "metadata" field with a metadata name that is a case-insensitive match to the pattern "x-*-meta-*", a corresponding response header shall be returned to the client where the header field-name shall be the metadata item name, and the header field-value shall be the metadata item value. If a header value to be return is not conformant with RFC 2616, the server may omit the field from the response headers.	Conditional

6.3.6 Response Message Body

No response body shall be provded, as per RFC 2616.

6.3.7 Response Status

Table 9 describes the HTTP status codes that occur when checking the presence of a data object using HTTP.

Table 9 - HTTP Status Codes – Inspect a Data Object using a Non-CDMI Content Type

HTTP Status	Description
200 OK	The data object exists.
302 Found	The URI is a reference to another URI.
400 Bad Request	The request contains invalid parameters or field names.
401 Unauthorized	The authentication credentials are missing or invalid.
403 Forbidden	The client lacks the proper authorization to perform this request.
404 Not Found	The resource was not found at the specified URI, or a requested field within the resource was not found.

6.3.8 Examples

EXAMPLE 1 HEAD to the data object URI to check for the presence of a data object with header metadata:

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

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 37
X-CDMI-Meta-Colour: Yellow
X-Object-Meta-Shape: Square
```

4) In existing Clause 6.3.5, add a new row at end of table "Table 9 - Response Headers - Read a Data Object using HTTP"

Header	Type	Description	Requirement
x-*-meta-*	Header String	If the "cdmi_header_metadata" capability is present, for each user metadata item in the "metadata" field with a metadata name that is a case-insensitive match to the pattern "x-*-meta-*", a corresponding response header shall be returned to the client where the header field-name shall be the metadata item name, and the header field-value shall be the metadata item value. If a header value to be return is not conformant with RFC 2616, the server may omit the field from the response headers.	Conditional

5) In Clause 6.3.8 "Examples", insert at end

EXAMPLE 3 GET to the data object URI to read the value of the data object with header metadata:

```
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
Content-Length: 37
X-CDMI-Meta-Colour: Yellow
X-Object-Meta-Shape: Square

This is the Value of this Data Object
```

6) In existing Clause 6.4.3, add a new row at end of table "Table 1 - Request Headers - Update a Data Object using HTTP"

Header	Type	Description	Requirement
x-*-meta-*	Header String	If the "cdmi_header_metadata" capability is present, for each request header matching the pattern "x-*-meta-*", a new user metadata item shall be created, or an existing metadata item shall be updated, with the metadata name set to the header field-name, and the metadata value set to the header field-value. If a metadata item already exists where the metadata name and the header field-name differ only in case, the existing metadata item value shall be updated. If an empty header field-value is specified, the corresponding metadata item shall be removed from the object. If the number of headers, the length of any of the headers, or the total size of the headers exceeds the limits specified in RFC 2616, or specified by the cdmi_header_metadata_maxitems, cdmi_header_metadata_maxsize, or the cdmi_header_metadata_maxtotalsize capabilities, a 400 Bad Request shall be returned to the client.	Conditional

7) In Clause 6.4.8 "Examples", insert at end

EXAMPLE 3 PUT to the data object URI to update the value and metadata of the data object:

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

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

The following shows the response.

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

8) In Clause 7.2.3, replace with the below text:

The HTTP request headers for creating a CDMI container object using HTTP are shown in Table 15.

Table 15 - Request Headers - Create a Container Object using HTTP

Header	Type	Description	Requirement
x-*-meta-*	Header String	If the "cdmi_header_metadata" capability is present, for each request header matching the pattern "x-*-meta-", a new user metadata item shall be created, with the metadata name set to the header field-name, and the metadata value set to the header field-value. If the number of headers, the length of any of the headers, or the total size of the headers exceeds the limits specified in RFC 2616, or specified by the cdmi_header_metadata_maxitems, cdmi_header_metadata_maxsize, or the cdmi_header_metadata_maxtotalsize capabilities, a 400 Bad Request shall be returned to the client.	Conditional

9) In Clause 7.2.8 "Examples", insert at end

EXAMPLE 2 PUT to the URI the container object name and metadata:

```
PUT /MyContainer/ HTTP/1.1
Host: cloud.example.com
X-CDMI-Meta-Colour: Yellow
```

The following shows the response.

```
HTTP/1.1 201 Created
```

10) After Clause 7.2, add a new clause "7.3 Inspect a Container Object using HTTP"

7.3.1 Synopsis

To check for the presence of a container object, the following request shall be performed:

HEAD <root URI>/<ContainerName>/<TheContainerName>/

Where:

- <root URI> is the path to the CDMI cloud.
- <ContainerName> is zero or more intermediate containers that already exist, with one slash (i.e., "/") between each pair of container names.
- <TheContainerName> is the name specified for the container object to be checked.

The container object shall also also be able to be checked at <root URI>/cdmi_objectid/<objectID>.

7.3.2 Capabilities

The following capabilities describe the supported operations that may be performed when reading an existing container object:

- Support for the ability to read the metadata of an existing container object is indicated by the presence of the cdmi_read_metadata capability in the specified container object.

7.3.3 Request Headers

Request headers may be provided as per RFC 2616.

7.3.4 Request Message Body

A request message body shall not be provided.

7.3.5 Response Headers

The HTTP response headers for checking for the presence of a CDMI container object using HTTP are shown in Table 16.

Table 16 - Response Headers - Inspect a Container Object using HTTP

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdmi-container"	Mandatory
Location	Header String	The server shall respond with the URI that the reference redirects to if the object is a reference.	Conditional
x-*-meta-*	Header String	If the "cdmi_header_metadata" capability is present, for each user metadata item in the "metadata" field with a metadata name that is a case-insensitive match to the pattern "x-*-meta-*", a corresponding response header shall be returned to the client where the header field-name shall be the metadata item name, and the header field-value shall be the metadata item value. If a header value to be return is not conformant with RFC 2616, the server may omit the field from the response headers.	Conditional

7.3.6 Response Message Body

No response body shall be provded, as per RFC 2616.

7.3.7 Response Status

Table 17 describes the HTTP status codes that occur when checking the presence of a container object using HTTP.

Table 17 - HTTP Status Codes – Inspect a Container Object using HTTP

HTTP Status	Description
200 OK	The container object content exists.
302 Found	The URI is a reference to another URI.
400 Bad Request	The request contains invalid parameters or field names.
401 Unauthorized	The authentication credentials are missing or invalid.
403 Forbidden	The client lacks the proper authorization to perform this request.
404 Not Found	The resource was not found at the specified URI, or a requested field within the resource was not found.

7.3.8 Examples

EXAMPLE 1 HEAD to the container object URI to check for the presence of a container object with header metadata:

```
HEAD /MyContainer/ HTTP/1.1
Host: cloud.example.com
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-container
X-CDMI-Meta-Colour: Yellow
```

11) Replace existing Clause "7.3 Read a Container Object using HTTP" with:

7.3.1 Synopsis

Reading a container object using HTTP is not defined by this version of this international standard. Clause 9.3 describes how to read a container object using CDMI.

A server implementation is free to respond to HTTP GETs for Container Objects in any way that conforms with RFC 2616.

If container object metadata items matching the pattern "x-*-meta-*" are present, these metadata items shall be returned as response headers as per clause 9.3 Read a Container Object using CDMI.

12) Replace existing Clause "7.4 Update a Container Object using HTTP" with:

7.4.1 Synopsis

Updating a container object using HTTP is not defined by this version of this international standard. Clause 9.4 describes how to update a container object using CDMI.

A server implementation is free to respond to HTTP PUTs for existing Container Objects in any way that conforms with RFC 2616.

If header metadata matching the pattern "x-*-meta-*" is present in the request headers, these headers shall be processed as per clause 9.4 Update a Container Object using CDMI.

13) In Clause 12.1.1, Add new row at end of table "Table 100 - System-Wide Capabilities"

Capability Name	Type	Description
cdmi_header_metadata	JSON String	If present and "true", this capability indicates that the cloud storage system supports header-visible metadata.
cdmi_header_metadata_maxitems	JSON String	If present, this capability indicates the maximum number of user-defined header metadata items supported per object. If absent, there is no additional limit placed on the number of user-defined metadata items.
cdmi_header_metadata_maxsize	JSON String	If present, this capability indicates the maximum size, in bytes, of each user-defined header metadata item. If absent, there is no additional limit placed on the size of user-defined metadata items.
cdmi_header_metadata_maxtotalsize	JSON String	If present, this capability indicates the maximum size, in bytes, of all user-defined header metadata per object. If absent, there is no additional limit placed on the size of user-defined metadata.

14) In Clause 16.2 "Support for User Metadata", Add to end:

If metadata items with a name is a case-insensitive match to the pattern "x-*-meta-*" are created or updated through a CDMI request, the following conditions shall be true, or else a 400 Bad Request result code shall be returned to the client:

- The metadata name shall be a valid HTTP header field-name
- The metadata value that is a valid HTTP header field-value
- The number of matching headers shall not exceed the limits specified by RFC 2616, and shall not exceed the number specified in the cdmi_header_metadata_maxitems capability.
- The size of each matching header shall not exceed the limits specified by RFC 2616, and shall not exceed the number specified in the cdmi_header_metadata_maxsize capability.
- The total size of all of the matching headers shall not exceed the limits specified by RFC 2616, and shall not exceed the number specified in the cdmi_header_metadata_maxtotalsize capability.