CDMI Extension: JSON Transfer Encoding

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
08-28-2014	1.1a	Marie McMinn, Cloud TWG Editor	New version of JSON Transfer Encoding extension created for the CDMI 1.1.0 revision of the standard. Derived from the 1.0f draft.

JSON Transfer Encoding Extension

Overview

CDMI 1.0.2 supports two transfer encodings, UTF-8 and Base64. UTF-8 works well for text but requires all JSON special characters (such as quotes ") to be escaped. Base64 works well for binary data and does not require any special escaping, but is not human readable.

Many use cases for CDMI, such as for Web 2.0 applications, are based around the storage of JSON data. Currently, this requires either the escaping of JSON data when stored as UTF-8 or the Base64 encoding of the JSON data. This represents an additional overhead and an additional processing pass and makes stored data less human readable.

This extension proposes a third transfer encoding, a "JSON" transfer encoding, that allows stored JSON data in values to be directly specified and accessed as JSON.

For example, assuming that the following JSON is to be stored in a data object:

```
{
   "test" : "value"
}
```

Accessing this with UTF-8 transfer encoding looks like:

```
{
   "valuetransferencoding" : "utf-8",
   "value" : "{\r\t\"test\" : \"value\"}\r}"
}
```

Accessing this with Base64 transfer encoding looks like:

```
{
   "valuetransferencoding" : "base64",
   "value" : "ew0KCSJ0ZXN0IiA6ICJ2YWx1ZSINCn0="
}
```

This extension proposes a third transfer encoding, that would look like:

```
{
   "valuetransferencoding" : "json",
   "value" : { "test" : "value" }
}
```

With this transfer encoding, the value will be a JSON object instead of a string, and the server will be responsible for validating the specified JSON.

Changes to CDMI 1.1.0f:

1) Insert new paragraph at end of 5.15.1 Value Transfer Encoding

CDMI version 1.1 introduces a new "json" value transfer encoding. Data objects with a value transfer encoding of json shall be accessible to CDMI 1.1.0 clients with a value transfer encoding of UTF-8.

2) Insert new bullet at end of valuetransferencoding bullet list in 8.1 Overview

- If the value transfer encoding of the object is set to "json", the data stored in the value of the data object shall contain a valid JSON object and shall be transported as JSON data in the value field. The JSON stored and returned shall be semantically equivalent but may not be syntactically identical. For example, whitespace outside of JSON-quoted strings may be removed or added by either client libraries or by the server. This means that the number of bytes sent may not be the same as the number of bytes stored.

3) Insert new bullet at end of "valuetransferencoding" description entry in Table 21 - Request Message Body - Create a Data Object using CDMI

- "json" indicates that the data object contains a valid JSON object, and the value field shall be a JSON object containing this JSON data. If the contents of the data object value field are set to any value other than a valid JSON object, an HTTP status code of 400 Bad Request shall be returned to the client.

4) Insert new example at end of 8.2.9 Examples

EXAMPLE 5 PUT to the container URI the data object name and JSON contents:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.1

{
   "mimetype" : "text/plain",
   "metadata" : { },
   "valuetransferencoding" : "json"
   "value" : {
        "test" : "value"
   }
}
```

The following shows the response.

```
HTTP/1.1 201 Created
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.1

{
   "objectType": "application/cdmi-object",
   "objectID": "0000706D0010374085EF1A5C7018D774",
```

```
    "objectName": "MyDataObject.txt",
    "parentURI": "/MyContainer/",
    "parentID" : "00007ED90010067404EDED32860C086A",
    "domainURI": "/cdmi_domains/MyDomain/",
    "capabilitiesURI": "/cdmi_capabilities/dataobject/",
    "completionStatus": "Complete",
    "mimetype": "text/plain",
    "metadata": {
        "cdmi_size": "21"
    }
}
```

5) Replace the contents of the "Content-Type" description entry in Table 6 – Request Headers - Create a CDMI Data Object using HTTP

The content type of the data to be stored as a data object. The value specified here shall be used as the mimetype field of the CDMI data object.

- If the content type includes the charset parameter as defined in RFC 2046 of "utf-8" (e.g., ";charset=utf-8"), the valuetransferencoding field of the CDMI data object shall be set to "utf-8".
- If the content type is "application/json", the valuetransferencoding field of the CDMI object shall be set to "json".
- Otherwise, the valuetransferencoding field of the CDMI data object shall be set to "base64".

6) Insert new bullet at end of "valuetransferencoding" description entry in Table 27 - Response Message Body - Read a Data Object using CDMI

- "json" indicates that the data object contains a valid JSON object, and the value field shall be a JSON object containing this JSON data.

7) Insert new bullet after the second bullet in the "value" description entry in Table 27 - Response Message Body - Read a Data Object using CDMI

- If the value transfer encoding field indicates JSON encoding, the value field shall contain a valid JSON object.

8) Insert new example at end of 8.4.8 Examples

EXAMPLE 11 GET to the data object URI to read the value and valuetransferencoding fields of a data object storing JSON data:

```
GET
/cdmi_objectid/0000706D0010374085EF1A5C7018D774?valuetransferencoding;v
alue HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
X-CDMI-Specification-Version: 1.1
```

The following shows the response.

```
HTTP/1.1 200 OK
```

```
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.1

{
   "valuetransferencoding" : "json"
   "value" : { "test" : "value" }
}
```

9) Insert new bullet at end of "valuetransferencoding" description in Table 30 - Request Message Body - Update a CDMI Data Object using CDMI

The value transfer encoding used for the data object value. Three value transfer encodings are defined:

- "utf-8" indicates that the data object contains a valid UTF-8 string and shall be transported as a UTF-8 string in the value field. If the contents of the data object value field are set or updated to any value other than a valid UTF-8 string, an HTTP status code of 400 Bad Request shall be returned to the client.
- "base64" indicates that the data object contains arbitrary binary sequences and shall be transported as a base 64-encoded string in the value field.If the contents of the data object value field are set to any value other than a valid base 64 string, an HTTP status code of 400 Bad Request shall be returned to the client.
- "json" indicates that the data object contains a valid JSON object and shall be transported as a JSON object in the value field. If the contents of the data object value field are set or updated to any value other than a valid JSON object, an HTTP status code of 400 Bad Request shall be returned to the client.

This field shall only be included when updating a data object by value and shall be stored as part of the object. If not specified, the existing value of the valuetransferencoding field shall be used.

When updating or appending to an object with a value transfer encoding of "json" or "utf-8", the client shall use a value transfer encoding of "base64".

This field shall be stored as part of the object.

10) Insert new bullet after the second bullet in the "value" description entry in Table 30 - Request Message Body - Update a CDMI Data Object using CDMI

- If the valuetransferencoding field indicates JSON encoding, the value field shall contain a valid JSON object.

11) Insert new text at end of "Content-Range" description entry in Table 11 - Request Headers - Update a CDMI Data Object using HTTP

When updating by range or appending to an object with a value transfer encoding of "json" or "utf-8", the valuetransferencoding field of the object shall be set to "base64".

12) Insert new bullet at end of "valuetransferencoding" description entry in Table 52 - Request Message Body - Create a New Data Object using CDMI

- "json" indicates that the data object contains a valid JSON object, and the value field shall be a JSON object containing this JSON data. If the contents of the data object value field are set or updated to any value other than a valid JSON object, an HTTP status code of 400 Bad Request shall be returned to the client.

13) Replace the contents of the "Content-Type" description entry in Table 17 - Request Header - Create a New Data Object using HTTP

The content type of the data to be stored as a data object. The value specified here shall be used as the mimetype field of the CDMI data object. If the content type includes the charset parameter as defined in RFC 2046 of "utf-8" (e.g., ";charset=utf-8"), the valuetransferencoding field of the CDMI data object shall be set to "utf-8". If the content type is "application/json", the valuetransferencoding field of the CDMI object shall be set to "json". Otherwise, the valuetransferencoding field of the CDMI data object shall be set to "base64".

14) Insert new bullet at end of valuetransferencoding bullet list in 11.1 Overview

- If the value transfer encoding of the object is set to "json", the data stored in the value of the queue object can contain a valid JSON object, and it shall be transported as JSON data in the value field.

15) Insert new bullet at end of "valuetransferencoding" description entry in Table 87 - Response Message Body - Read a Queue Object using CDMI

- "json" indicates that the queue object value contains a valid JSON object, and it shall be transported as a JSON object in the value field.

16) Insert new bullet after the third bullet in the "value" description entry in Table 87 - Response Message Body - Read a Queue Object using CDMI

- If the valuetransferencoding field indicates JSON encoding, the corresponding value field shall contain a valid JSON object.

17) Insert new bullet at end of "valuetransferencoding" description entry in Table 96 - Request Message Body - Enqueue a New Queue Object Value using CDMI

- "json" indicates that the queue object value contains a valid JSON object and shall be transported as a JSON object containing this JSON data. If the contents of the queue object value field are set to any value other than a valid JSON object, an HTTP status code of 400 Bad Request shall be returned to the client.

18) Update Example 5 in clause "11.6.8 Examples"

```
"mimetype": [
   "text/plain",
   "text/plain",
   "application/json"
],
```

```
"valuetransferencoding": [
    "utf-8",
    "base64",
    "json"
],
"value": [
    "First",
    "U2Vjb25k",
    {
        "value" : "test"
    }
]
```

19) Insert at end of Table 100 - System-Wide Capabilities

Capability Name	Type	Description
cdmi_valuetransferencoding_json	JSON String	If present and "true", this capability indicates that the cloud storage system supports JSON value transfer encoding.