



Cloud Data Management Interface Extension: Multi-part MIME Transfers

Version 1.0g

"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
2011-12-06	1.0a	David Slik	Initial Creation
2011-12-14	1.0b	Kevin Jamieson	Added sections for data object update and queue read and enqueue.
2012-01-13	1.0c	Kevin Jamieson	Added sections for deserializevalue in data and queue object create and update. Various other minor clarifications.
2012-01-18	1.0d	Marie McMinn	Performed technical edit.
2012-01-19	1.0e	Kevin Jamieson	Added text for update and retrieval of multiple byte ranges of a data object.
2012-01-20	1.0f	Kevin Jamieson	Changed deserializevalue to allow the serialized object to be transmitted in the second MIME part.
2012-01-23	1.0g	Kevin Jamieson	Updates from CDMI TWG review.

The SNIA hereby grants permission for individuals to use this document for personal use only, and for corporations and other business entities to use this document for internal use only (including internal copying, distribution, and display) provided that:

Any text, diagram, chart, table, or definition reproduced shall be reproduced in its entirety with no alteration, and,

Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced shall acknowledge the SNIA copyright on that material, and shall credit the SNIA for granting permission for its reuse.

Other than as explicitly provided above, you may not make any commercial use of this document, sell any excerpt or this entire document, or distribute this document to third parties. All rights not explicitly granted are expressly reserved to SNIA.

Permission to use this document for purposes other than those enumerated above may be requested by e-mailing tcmd@snia.org. Please include the identity of the requesting individual and/or company and a brief description of the purpose, nature, and scope of the requested use.

Copyright © 2012 Storage Networking Industry Association.

Multi-part MIME Transfer Extension

Overview

CDMI provides three methods by which the value of a data object may be transferred between CDMI clients and servers:

- UTF-8 encoding in JSON using a CDMI content type request/response,
- Base64 encoding in JSON using a CDMI content type request/response, and
- raw binary using a non-CDMI content type request/response.

UTF-8 encoding is sufficient for most text use cases, and using raw binary transfer provided by the non-CDMI PUT and GET operations is sufficient for some binary use cases. However, there is a need to be able to efficiently transfer binary data alongside CDMI object metadata without incurring the overhead of the UTF-8 or Base64 encoding and validation required to represent binary data in JSON.

This proposed extension adds the ability to use a multi-part MIME body with CDMI to allow the value to be included as raw binary data in a separate MIME part of a single CDMI content type request/response that does not require any encoding or validation of the data.

Modifications to the CDMI 1.0.1 spec:

1) Insert into Clause "2 Normative References"

RFC 2046, *Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types* - <http://www.ietf.org/rfc/rfc2046.txt>

2) Append to end of Clause "8.1 Overview"

The value of a data object may also be specified and retrieved using multi-part MIME, where the CDMI JSON is transferred in the first MIME part, and the raw object value is transferred in the second MIME part. Each MIME part, including any header fields, shall conform to RFC 2045, RFC 2046, and RFC 2616. The length of each part may optionally be specified by a Content-Length header in addition to the MIME boundary delimiter.

Multiple non-overlapping ranges of the value of a data object may also be accessed or updated in a multi-part MIME operation by transferring one MIME part for each range of the value. The byte ranges for these operations shall be specified as per Section 14.35.1 of RFC 2616.

Multi-part MIME enables the efficient transfer of binary data alongside CDMI object metadata without incurring the overhead of the UTF-8 or Base64 encoding and validation required to represent binary data in JSON.

3) Append to end of Clause "8.2.3 Capabilities"

- Support for the ability to create the value of a new data object in specified byte ranges is indicated by the presence of the "cdmi_create_value_range" capability in the parent container.
- Support for the ability to create a new data object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

4) **Modify Clause "8.2.4 Request Headers", Table "Table 5 - Request Headers for Creating a CDMI Data Object using CDMI Content Type"**

Header	Type	Description	Requirement
Content-Type	Header String	<p>"application/cdm-object" or "multipart/mixed".</p> <p>If "multipart/mixed" and the "deserializevalue" field is not specified, the body shall consist of at least two MIME parts, where the first part shall contain a body of content-type "application/cdm-object", and the second and subsequent parts shall contain one or more byte ranges of the value as described in 8.3. If multiple byte ranges are included and the "Content-Range" header is omitted for a part, the data in the part shall be appended to the data in the preceding part, with the first part having a byte offset of zero.</p> <p>If "multipart/mixed" and the "deserializevalue" field is specified with the value of the MIME "boundary" parameter, the body shall consist of two or three MIME parts, where the first part shall contain a body of content-type "application/cdm-object", the second part shall contain the serialized data object, and the third part shall optionally contain the value as described in 8.3.</p>	Mandatory

5) **Modify Clause "8.2.5 Request Message Body", Table "Table 6 - Request Message Body - Create a Data Object using CDMI Content Type"**

Field Name	Type	Description	Requirement
mimetype	JSON String	<p>MIME type of the data contained within the value field of the data object</p> <ul style="list-style-type: none"> • This field may be included when creating by value or when deserializing, serializing, copying, and moving a data object. • If this field is not included and multi-part MIME is not being used, the value of "text/plain" shall be assigned as the field value. • If this field is not included and multi-part MIME is being used, the value of the "Content-Type" header of the second MIME part shall be assigned as the field value. • This field shall be stored as part of the object. • This mimetype value shall be converted to lowercase before being stored. • This field shall not be included when creating a reference. 	Optional

Field Name	Type	Description	Requirement
deserializevalue	JSON String	<p>A data object serialized as specified in Clause 15 and encoded using base 64 encoding rules described in RFC 4648.</p> <p>If multi-part MIME is being used and this field contains the value of the MIME “boundary” parameter, the contents of the second MIME part shall be assigned as the field value. If the serialized data object in the second MIME part does not include a “value” field the contents of the third MIME part shall be assigned as the field value of the “value” field.</p>	Optional ^a
valuetransferencoding	JSON String	<p>The value transfer encoding used for the data object value. Two value transfer encodings are defined.</p> <ul style="list-style-type: none"> • "utf-8" indicates that the data object contains a valid UTF-8 string, and it shall be transported as a UTF-8 string in the value field. • "base64" indicates that the data object may contain arbitrary binary sequences, and it shall be transported as a base 64-encoded string in the value field. Setting the contents of the data object value field to any value other than a valid base 64 string shall result in error 400 Bad Request being returned to the client. <p>This field shall only be included when creating a data object by value.</p> <ul style="list-style-type: none"> • If this field is not included and multi-part MIME is not being used, the value of "utf-8" shall be assigned as the field value. • If this field is not included and multi-part MIME is being used, the value of "utf-8" shall be assigned as the field value if the “Content-Type” header of the second and all subsequent MIME parts includes the charset parameter as defined in RFC 2046 of "utf-8" (e.g., ";charset=utf-8"). Otherwise, the value of "base64" shall be assigned as the field value. This field applies only to the encoding of the value when represented in JSON; the “Content-Transfer-Encoding” header of the part specifies the encoding of the value within a multi-part MIME request, as defined in RFC 2045. <p>This field shall be stored as part of the object.</p>	Optional

Field Name	Type	Description	Requirement
value	JSON String	<p>The data object value.</p> <ul style="list-style-type: none"> If this field is not included and multi-part MIME is not being used, an empty JSON String (i.e., "") shall be assigned as the field value. If this field is not included and multi-part MIME is being used, the contents of the second MIME part shall be assigned as the field value. If the <code>valuetransferencoding</code> field indicates UTF-8 encoding, the value shall be a UTF-8 string escaped using the JSON escaping rules described in RFC 4627. If the <code>valuetransferencoding</code> field indicates base64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648. 	Optional ^a

6) Append to end of Clause "8.2.9 Examples"

EXAMPLE 3 - PUT to the container URI the data object name and binary contents using multi-part MIME:

```

PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdm-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdm-object

{
  "domainURI": "/cdmi_domains/MyDomain/",
  "metadata": {
    "colour": "blue"
  }
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response.

```

HTTP/1.1 201 Created
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.1

```

```
{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID": "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "application/octet-stream",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  }
}
```

EXAMPLE 4 - PUT to the container URI the data object name and binary contents using multi-part MIME with optional content-lengths for the parts:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
Content-Length: 82

{
  "domainURI": "/cdmi_domains/MyDomain/",
  "metadata": {
    "colour": "blue"
  }
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-Length: 37

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

```
HTTP/1.1 201 Created
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.1

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID": "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
```

```

    "capabilitiesURI": "/cdmi_capabilities/dataobject/",
    "completionStatus": "Complete",
    "mimetype": "application/octet-stream",
    "metadata": {
      "cdmi_size": "37",
      "colour": "blue"
    }
  }
}

```

EXAMPLE 5 - PUT to the container URI a serialized data object using multi-part MIME:

```

PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "deserializevalue" : "gc0p4Jq0M2Yt08j34c0p"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID" : "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  },
  "valuerange" : "0-36",
  "valuetransferencoding" : "utf-8",
  "value" : "This is the Value of this Data Object"
}

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response.

```

HTTP/1.1 201 Created
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.1

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",

```

```

"parentURI": "/MyContainer/",
"parentID" : "00007E7F00102E230ED82694DAA975D2",
"domainURI": "/cdmi_domains/MyDomain/",
"capabilitiesURI": "/cdmi_capabilities/dataobject/",
"completionStatus": "Complete",
"mimetype": "text/plain",
"metadata": {
  "cdmi_size": "37",
  "colour": "blue"
}
}

```

EXAMPLE 6 - PUT to the container URI a serialized data object and binary contents using multi-part MIME:

```

PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "deserializevalue" : "gc0p4Jq0M2Yt08j34c0p"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID" : "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "application/octet-stream",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  },
  "valuerange" : "0-36",
  "valuetransferencoding" : "base64"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response.

```

HTTP/1.1 201 Created
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.1

{
  "objectType": "application/cdm-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID": "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "application/octet-stream",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  }
}

```

7) Append to end of Clause "8.3.2 Capabilities"

- Support for the ability to create the value of a new data object in specified byte ranges is indicated by the presence of the "cdmi_create_value_range" capability in the parent container.

8) Modify Clause "8.3.3 Request Headers", Table "Table 10 - Request Headers - Create a CDMI Data Object using a Non-CDMI Content Type"

Header	Type	Description	Requirement
Content-Range	Header String	A valid ranges-specifier (see RFC 2616 Section 14.35.1)	Optional

9) Append to end of Clause "8.4.2 Capabilities"

- Support for the ability to read a data object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

10) Modify Clause "8.4.3 Request Headers", Table "Table 12 - Request Headers - Read a CDMI Data Object using CDMI Content Type"

Header	Type	Description	Requirement
Accept	Header String	"application/cdm-object" or "multipart/mixed".	Mandatory

11) Modify Clause "8.4.5 Response Headers", Table "Table 13 - Response Headers - Read a CDMI Data Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdm-object" or "multipart/mixed". If "multipart/mixed", the body shall consist of at least two MIME parts, where the first part shall contain a body of content-type "application/cdm-object" and the second and subsequent parts shall contain the requested byte ranges of the value as described in 8.5. If multiple byte ranges are included and the "Content-Range" header is omitted for a part, the data in the part shall be appended to the data in the preceding part, with the first part having a byte offset of zero.	Mandatory

12) Modify Clause "8.4.6 Response Message Body", Table "Table 14 - Response Message Body - Read a Data Object using CDMI Content Type"

Field Name	Type	Description	Requirement
value	JSON String	The data object value. <ul style="list-style-type: none"> • If the valuetransferencoding field indicates UTF-8 encoding, the value field shall contain a UTF-8 string using JSON escaping rules described in RFC 4627. • If the valuetransferencoding field indicates base64 encoding, the value field shall contain a base 64-encoded string as described in RFC 4648. • The value field shall not be provided when using multi-part MIME. • The value field shall only be provided when the completionStatus field contains "Complete". • When reading a value, zeros shall be returned for any gaps resulting from non-contiguous writes. 	Conditional

13) Append to end of Clause "8.4.8 Examples"

EXAMPLE 5 - GET to the data object URI to read the data object using multi-part MIME:

```
GET /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: multipart/mixed
X-CDMI-Specification-Version: 1.0.1
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID": "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "application/octet-stream",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  },
  "valuerange": "0-36",
  "valuetransferencoding": "base64"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--
```

EXAMPLE 6 - GET to the data object URI to read the data object using multi-part MIME, with optional content-lengths for the parts:

```
GET /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: multipart/mixed
X-CDMI-Specification-Version: 1.0.1
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
Content-Length: 505
```

```
{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID": "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "application/octet-stream",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  },
  "valuerange": "0-36",
  "valuetransferencoding": "base64"
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-Length: 37
```

<37 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p-
```

EXAMPLE 7 - GET to the data object URI to read the metadata and multiple byte ranges of the binary contents using multi-part MIME:

```
GET /MyContainer/MyDataObject.txt?metadata;value:0-10;value:21-24 HTTP/1.1
Host: cloud.example.com
Accept: multipart/mixed
X-CDMI-Specification-Version: 1.0.1
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
```

```
{
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  }
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-Range: bytes 0-10/37
```

<11 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
Content-Range: bytes 21-24/37

<4 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--
```

14) Append to end of Clause "8.6.2 Capabilities"

- Support for the ability to modify an existing data object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

15) Modify Clause "8.6.3 Request Headers", Table "Table 19 - Request Headers - Update a CDMI Data Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	<p>"application/cdmi-object" or "multipart/mixed".</p> <p>If "multipart/mixed" and the "deserializevalue" field is not specified, the body shall consist of at least two MIME parts, where the first part shall contain a body of content-type "application/cdmi-object" and the second and subsequent parts shall contain one or more byte ranges of the value as described in 8.7. If multiple byte ranges are included and the "Content-Range" header is omitted for a part, the data in the part shall be appended to the data in the preceding part, with the first part having a byte offset of zero.</p> <p>If "multipart/mixed" and the "deserializevalue" field is specified with the value of the MIME "boundary" parameter, the body shall consist of two or three MIME parts, where the first part shall contain a body of content-type "application/cdmi-object", the second part shall contain the serialized data object, and the third part shall optionally contain the value as described in 8.7.</p>	Mandatory

16) Modify Clause "8.6.4 Request Message Body", Table "Table 20 - Request Message Body - Update a CDMI Data Object using CDMI Content Type"

Field Name	Type	Description	Requirement
mimetype	JSON String	<p>MIME type of the data contained within the value field of the data object. If present, this replaces the existing mimetype.</p> <ul style="list-style-type: none"> • This field may be included when updating by value, deserializing, and copying a data object. • If this field is not included, the existing value of the mimetype shall be left unchanged. • This field shall be stored as part of the object. • This mimetype value shall be converted to lowercase before being stored. 	Optional
deserializevalue	JSON String	<p>A data object serialized as specified in Clause 15 and encoded using base 64 encoding rules described in RFC 4648. The object ID of the serialized data object shall match the object ID of the destination data object.</p> <p>If multi-part MIME is being used and this field contains the value of the MIME "boundary" parameter, the contents of the second MIME part shall be assigned as the field value. If the serialized data object in the second MIME part does not include a "value" field the contents of the third MIME part shall be assigned as the field value of the "value" field.</p>	Optional ^a

Field Name	Type	Description	Requirement
valuetransferencoding	JSON String	<p>The value transfer encoding used for the data object value. Two value transfer encodings are defined.</p> <ul style="list-style-type: none"> • "utf-8" indicates that the data object contains a valid UTF-8 string, and it shall be transported as a UTF-8 string in the value field. • "base64" indicates that the data object may contain arbitrary binary sequences, and it shall be transported as a base 64-encoded string in the value field. Setting the contents of the data object value field to any value other than a valid base 64 string shall result in error 400 Bad Request being returned to the client. <p>This field shall only be included when updating a data object by value.</p> <ul style="list-style-type: none"> • If this field is not included and multi-part MIME is not being used, the existing value of "valuetransferencoding" shall be left unchanged. • If this field is not included and multi-part MIME is being used, the value of "utf-8" shall be assigned as the field value if the "Content-Type" header of the second and all subsequent MIME parts includes the charset parameter as defined in RFC 2046 of "utf-8" (e.g., ";charset=utf-8"). Otherwise, the value of "base64" shall be assigned as the field value. This field applies only to the encoding of the value when represented in JSON; the "Content-Transfer-Encoding" header of the part specifies the encoding of the value within a multi-part MIME request, as defined in RFC 2045. <p>This field shall be stored as part of the object.</p>	Optional

Field Name	Type	Description	Requirement
value	JSON String	<p>This field contains the new data for the object. If present, this replaces the existing value.</p> <ul style="list-style-type: none"> • If this field is not included and multi-part MIME is being used, the contents of the second and subsequent MIME parts shall be assigned to the corresponding byte ranges of the field value. • If the <code>valuetransferencoding</code> field indicates UTF-8 encoding, the value shall be a UTF-8 string escaped using the JSON escaping rules described in RFC 4627. • If the <code>valuetransferencoding</code> field indicates base64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648. • If a value range was specified in the request, the new data shall be inserted at the location specified by the range. Any resulting gaps between ranges shall be treated as if zeros had been written and shall be included when calculating the size of the value. When storing a range, the value shall be encoded using base 64, and the <code>valuetransferencoding</code> field shall be set to "base64". 	Optional ^a

17) Append to end of Clause "8.6.8 Examples"

EXAMPLE 7 - PUT to the data object URI to set new field values and the binary contents using multi-part MIME:

```

PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "metadata": {
    "colour": "red",
    "number": "7"
  }
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

```

```
--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

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

EXAMPLE 8 - PUT to the data object URI to replace just one metadata item and update multiple byte ranges within the binary contents of the data object using multi-part MIME:

```
PUT /MyContainer/BinaryObject.txt?metadata:colour HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
```

```
{
  "metadata": {
    "colour": "green"
  }
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Range: bytes 0-10/37
```

<11 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Range: bytes 21-24/37
```

<4 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

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

EXAMPLE 9 - PUT to the data object URI a serialized data object using multi-part MIME:

```
PUT /MyContainer/BinaryObject.txt HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
```

```
{
  "deserializevalue": "gc0p4Jq0M2Yt08j34c0p"
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object
```

```

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "parentID": "00007E7F00102E230ED82694DAA975D2",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  },
  "valuerange": "0-36",
  "valuetransferencoding": "utf-8",
  "value": "This is the Value of this Data Object"
}

```

```
--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

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

18) Append to end of Clause "9.9.3 Capabilities"

- If the new data object is being created in "/cdmi_objectid/", support for the ability to create the value of the new data object in specified byte ranges is indicated by the presence of the "cdmi_create_value_range_by_ID" system capability.
- If the new data object is being created in a container object, support for the ability to create the value of the new data object in specified byte ranges is indicated by the presence of the "cdmi_create_value_range" capability in the parent container.
- Support for the ability to create a new data object by ID using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

19) Modify Clause "9.9.4 Request Headers", Table "Table 49 - Request Headers - Create a New Data Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	<p>"application/cdmi-object" or "multipart/mixed".</p> <p>If "multipart/mixed" and the "deserializevalue" field is not specified, the body shall consist of at least two MIME parts, where the first part shall contain a body of content-type "application/cdmi-object" and the second and subsequent parts shall contain one or more byte ranges of the value as described in 8.3. If multiple byte ranges are included and the "Content-Range" header is omitted for a part, the data in the part shall be appended to the data in the preceding part, with the first part having a byte offset of zero.</p> <p>If "multipart/mixed" and the "deserializevalue" field is specified with the value of the MIME "boundary" parameter, the body shall consist of two or three MIME parts, where the first part shall contain a body of content-type "application/cdmi-object", the second part shall contain the serialized data object, and the third part shall optionally contain the value as described in 8.3.</p>	Mandatory

20) Modify Clause "9.9.5 Request Message Body", Table "Table 50 - Request Message Body - Create a New Data Object using CDMI Content Type"

Field Name	Type	Description	Requirement
mimetype	JSON String	<p>MIME type of the data contained within the value field of the data object</p> <ul style="list-style-type: none"> • This field may be included when creating by value or when deserializing, serializing, copying, or moving a data object. • If this field is not included and multi-part MIME is not being used, the value of "text/plain" shall be assigned as the field value. • If this field is not included and multi-part MIME is being used, the value of the "Content-Type" header of the second MIME part shall be assigned as the field value. • This field shall be stored as part of the object. • This mimetype value shall be converted to lowercase before being stored. • This field shall not be included when creating a reference. 	Optional

Field Name	Type	Description	Requirement
deserializevalue	JSON String	<p>A data object serialized as specified in Clause 15 and encoded using base 64 encoding rules described in RFC 4648.</p> <p>If multi-part MIME is being used and this field contains the value of the MIME “boundary” parameter, the contents of the second MIME part shall be assigned as the field value. If the serialized data object in the second MIME part does not include a “value” field the contents of the third MIME part shall be assigned as the field value of the “value” field.</p>	Optional ^a
valuetransferencoding	JSON String	<p>The value transfer encoding used for the data object value. Two value transfer encodings are defined.</p> <ul style="list-style-type: none"> • "utf-8" indicates that the data object contains a valid UTF-8 string, and it shall be transported as a UTF-8 string in the value field. • "base64" indicates that the data object may contain arbitrary binary sequences, and it shall be transported as a base 64-encoded string in the value field. Setting the contents of the data object value field to any value other than a valid base 64 string shall result in error 400 Bad Request being returned to the client. <p>This field shall only be included when creating a data object by value.</p> <ul style="list-style-type: none"> • If this field is not included and multi-part MIME is not being used, the value of "utf-8" shall be assigned as the field value. • If this field is not included and multi-part MIME is being used, the value of "utf-8" shall be assigned as the field value if the “Content-Type” header of the second and all subsequent MIME parts includes the charset parameter as defined in RFC 2046 of "utf-8" (e.g., ";charset=utf-8"). Otherwise, the value of "base64" shall be assigned as the field value. This field applies only to the encoding of the value when represented in JSON; the “Content-Transfer-Encoding” header of the part specifies the encoding of the value within a multi-part MIME request, as defined in RFC 2045. <p>This field shall be stored as part of the object.</p>	Optional

Field Name	Type	Description	Requirement
value	JSON String	<p>The data object value</p> <ul style="list-style-type: none"> If this field is not included and multi-part MIME is not being used, an empty JSON String (i.e., "") shall be assigned as the field value. If this field is not included and multi-part MIME is being used, the contents of the second MIME part shall be assigned as the field value. If the <code>valuetransferencoding</code> field indicates UTF-8 encoding, the value shall be a UTF-8 string escaped using the JSON escaping rules described in RFC 4627. If the <code>valuetransferencoding</code> field indicates base64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648. 	Optional ^a

21) Append to end of Clause "9.9.9 Examples"

EXAMPLE 3 - POST to the object ID URI the data object fields and binary contents using multi-part MIME:

```

POST /cdmi_objectid/ HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "domainURI": "/cdmi_domains/MyDomain/",
  "metadata": {
    "colour": "blue"
  }
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response.

```

HTTP/1.1 201 Created
Location:
http://cloud.example.com/cdmi_objectid/0000706D0010B84FAD185C425D8B537E
Content-Type: application/cdmi-object

```

```
X-CDMI-Specification-Version: 1.0.1

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "application/octet-stream",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  }
}
```

EXAMPLE 4 - POST to the object ID URI a serialized data object using multi-part MIME:

```
POST /cdmi_objectid/ HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "deserializevalue": "gc0p4Jq0M2Yt08j34c0p"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-object

{
  "objectType": "application/cdmi-object",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  },
  "valuerange" : "0-36",
  "valuetransferencoding" : "utf-8",
  "value" : "This is the Value of this Data Object"
}

--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

```
HTTP/1.1 201 Created
Location:
http://cloud.example.com/cdmi_objectid/0000706D0010B84FAD185C425D8B537E
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.1
```

```

{
  "objectType": "application/cdm-object",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37",
    "colour": "blue"
  }
}

```

22) Append to end of Clause "9.10.2 Capabilities"

- If the new data object is being created in "/cdmi_objectid/", support for the ability to create the value of the new data object in specified byte ranges is indicated by the presence of the "cdmi_create_value_range_by_ID" system capability.
- If the new data object is being created in a container object, support for the ability to create the value of the new data object in specified byte ranges is indicated by the presence of the "cdmi_create_value_range" capability in the parent container.

23) Modify Clause "9.10.3 Request Headers", Table "Table 54 - Request Headers - Create a New Data Object using a Non-CDMI Content Type"

Header	Type	Description	Requirement
Content-Range	Header String	A valid ranges-specifier (see RFC 2616 Section 14.35.1)	Optional

24) Append to end of Clause "9.11.3 Capabilities"

- Support for the ability to create a queue object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

25) Modify Clause "9.11.4 Request Headers", Table "Table 57 - Request Headers - Create a New Queue Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdmi-queue" or "multipart/mixed". If "multipart/mixed" and the "deserializevalue" field is specified with the value of the MIME "boundary" parameter, the body shall consist of two or more MIME parts, where the first part shall contain a body of content-type "application/cdmi-queue", the second part shall contain the serialized queue object, and optionally the third and subsequent parts shall each contain a queue value as described in 8.5.	Mandatory

26) Modify Clause "9.11.5 Request Message Body", Table "Table 58 - Request Message Body - Create a New Queue Object using CDMI Content Type"

Field Name	Type	Description	Requirement
deserializevalue	JSON String	A queue object serialized as specified in Clause 15 and encoded using base 64 encoding rules described in RFC 4648. If multi-part MIME is being used and this field contains the value of the MIME "boundary" parameter, the contents of the second MIME part shall be assigned as the field value. If the serialized queue object in the second MIME part does not include a "value" field the contents of the third and subsequent MIME parts shall be assigned as the field value of the "value" field.	Optional ^a

27) Append to end of Clause "9.11.9 Examples"

EXAMPLE 2 - POST to the container object URI a serialized queue object using multi-part MIME:

```
POST /MyContainer/ HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
Accept: application/cdmi-queue
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{
  "deserializevalue": "gc0p4Jq0M2Yt08j34c0p"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{
```

```

"objectType": "application/cdmi-queue",
"objectID": "0000706D0010B84FAD185C425D8B537E",
"objectName": "0000706D0010B84FAD185C425D8B537E",
"parentURI": "/MyContainer/",
"parentID": " 0000706D0010B84FAD185C425D8B537E",
"domainURI": "/cdmi_domains/MyDomain/",
"capabilitiesURI": "/cdmi_capabilities/queue/",
"completionStatus": "Complete",
"metadata": {},
"queueValues": "0-1",
"mimetype": [
  "text/plain",
  "text/plain"
],
"valuetransferencoding": [
  "utf-8",
  "utf-8"
],
"value": [
  "First Enqueued Value",
  "Second Enqueued Value"
]
}

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response:

```

HTTP/1.1 201 Created
Content-Type: application/cdmi-queue
X-CDMI-Specification-Version: 1.0.1
Location:
http://cloud.example.com/MyContainer/0000706D0010B84FAD185C425D8B537E

```

```

{
"objectType": "application/cdmi-queue",
"objectID": "0000706D0010B84FAD185C425D8B537E",
"objectName": "0000706D0010B84FAD185C425D8B537E",
"parentURI": "/MyContainer/",
"parentID": " 0000706D0010B84FAD185C425D8B537E",
"domainURI": "/cdmi_domains/MyDomain/",
"capabilitiesURI": "/cdmi_capabilities/queue/",
"completionStatus": "Complete",
"metadata": {},
"queueValues": "0-1"
}

```

EXAMPLE 3 - POST to the container object URI a serialized queue object and its values using multi-part MIME:

```

POST /MyContainer/ HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
Accept: application/cdmi-queue
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

```

```

{
  "deserializevalue": "gc0p4Jq0M2Yt08j34c0p"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{
  "objectType": "application/cdmi-queue",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "0000706D0010B84FAD185C425D8B537E",
  "parentURI": "/MyContainer/",
  "parentID": " 0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "0-1"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<20 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response:

```

HTTP/1.1 201 Created
Content-Type: application/cdmi-queue
X-CDMI-Specification-Version: 1.0.1
Location:
http://cloud.example.com/MyContainer/0000706D0010B84FAD185C425D8B537E

{
  "objectType": "application/cdmi-queue",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "0000706D0010B84FAD185C425D8B537E",
  "parentURI": "/MyContainer/",
  "parentID": " 0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "0-1"
}

```

28) Append to end of Clause "11.1 Overview"

The value of a queue object may also be specified and retrieved using multi-part MIME, where the CDMI JSON is transferred in the first MIME part and the raw queue values are transferred in the subsequent MIME parts. Each MIME part, including any header fields, shall conform to RFC 2045, RFC 2046, and RFC 2616, and the length of each part may optionally be specified by a Content-Length header in addition to the MIME boundary delimiter.

29) Append to end of Clause "11.2.3 Capabilities"

- Support for the ability to create a queue object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

30) Modify Clause "11.2.4 Request Headers", Table "Table 82 - Request Headers - Create a Queue Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdmi-queue" or "multipart/mixed". If "multipart/mixed" and the "deserializevalue" field is specified with the value of the MIME "boundary" parameter, the body shall consist of two or more MIME parts, where the first part shall contain a body of content-type "application/cdmi-queue", the second part shall contain the serialized queue object, and optionally the third and subsequent parts shall each contain a queue value as described in 8.5.	Mandatory

31) Modify Clause "11.2.5 Request Message Body", Table "Table 83 - Request Message Body - Create a Queue Object using CDMI Content Type"

Field Name	Type	Description	Requirement
deserializevalue	JSON String	A queue object serialized as specified in Clause 15 and encoded using base 64 encoding rules described in RFC 4648. If multi-part MIME is being used and this field contains the value of the MIME "boundary" parameter, the contents of the second MIME part shall be assigned as the field value. If the serialized queue object in the second MIME part does not include a "value" field the contents of the third and subsequent MIME parts shall be assigned as the field value of the "value" field.	Optional ^a

32) Append to end of Clause "11.2.9 Examples"

EXAMPLE 2 - PUT to the container object URI the queue object name and a serialized queue object and its values using multi-part MIME:

```
PUT /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
Accept: application/cdmi-queue
X-CDMI-Specification-Version: 1.0.1
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue
{
  "deserializevalue": "gc0p4Jq0M2Yt08j34c0p"
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{
  "objectType": "application/cdmi-queue",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyQueue",
  "parentURI": "/MyContainer/",
  "parentID": " 0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "0-1"
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

<20 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

<37 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response:

```
HTTP/1.1 201 Created
Content-Type: application/cdmi-queue
X-CDMI-Specification-Version: 1.0.1

{
  "objectType": "application/cdmi-queue",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyQueue",
  "parentURI": "/MyContainer/",
  "parentID": " 0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "0-1"
```

}

33) Append to end of Clause "11.3.2 Capabilities"

- Support for the ability to read a queue object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

34) Modify Clause "11.3.3 Request Headers", Table "Table 87 - Request Headers - Read a Queue Object using CDMI Content Type"

Header	Type	Description	Requirement
Accept	Header String	"application/cdmi-queue" or "multipart/mixed".	Mandatory

35) Modify Clause "11.3.5 Response Headers", Table "Table 88 - Response Headers - Read a Queue Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdmi-queue" or "multipart/mixed". If "multipart/mixed", the body shall consist of one or more MIME parts, where the first part shall contain a body of content-type "application/cdmi-queue", and the second and subsequent parts shall each contain a queue value as described in 8.5.	Mandatory

36) Modify Clause "11.3.6 Response Message Body", Table "Table 89 - Response Message Body - Read a Queue Object using CDMI Content Type"

Field Name	Type	Description	Requirement
value	JSON Array of JSON Strings	The oldest enqueued queue object values. <ul style="list-style-type: none"> • The values in the JSON array are returned in order from oldest to newest. • If the valuetransferencoding field indicates UTF-8 encoding, the corresponding value field shall contain a UTF-8 string using JSON escaping rules described in RFC 4627. • If the valuetransferencoding field indicates base64 encoding, the corresponding value field shall contain a base 64-encoded string as described in RFC 4648. • The value field shall not be provided when using multi-part MIME. • The value field shall only be provided when the completionStatus field contains "Complete". 	Conditional

37) Append to end of Clause "11.3.8 Examples"

EXAMPLE 5 - GET to the queue object URI to read the queue object using multi-part MIME:

```
GET /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Accept: multipart/mixed
X-CDMI-Specification-Version: 1.0.1
```

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue
```

```
{
  "objectType": "application/cdmi-queue",
  "objectID": "00007ED9001035E14BD1BA70C2EE98FC",
  "objectName": "MyQueue",
  "parentURI": "/MyContainer/",
  "parentID" : " 0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "1-2",
  "mimetype": [
    "application/octet-stream",
    "application/octet-stream"
  ],
  "valuerange": [
    "0-19",
    "0-36"
  ],
  "valuetransferencoding": [
    "base64",
    "base64"
  ]
}
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

<20 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

<37 bytes of binary data>

```
--gc0p4Jq0M2Yt08j34c0p--
```

38) Append to end of Clause "11.4.2 Capabilities"

- Support for the ability to modify an existing queue object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

39) Modify Clause "11.4.3 Request Headers", Table "Table 91 - Request Headers - Update a Queue Object using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdmi-queue" or "multipart/mixed". If "multipart/mixed" and the "deserializevalue" field is specified with the value of the MIME "boundary" parameter, the first part shall contain a body of content-type "application/cdmi-queue", the second part shall contain the serialized queue object, and the subsequent parts shall optionally contain the queue values as described in 8.5.	Mandatory

40) Modify Clause "11.4.4 Request Message Body", Table "Table 92 - Request Message Body - Update a Queue Object using CDMI Content Type"

Field Name	Type	Description	Requirement
deserializevalue	JSON String	URI of a serialized CDMI queue object that shall be deserialized to update an existing queue object. The object ID of the serialized queue object shall match the object ID of the destination queue object. All enqueued items in the serialized queue object shall be added to the destination queue object. If multi-part MIME is being used and this field contains the value of the MIME "boundary" parameter, the contents of the second MIME part shall be assigned as the field value. If the serialized queue object in the second MIME part does not include a "value" field the contents of the third and subsequent MIME parts shall be assigned as the field value of the "value" field.	Optional ^a

41) Append to end of Clause "11.4.8 Examples"

EXAMPLE 2 - PUT to the queue object URI a serialized queue object and its values using multi-part MIME:

```
PUT /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue
```

```

{
  "deserializevalue": "gc0p4Jq0M2Yt08j34c0p"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{
  "objectType": "application/cdmi-queue",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyQueue",
  "parentURI": "/MyContainer/",
  "parentID": "0000706D0010B84FAD185C425D8B537E",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "0-1"
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<20 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--

```

The following shows the response.

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

42) Append to end of Clause "11.6.2 Capabilities"

- Support for the ability to modify the value of an existing queue object using multi-part MIME is indicated by the presence of the "cdmi_multipart_mime" system-wide capability.

43) Modify Clause "11.6.3 Request Headers", Table "Table 97 - Request Headers - Enqueue a New Queue Object Value using CDMI Content Type"

Header	Type	Description	Requirement
Content-Type	Header String	"application/cdmi-object" or "multipart/mixed". If "multipart/mixed", the first part shall contain a body of content-type "application/cdmi-queue", and the subsequent parts shall contain the queue values as described in 8.3.	Mandatory

44) Modify Clause "11.6.4 Request Message Body", Table "Table 98 - Request Message Body - Enqueue a New Queue Object Value using CDMI Content Type"

Field Name	Type	Description	Requirement
mimetype	JSON Array of JSON Strings	MIME type of the data to be enqueued into the queue object. <ul style="list-style-type: none"> • This field shall be stored as part of the object. • If this field is not included and multi-part MIME is not being used, the value of "text/plain" shall be assigned as the field value. • If this field is not included and multi-part MIME is being used, the value of the "Content-Type" header of the corresponding MIME part shall be assigned as the field value. • The same number of array elements shall be present as is present in the value field, and the mimetype shall be associated with the value in the corresponding position. • This mimetype value shall be converted to lowercase before being stored. 	Optional

Field Name	Type	Description	Requirement
valuetransferencoding	JSON Array of JSON Strings	<p>The value transfer encoding used for the queue object value. Two value transfer encodings are defined.</p> <ul style="list-style-type: none"> • "utf-8" indicates that the data object contains a valid UTF-8 string, and it shall be transported as a UTF-8 string in the value field. • "base64" indicates that the data object may contain arbitrary binary sequences, and it shall be transported as a base 64-encoded string in the value field. Setting the contents of the data object value field to any value other than a valid base 64 string shall result in error 400 Bad Request being returned to the client. • If this field is not included and multi-part MIME is not being used, the value of "utf-8" shall be assigned as the field value. • If this field is not included and multi-part MIME is being used, the value of "utf-8" shall be assigned as the field value if the "Content-Type" header of the corresponding MIME part includes the charset parameter as defined in RFC 2046 of "utf-8" (e.g., ";charset=utf-8"). Otherwise, the value of "base64" shall be assigned as the field value. This field applies only to the encoding of the value when represented in JSON; the "Content-Transfer-Encoding" header of the part specifies the encoding of the value within a multi-part MIME request, as defined in RFC 2045. • This field shall be stored as part of the object. 	Optional
value	JSON Array of JSON Strings	<p>Data to be enqueued into the queue object.</p> <ul style="list-style-type: none"> • If this field is not included and multi-part MIME is being used, the contents of the MIME parts shall be assigned as the field value. • If the corresponding valuetransferencoding field indicates UTF-8 encoding, the value shall be a UTF-8 string escaped using the JSON escaping rules described in RFC 4627. • If the corresponding valuetransferencoding field indicates base64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648. 	Optional ^a

45) Append to end of Clause "11.6.8 Examples"

EXAMPLE 6 - POST to the queue object URI the binary contents of two new values using multi-part MIME:

```
POST /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<20 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<37 bytes of binary data>

--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

```
HTTP/1.1 204 No content
```

EXAMPLE 7 - POST to the queue object URI the mime types and binary contents of two new values using multi-part MIME:

```
POST /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: multipart/mixed; boundary=gc0p4Jq0M2Yt08j34c0p
X-CDMI-Specification-Version: 1.0.1

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/cdmi-queue

{
  "mimetype" : [
    "application/pdf",
    "image/jpeg"
  ]
}

--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary

<20 bytes of binary data>
```

```
--gc0p4Jq0M2Yt08j34c0p
Content-Type: application/octet-stream
Content-Transfer-Encoding: binary
```

```
<37 bytes of binary data>
```

```
--gc0p4Jq0M2Yt08j34c0p--
```

The following shows the response.

```
HTTP/1.1 204 No content
```

46) Insert into Clause "12.1.1 Cloud Storage System-Wide Capabilities", Table "Table 102 - System-Wide Capabilities"

Capability Name	Type	Description
cdmi_multipart_mime	JSON String	If present and "true", this capability indicates that the cloud storage system supports storing and retrieving the value of data and queue objects using multi-part MIME.
cdmi_create_value_range_by_ID	JSON String	If present and "true", this capability indicates that the system allows a new data object's value to be created with byte ranges through "/cdmi_objectid/".

47) Insert into Clause "12.1.5 Container Capabilities", Table "Table 106 – Capabilities for Containers"

Capability Name	Type	Description
cdmi_create_value_range	JSON String	If present and "true", this capability indicates that the container allows a new data object's value to be created with byte ranges.