

CDMI 1.0.1 Errata

for Cloud Data Management Interface Version 1.0

Trac Ticket #39

Reintroduce optional power/green account summary items.

Description: At some point during the editing over the last couple of months, the account summary items measuring power that we talked about with the Green TWG were dropped. Should we re-introduce these?

Discussed proposal was to introduce one or more of the below optional domain summary fields:

- `cdmi_summary_kwhours`, JSON String, The sum of power consumed by the domain during the summary time period, Optional
- `cdmi_summary_kwmin`, JSON String, The maximum power consumption by the domain during the summary time period, Optional
- `cdmi_summary_kwmax`, JSON String, The minimum power consumption by the domain during the summary time period, Optional
- `cdmi_summary_kwaverage`, JSON String, The average power consumption by domain during the summary time period, Optional

This is ready to be incorporated into the specification.

CHANGES: Added these fields to Table 5 - Contents of Domain Summary Objects in Section 10.1.2.

- Changed the description for `cdmi_summary_kwmin` from “maximum” to “minimum.”
- Changed the description for `cdmi_summary_kwmax` from “minimum” to “maximum.”

Trac Ticket #79

Need syntax to indicate zero children.

Currently we are using "0-0" to indicate no children in the `childrenrange` field. According to the convention presented yesterday, this would indicate one item. We need to decide on a convention to indicate zero items in a range and make the corresponding updates to the spec.

Proposal for discussion:

- "" - Empty string, no items
- "0-0" - one item
- "0-10" - eleven items

CS TWG agreed with the proposal. Marie will make the update.

CHANGES: Changed "0-0" to "" in Sections 6.2 and 9.2.

Trac Ticket #81 Adding Versioning to Annex B.

Description: Add versioning to annex B from 2/23/10 face-to-face discussions.

Proposal for discussion:

Changed 5 months ago by dslik: Proposal based on discussions at the San Jose face-to-face uploaded to the group for discussion tomorrow.

Changed 4 months ago by hhines: Add to annex B as an example of how to extend the interface. David will modify to use example vendor.

Changed 2 months ago by dslik: Modified and uploaded to the document repository.

Changed 2 days ago by Mark: owner changed from dslik to marie. Marie - please incorporate this into Annex B in the next release of the Errata version.

CHANGES: Added 2010-05-19 CDMI Versioning R6.doc to Annex B - Data Object Versioning.

Trac Ticket #82 Disambiguate `max_throughput` and `max_latency`.

There was a significant amount of confusion at the CDMI presentation about the meanings of `cdmi_max_latency` and `cdmi_max_throughput`. The decision was to reevaluate the names of these two fields to determine if a replacement might be less confusing.

- 1 Change `cdmi_max_latency` to `cdmi_latency` everywhere in the document.
- 2 Change `cdmi_max_throughput` to `cdmi_throughput` everywhere.
- 3 Change `cdmi_max_latency_billed` to `cdmi_latency_billed` everywhere.
- 4 Change `cdmi_max_throughput_billed` to `cdmi_throughput_billed` everywhere.

CHANGES: Changes made to 12.1.3, 16.4, and 16.5.

Trac Ticket #86 **Non-CDMI Container PUT should have URI ending in "/"**.

Operations performed on containers should have a URI that ends in "/". This is required for the non-CDMI container PUT but should also be present for CDMI container operations.

CHANGES Added "/" to Container operations in Sections 8.2, 8.3, 8.6, 8.7, 9.2, 9.3, 9.6, 11.2, and 11.4.

Note: Some of these changes were reversed; see "Trac Ticket #149" on page 17.

Trac Ticket #87 **Incorrect HTTP Response Code for DELETE**

According to the HTTP RFC, for DELETE methods:

"A successful response SHOULD be 200 (OK) if the response includes an entity describing the status, 202 (Accepted) if the action has not yet been enacted, or 204 (No Content) if the action has been enacted but the response does not include an entity."

We currently specify that a "200 OK" should be returned for a successful DELETE but also specify that the response shall not include an entity. Thus, the response code specified should be "204 No Content".

CHANGES: Changed "200 OK" to "204 No Content" in example in Section 6.7 and in tables and examples in Sections 8.8, 8.9, 9.7, 9.8, 10.5, 11.5, and 11.7.

Trac Ticket #92 **Need way to recover from failure during queue delete.**

When you dequeue an item, a failure may result in the client not knowing if the DELETE transaction completed. Currently, there is no way for the client to easily determine if the DELETE completed.

One way to solve this problem is to add a unique value identifier for each enqueued item. That way, when the client performs a DELETE and they don't know if it succeeded, they can do a GET for this unique value identifier to determine if the DELETE completed and reissue the DELETE, if required.

Proposed method is to change QueueValues?? into a range, where the first value goes up as items are deleted, and the second value goes up as items are enqueued.

e.g.,

- Create: ""
- Enqueue: "0-0"
- Enqueue: "0-1"
- Enqueue: "0-2"
- Delete: "1-2"

- Delete: "2-2"
- Delete: ""
- Enqueue: "3-3"

CHANGES: Changed description of *queuevalues* field in Sections 11.2 and 11.3. Changed *queuevalue* example output in Sections 11.2, 11.3, and 15.2.3.

Trac Ticket #93 Complete Section 13 for iSCSI.

Based on discussion during face-to-face meeting, Scott will write up text and send to Marie for adding to CDMI errata spec.

Content for Section 13 uploaded 3/30/2010 to API folder.

Reviewed content for Section 13 at last week's meeting. iSCSI ready to go into errata document.

Include "container export capability" document into the SNIA Architecture document.

CHANGES: Added/formatted contents of iSCSI Export.doc to Section 13.3.

Trac Ticket #95 Remove FCOE from Section 13.

CHANGES: Removed section.

13.5 FCOE Exported Protocol
 CDMI defines an export protocol structure for the FCoE standard as follows:

- Protocol is "Network/FCoE"
- World-Wide Port Name (WWPN) and Logical-unit name

Note: This section is incomplete and needs further definition.

Trac Ticket #96 Inconsistent JSON for single vs. multiple ACE ACLs

In 16.1.11, the format of the JSON is different if there is a single ACE vs. when there are multiple ACEs. We should be consistent and always include a JSON array. Drop first ACL and example in Section 16.1.11. David will work with Marie for the changes.

CHANGES Changed the following in Section 16.1.1:

- 1 Deleted 2nd paragraph beginning "ACLs containing a single ACE" and the JSON example that followed it.
- 2 In third paragraph, (new 2nd paragraph), changed "multiple" to "one or more."

- 3 In second example, to create the JSON array, enclosed the following code in square brackets:

```
{
  "acetype" : "0x00",
  "identifier" : "EVERYONE@",
  "aceflags" : "0x00",
  "acemask" : "0x00020089",
  "acetime" : 12345678
}
```

- 4 Reformatted the JSON code.

Trac Ticket #116 Add bulk queue operations.

Add the ability to perform bulk enqueue and dequeue operations to Annex B.

One approach would be to add GET/PUT <root URI>/<ContainerName?>/<QueueName?>?items:<range> and to use JSON arrays for the value/content-type. This would only be for CDMI-style PUT and GET. A similar syntax can be used to allow multi-delete. This approach depends on the solution proposed in trac ticket #92. Will discuss at face-to-face meetings.

Proposed specification changes uploaded to Kavi. See 2010-03-24 CDMI MultiOp Queue Spec Update.doc

CHANGES Changes made in the following sections:

- 11.3 “Read a Queue Object (CDMI Content Type)”
- 11.6 “Enqueue a New Queue Value (CDMI Content Type)”
- 11.7 “Delete a Queue Value (CDMI Content Type)”

Trac Ticket #117 Missing capability for snapshots

We need to add a "cdmi_snapshots" system-wide capability to Section 12.1.1 to indicate if a system supports snapshots.

Proposal: Add to Table 13:

- cdmi_snapshots
- If present and "true", the cloud storage system supports snapshots.

CHANGES Added the above information to Table 13 - System-Wide Capabilities in Section 12.1.1.

Trac Ticket #118 Change `cdmi_delete_method` to `cdmi_sanitization_method`.

Change `cdmi_delete_method` to `cdmi_sanitization_method` in CDMI.

CHANGES Made the referenced change to Table 15 - Capabilities for Data System Metadata in Section 12.1.3.

Trac Ticket #119 Clarify *cdmi_encryption* capability description.

The text description of the *cdmi_encryption* data system metadata is not clear about what is encrypted when enabled. At the San Antonio face-to-face meeting, it was indicated that the intent for this capability is for all data related to the data object/container to be encrypted when set, including metadata, and the spec text should be updated to clearly indicate this.

Add the following to the end of the description for *cdmi_encryption* in Table 24:

All data related to the data object/container shall be encrypted when this value is set, including metadata.

CHANGES Added the specified text to Table 24 - Metadata for CDMI Copies.

Trac Ticket #120 Add *cdmi_serialize_container* to table 17.

Need to add missing capability in table.

CHANGES None. *cdmi_serialize_container* is already in Table 17.

Trac Ticket #121 Verify that *serialize* and *deserialize* are used together.

Verify in CDMI that where *serialize* is used that *deserialize* is used and vice versa. In other words, we need to make sure that in the tables of capabilities, that we consistently have capabilities for being able to *serialize* and *deserialize* the basic CDMI object types, *dataobject*, *container*, *queue*, and *domain*.

CHANGES Added *cdmi_deserialize_container* to Table 17.

Trac Ticket #122 Delete duplicate capability entry for *serializing a queue*.

Delete duplicate entries for *serializing a queue* in table 19.

CHANGES Deleted *cdmi_serialize_queue* from Table 19.

Trac Ticket #123 Get CDMI MIME types

Description: Get CDMI MIME types.

Change history:

- Note that we are likely removing the `cdmi.object` mime type from the specification and we should not request this in the RFC.
- I believe this is complete, and already applied in the 1.0.1f spec version.
- Waiting for RFC number to add to normative references section. Status can be monitored at: <http://www.iana.org/assignments/media-types/application/index.html>.
- We now have a RFC number, so we should be able to proceed with this update to the normative references section. Once published it will be at <http://www.ietf.org/rfc/rfc6208.txt>.

Discussed on April 13 TWG call and decision is to include in 1.0.1i.

- The RFC link is not yet active, and we wouldn't be able to release 1.0.1i until such time as this link becomes active.
- The IETF link is now active, so we can update the normative references section and close off this ticket.

CHANGES Added [RFC6208] to Normative References in Chapter 2.

Trac Ticket #124 Clarify response for operations against references

Description: We need to update the specification text for individual operations to more clearly indicate that GET/PUT/POST operations against a reference do not return a response body. (The provider responds with a 302 redirect.)

We should also add an example of an operation against a reference.

Here is an example to add to the end of section 7.3. This should make it clear that no response body is returned.

Example Request:

```
GET to a URI, where the URI is a reference:
GET /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/vnd.org.snia.cdmi.dataobject+json
X-CDMI-Specification-Version: 1.0
```

The response looks like:

```
HTTP/1.1 302 Found
Location: http://cloud.example.com/MyContainer/MyOtherDataObject.txt.
```

Note that the wiki markup mangled the last line in the example. It should not have a period at the end.

CHANGES Added the example request at the end of Section 7.3 Object References.

Trac Ticket #128 Return 403 on hold instead of 407

The complete fix for this ticket is to change '407' to '403' in the document. Unfortunately, I have no idea how to resolve this ticket and my request for help on the mailing list is as yet unanswered. The change is straightforward; I think anyone who knows the procedure can resolve this ticket.

CHANGES Changed 407 to 403 in Section 18.2 CDMI Retention and Section 18.3 CDMI Hold.

Trac Ticket #130 Investigate whether CDMI clobber can be replaced with standard HTTP.

After group discussion, the following was approved. Etag functionality is a larger and thornier issue than what was proposed. Suggest a new, separate trac ticket and face-to-face discussion for that.

Marie: please make the following changes:

- 1 Drop all mention of X-CDMI-NoClobber?? and X-CDMI-MustExist??
- 2 Add the following text to Section 5:

5.13 Required HTTP Support

CDMI-compliant implementations shall support the following RFC 2616 (HTTP/1.1) features:

5.13.1 If-Match: *

CDMI clients can ensure that they are updating an existing resource by including an If-Match header with the value *. The operation will fail with 412 (Precondition Failed) if there is no current resource.

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
If-Match: *
```

```
HTTP/1.1 412 Precondition Failed
```

5.13.2 If-None-Match: *

CDMI clients can ensure that they are creating a new resource by including an If-None-Match header with the value *. The operation will fail with 412 (Precondition Failed) if there is a current resource.

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
If-None-Match: *
```

```
HTTP/1.1 412 Precondition Failed
```

5.13.3: Etag

All CDMI GET and HEAD responses will include an Etag header. The value of the Etag header is opaque, but a) must change when the resource changes, and b) must not change until the resource changes.

5.13.4: If-Match: <etag>

CDMI clients can ensure that they are updating the current version of a resource by supplying an If-Match header with the resources Etag. The operation will fail with 412 (Precondition Failed), if the Etag of the resource does not match the value supplied in the If-Match header (typically because it has been updated by another client).

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
If-Match: "abcdefg"

HTTP/1.1 412 Precondition Failed
```

CHANGES All changes made as follows:

- 1 X-CDMI-NoClobber removed from Sections 8.2, 9.2, 10.2, and 11.2.
- 2 X-CDMI-MustExist removed from Sections 8.6, 9.6, 10.4, and 11.4.
- 3 Added Section 5.13 Required HTTP Support and subsections as referenced in trac ticket.

Trac Ticket #134 Updates to Figure 8

Change account to domain.

CHANGES Changed account to domain in Figure 8 - Hierarchy of Capabilities.

Trac Ticket #135 Specify Capabilities JSON data type.

Lists of capabilities, e.g., 12.1.7, should clearly indicate that they are a JSON string, with value "true" that means ...

See 16.4.16 for an example of this.

CHANGES Added "A JSON string that" to every capabilities definition in Sections 12.1.1, 12.1.2... 12.1.7 (Tables 13-19).

Trac Ticket #137 CDMI Object IDs contain non URI-safe characters

Currently, CDMI Object IDs are encoded with Base64. However, this can contain characters that are not safe for use unescaped in URIs, such as "/".

- 1 Propose to use hex instead of Base64. David will provide Marie with hex examples instead of Base64. See e-mail to mailing list with examples.

- Note that we also have to change the last bullet of Section 5.11 in the spec.

CHANGES Changed the following Base64 examples with hex examples:

- Changed the *objectid* value to 0000706D00100C435125A61B4C289455 in 6.1.
- Changed the *objectid* value to 0000706D0010D538DEEE8E38399E2815 in 6.2 and 6.4.
- Changed the *objectid* value to 0000706D0010734CE0BAEB29DD542B51 in 6.3, 6.5, and 16.1.1.
- Changed the *objectid* and/or *identifier* values to 0000706D0010B84FAD185C425D8B537E in 8.1, 8.2., 8.4, 9.1, 9.2, 9.3, 9.6, 9.9, and 9.10.
- Changed the *permissions* value to "0000706D00107B85BFE6D20B84D603CA" in 9.2, 9.4, 9.6, 13.2, and 15.2.3.
- Changed the *objectid* value to 0000706D00101ADEBC119D1BFE98672A in 10.2, 10.3, 11.2, 11.3, 12.2, 13.2, and 15.2.3.
- Changed the *cdmi_objectID* value to 0000706D0010D538DEEE8E38399E2815 in 11.1.1 and 11.1.3.
- Changed the *cdmi_objectID* value to 0000706D0010734CE0BAEB29DD542B51 in 11.1.3.

Trac Ticket #138 Miscellaneous CDMI Spec Issues - Errata

- Section 13.1 Extended Protocol Structure, second bullet - Change "identify" to "identity". (Done)
- Table 22; description for *cdmi_acl* - Remove the "(Based on NFSv4)" comment and adjust the description as necessary. (Done)
- There seems to be some inconsistencies between the *cdmi_hash* and *cdmi_value_hash* descriptions in Tables 14, 15, 22, and 23; the *cdmi_hash* description in Table 14 seems to be the source of the problem. For hashing, we need to be able to determine the hashing options (if any), request a particular hash, be told what hash is actually being used (*cdmi_value_hash_billed*), and then store a hash value in the metadata for each item that is hashed. NOTE: It is unclear how the hash is calculated (i.e., over what data and/or metadata). (No changes.)
- Table 24 - Metadata for CDMI Data Copies currently contains the full description of *cdmi_encryption*, which is split over two table entries; these two table entries should be combined. More importantly, the *cdmi_encryption* description in Table 24 should actually be included in Table 23 (added); the existing description in Table 24 could remain or be reduced like the *cdmi_encryption_billed* in Table 25. (Done.)
- Hashes currently only cover the value of the object, hence, *cdmi_value_hash*. There was some discussion about having a *cdmi_metadata_hash* covering the metadata, and/or a *cdmi_object_hash* covering the entire object. (No changes.)
- It looks like the hash field names got mixed up in 12.1.2. Here's what it should be: (No changes to descriptions, except 12.1.2, as described in next bullet.)

- Section 12.1.2 - The *cdmi_hash* storage system metadata capability indicates if the system can generate value hashes and place them in storage system metadata.
 - Section 12.1.3 - The *cdmi_value_hash* data system metadata capability indicates which algorithms are supported for the value hash.
 - Section 16.3 - The Storage System Metadata *cdmi_hash* contains a hash, if requested and supported.
 - Section 16.4 - The Data System Metadata *cdmi_value_hash* allows a client to request a hash in a given algorithm/key length.
 - Section 16.4.16 - The Data System Metadata Billed *cdmi_value_hash_billed* indicates to a client what hash algorithm/key length is being used. If present, there will be a hash in the storage system metadata *cdmi_hash* metadata item.
- 7 Only the description of *cdmi_hash* in Section 12.1.2 needs to be changed. It should read:
- "If present, the cloud storage system will generate a *cdmi_hash* storage system metadata for each stored object using the algorithm specified in the value of the *cdmi_value_hash* data system metadata." (Done.)

CHANGES See comments in red.

Trac Ticket #142 Query queue text is incorrect.

The text around Table 11 is incorrect. It states,

"When performing a query, the metadata described in Table 11 shall be PUT to a queue.

Table 11 – Required Query Metadata PUTS to a Queue"

This is not correct, as this is not metadata. It should read,

"When performing a query, the JSON elements described in Table 11 shall be PUT to a query queue. (Done)

Table 11 - Required Query JSON" (Done)

CHANGES See comments in red.

Trac Ticket #143 Normalize tables in Framemaker version.

We have many different table columns, which we should consider normalizing. We should also have consistent table column widths.

- 1 All HTML header tables should be:
|Header|Type|Description|Mandatory/Optional| (Done)
- 2 All JSON bodies should be

|Name|Type|Description|Mandatory/Optional|

Added Mandatory/Optional column to Table 20 - Snapshot Parameter of the Container Update Operation

- 3 When we talk about metadata, we should have:

|Metadata Name|Type|Description|Permissions|

"Source" should be dropped, as permissions indicates if a client can modify or if it is read-only.

- Added "Type" and Removed "Source" column from 9.1.1, 10.1.1, Tables 5-11, and Tables 22-24.
- Need Type added to all of these tables, except Table 11.
- Need Permissions added to Table 11.
- Need Type and Permissions added to Table 24.

- 4 When we talk about JSON data structures, we should have:

|Field Name|Type|Description|

Changed "Data Type" to "Type" and removed "Source" column in 11.1.1 and Table 12.

- 5 For capabilities, we should have:

|Capability Name|Type|Description|

- Changed "Capability" to "Capability Name" in Tables 13-19.
- Added "Type" column to Tables 13-19.
- Added "JSON String" to the "Type" column for each capability name in Tables 13-19.

CHANGES Made all changes as indicated. See comments in red.

Trac Ticket #144 Section 16.5 Duplicates 16.4

Description: Section 16.5, "Support for Data Copies" duplicates much of section 16.4. We should merge these sections together. dslik will update this trac ticket with merged table text for Marie.

Proposed merged table below:

- Metadata Name
 - Type
 - Description
 - Mandatory/Optional
- cdmi_data_redundancy
 - JSON String
 - Contains the desired number of complete copies of the data object to be maintained. This determines the minimum number of primary copies of the data

that the cloud shall maintain. Additional primary copies may be made to satisfy demand for the value.

- Optional
- *cdmi_immediate_redundancy*
 - JSON String
 - If present and set to the value "true", indicates that at least a *cdmi_data_redundancy* number of copies will contain the newly written value before the operation completes. This metadata is used to make sure that multiple copies of the data are written to permanent storage to prevent possible data loss.
 - Optional
- *cdmi_assignedsize* (only valid for a container object.) See Chapter 9, "Container Objects".
 - JSON String
 - Contains the number of bytes that are reported via exported protocols (and may be thin provisioned by the system). This number may limit *cdmi_size* for the container. This metadata is the size that will be shown through any number of data path protocols that are used to export a container. If the container is thin provisioned, this may be greater than the actual storage consumed.
 - Optional
- *cdmi_infrastructure_redundancy*
 - JSON String
 - Contains the number of desired independent storage infrastructures supporting the data. This metadata is used to convey that, of the primary copies specified in *cdmi_data_redundancy*, these copies will be stored on this many separate infrastructures. Any two infrastructures cannot share common elements, such as a network or power source.
 - Optional
- *cdmi_data_dispersion*
 - JSON String
 - Contains the desired distance (km) between the infrastructures supporting the multiple copies of data. This metadata is used to separate the (*cdmi_infrastructure_redundancy* number of) infrastructures by a minimum geographic distance to prevent data loss due to site disasters.
 - Optional
- *cdmi_geographic_placement*
 - JSON Array of JSON Strings
 - Contains a list of geopolitical identifiers, each specifying a region where the object is permitted to or not permitted to be stored. Geopolitical boundaries are a list of ISO-3166 country codes. A "!" in front of a country code excludes that country from the previous list of geopolitical boundaries. This metadata limits where the data may be placed physically and constrains all cloud move-

ment of the data within the cloud. It does not apply to data once it leaves the cloud. This metadata takes precedent over other metadata, such as `cdmi_data_dispersion`.

- Optional
- *cdmi_retention_id*
 - JSON String
 - Contains a user-specified retention identifier. This metadata is a user-specified text field that is used to tag a given object as being managed by a specific retention policy. It is not required to place an object under retention but is useful when needing to be able to perform a query to find all objects under a specific retention policy.
 - Optional
- *cdmi_retention_period*
 - JSON String
 - Contains an [ISO-8601] time interval specifying the period the object is to be protected by retention. This metadata is the time interval (in either an [ISO-8601] date-duration or an [ISO-8601] date-date) during which the object is under retention. Only the duration or end-date can be altered when updated. If an object is under retention, the object cannot be deleted and its value cannot be altered. After the retention duration has elapsed, the object can be deleted.
 - Optional
- *cdmi_retention_autodelete*
 - JSON String
 - This metadata is used to indicate if the object is to be automatically deleted when retention expires. The value of this metadata item shall be "true" when set.
 - Optional
- *cdmi_hold_id*
 - JSON Array of JSON Strings
 - This metadata is used to indicate if the object is to be placed under a retention hold. If the array is not empty, the object is under a hold, with each string in the array containing a user-specified hold identifier. If an object is under one or more holds, the object is completely immutable.
 - Optional
- *cdmi_encryption*
 - JSON String
 - This metadata is used to indicate if the object is to be encrypted, and indicates the desired encryption algorithm, the mode of operation, and the key size. This metadata is the desired encryption support that the client is requesting of the cloud. All data related to the data object/container shall be encrypted when this value is set, including metadata. This metadata is the desired encryption support that the client is requesting of the cloud. Using the template,

ALGORITHM_MODE_KEYLENGTH, the client is able to specify the encryption where:

- "ALGORITHM" is the encryption algorithm (e.g., "AES" or "3DES")
- "MODE" is the mode of operation (e.g., "XTS", "CBC", or "CTR")
- "KEYLENGTH" is the key size (e.g., "128", "192", "256")

To improve interoperability between CDMI implementations, the following designators should be used for the more common encryption combinations:

- "3DES_ECB_168" for the three-key Triple DES algorithm, the Cipher Block Chaining (CBC) mode of operation, and a key size of 168 bits
- "AES_CBC_128" for the AES algorithm, the CBC mode of operation, and a key size of 128 bits,
- "AES_CBC_256" for the AES algorithm, the CBC mode of operation, and a key size of 256 bits,
- "AES_XTS_128" for the AES algorithm, the XTS mode of operation, and a key size of 128 bits
- "AES_XTS_256" for the AES algorithm, the XTS mode of operation, and a key size of 256 bits

— Optional

- *cdmi_value_hash*

— JSON String

— This metadata is used to indicate if the object data is to be hashed, and indicates the desired hash algorithm and length. Supported algorithm/length values are provided by the "cdmi_value_hash" capability.

— Optional

- *cdmi_latency*

— JSON String

— Contains the desired maximum time to first byte, in milliseconds. This metadata is the desired latency (in milliseconds) to the first byte of data in a primary copy, as measured from the edge of the cloud and factoring out any propagation latency between the client and the cloud. For example, this metadata may be used to determine, in an interoperable way, from what type of storage medium the primary copy(s) of the data may be served.

— Optional

- *cdmi_throughput*

— JSON String

— Contains the desired maximum data rate on retrieve, in bytes per second. This metadata is the desired bandwidth (in Mbits/ sec) to the primary copy of data, as measured from the edge of the cloud and factoring out any bandwidth capability between the client and the cloud. This metadata is used to stage the

primary data copies in locations where there is sufficient bandwidth to accommodate a maximum usage.

- Optional
- *cdmi_RPO*
 - JSON String
 - Contains the largest acceptable duration in time between an update and when the update can be recovered, in [ISO-8601] duration representation. This metadata is used to indicate the desired backup frequency from the primary copy(s) of the data to the secondary copy(s). It is the maximum acceptable duration between a write to the primary copy and the backup to the secondary copy during which a failure of the primary copy(s) will result in data loss.
 - Optional
- *cdmi_RTO*
 - JSON String
 - Contains the largest acceptable duration in time to restore data, in [ISO-8601] duration representation. This metadata is used to indicate the desired maximum acceptable duration to restore the primary copy(s) of the data from a secondary backup copy(s).
 - Optional

CHANGES Merged Table 17 and Table 18 with above-referenced data.

- Deleted Table 18.
- Changed cross-reference in 16.5 Support for Data Copies to point to Table 17.
- Italicized all field names in Table 17.

Trac Ticket #145 Field name examples should explicitly show allowed multiplicity

Description: The examples showing how to request different fields is missing the indication that multiple field names can be requested for the value example.

For example, in the queues section, the text,

```
GET <root URI>/<ContainerName?>/<QueueName?>?value:<range>
```

should include the ;... at the end, and read as:

```
GET <root URI>/<ContainerName?>/<QueueName?>?value:<range>;...
```

The change only needs to be made to section 11.3:

```
GET <root URI>/<ContainerName?>/<QueueName?>?value:<range>
```

to

```
GET <root URI>/<ContainerName?>/<QueueName?>?value:<range>;...
```


- CHANGES** Made the referenced change in Section 11.3 Read a Queue Object (CDMI Content Type)

Trac Ticket #146 WebDAV export section is missing from Spec.

- 1 Add proposed new section:
 - 13.6 WebDAV Exported Protocol
 - CDMI defines an export protocol structure for the WebDAV [RFC4918] standard as follows:
 - Protocol is "Network/WebDAV"
 - The path of the WebDAV mount point as presented to clients (including server host name)
 - The list of who can access the share is determined by the standard CDMI ACLs for each resource as exported via WebDAV
 - This example shows an WebDAV export protocol structure in JSON:


```
"Network/WebDAV" : { "identifier" : "/users", "permissions" : "domain" }
```
 - In this example, the value "domain" in the permissions field indicates that user credentials should be mapped through the domain membership in the domain of the CDMI container being exported.
 - WebDAV supports locking, but it is up to implementations to support any locking of access through the CDMI as a result, and the interaction between the two protocols is purposely not described in this standard.
- 2 Add reference to Section 2:
 - [RFC4918] HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV) - <http://www.ietf.org/rfc/rfc4918.txt>

- CHANGES**
- 1 Added Section 13.5 WebDAV Exported Protocol with the above-referenced text.
 - 2 Added reference to Section 2.

Trac Ticket #147 Fix diagram 6.1 to have correct names.

The name in diagram 6.1 need to be updated.

- 1 "<offering>/Capabilities" should be changed to <offering>/cdmi_capabilities"
- 2 "<offering>/Accounts" should be changed to "<offering>/cdmi_domains".

- CHANGES** Changed Figure 6 - CDMI Interface Model in Section 5.8 as indicated.

Trac Ticket #149 Errata Bundle #1

Review of the draft errata, part 1:

- 1 Section 5.11

Last bullet needs to be updated.

Suggested replacement:

"The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation should be hex-encoded."

(Replaced)

2 Section 6.2

The PUT operation should have a trailing "/" on the URI, as per Trac Ticket #86.

```
PUT /MyContainer/ HTTP/1.1
```

(Added)

3 Section 6.3

In the response body, the `cdmi_size` metadata item should be inside the metadata object:

```
"mimetype" : "text/plain",
"metadata" : {
"cdmi_size" : "17"
}
```

(Fixed the code.)

4 Section 6.4

The `childrenrange` field is incorrect. It should read:

```
"childrenrange" : "0-0",
```

The metadata object is missing the `cdmi_size` example metadata item. It should include an example item, such as:

```
"cdmi_size" : "83"
```

(Added)

5 Section 6.5

The `valuerange` field is incorrect, as per Trac Ticket #79. It should read:

```
"valuerange" : "0-16",
```

(Fixed the code.)

6 Section 6.7

"GET from the Root URI:" should be replaced with "Perform a DELETE to the data object URI:"

(Replaced)

7 Section 8.2

The PUT example should not have a trailing slash at the end of the URI.

(Removed)

The first and second example response bodies should include a `cdmi_size` metadata item.

```
"cdmi_size" : "37"
```

(Added)

8 Section 8.3

The PUT example should not have a trailing slash at the end of the URI.

(Removed)

9 Section 8.4

- The first example response body should include a *cdmi_size* metadata item.

```
"cdmi_size" : "37"
```

(Added)

- The first example response body valuerange field should be "0-36"

```
"valuerange" : "0-36",
```

(Changed)

- The third example response body value field should be "This is the"

```
"value" : "This is the"
```

(Changed)

10 Section 8.6

None of the PUT examples should have a trailing slash.

```
PUT <root URI>/<ContainerName?>/<DataObjectName?>
```

```
PUT <root URI>/<ContainerName?>/<DataObjectName?>?metadata
```

```
PUT <root URI>/<ContainerName?>/<DataObjectName?>?value
```

```
PUT <root URI>/<ContainerName?>/<DataObjectName?>?value:<range>
```

(Removed)

CHANGES All changes made as indicated. See comments in red.

Trac Ticket #152 Handling MIME Type

Section 8.2 P 27 Request Message body table *mimetype* row says "This field shall *not* be included when ..."

We should changed the sentence to "This field shall be kept as part of the metadata and shall be included when ..."

Note: There is a slight dichotomy how mime type is handled. Maybe we should make it part of the metadata instead of a top-level attribute.

The text, "This field shall not be included when deserializing, serializing, copying, moving, or referencing a data object." means that the field should not be supplied as part of the message body of this operation. MIME type would be preserved on the operation as it is retained from the source object.

CHANGES Made specified change to mimetype description in Request Message Body table in Section 8.2 Create a Data Object (CDMI Content Type).

Trac Ticket #153 Change CDMI Mime types once we have an RFC #.

Change vnd.org.snia.cdmi to cdmi in all places according to the mime standard types we get.

See RFC XXXX from IANA (Arnold/Krishna)

CHANGES Made changes throughout document (too many places to reference).

Trac Ticket #154 Document affect of deserializing unsupported features.

Description: Clarify specification wording that implementations should silently ignore vendor extensions and future specification features that they are not coded to understand.

Add the following to Section 15.2 before the Section 15.2.1:

Deserialization operations shall restore all metadata from the specified source. If the original provider of the serialized data supported vendor extensions through custom metadata keys and values, these shall be restored on deserialization but may be treated as user metadata (preserved, but not interpreted) by the destination provider. This allows for the movement of custom data requirements between clouds without loss of this information.

CHANGES Edited the paragraph and added it before 15.2.1 as follows:

Deserialization operations shall restore all metadata from the specified source. If the original provider of the serialized data-supported vendor extensions is through custom metadata keys and values, then these customized requirements shall be restored when deserialized. However, the custom metadata keys and values may be treated as user metadata (preserved, but not interpreted) by the destination provider. Preservation allows custom data requirements to move between clouds without losing this information.

Trac Ticket #155 Add additional HTTP requirements to new Section 5.13.

Section 5.13 was added to get rid of the clobber functions outside of HTTP.

We need to add the additional HTTP requirements inferred from other places in the specification to this new, central location. We either infer or mention that specific parts of the HTTP spec are required to be implemented.

A conformant implementation of CDMI must also be a conformant implementation of RFC 2616 (HTTP 1.1). We call out areas of RFC 2616 that must be supported.

1 Content-Type negotiation

A client may optionally supply an Accept header. If the content type of the requested resource is not present in the header, the server must return a 406 (Not Acceptable) status code. (See Section 12 of RFC 2616.)

2 Range support

The server must support Range headers and partial content responses (see Section 14.16 of RFC 2616).

Mandatory Non-Compliance

- 1 CDMI uses the Content-Type header on body-less GET requests to trigger CDMI behavior. The server must accept a Content-Type header from a request that contains no content. This behavior is not specified by 14.17 but is mandatory for CDMI server implementations.
- 2 Responses can return 206 (Partial Content) for requests that did not include a Range header. This behavior violates 10.2.7.

Note: Section 8.5 of the spec lists no Request Headers, which might be an error in the document. If so, then this item can be removed.)

- 3 Responses can return 204 (No Content) to mean "Data object may not be complete, retry later". This behavior is not specified in 10.2.5.

Implement the following changes:

- 1 Create a new Section 5.13 to contain the following:

A conformant implementation of CDMI must also be a conformant implementation of RFC 2616 (HTTP 1.1). We call out areas of RFC 2616 that must be supported.

Content-Type negotiation

A client may optionally supply an Accept header. If the content type of the requested resource is not present in the header, the server must return a 406 (Not Acceptable) status code. (See Section 12 of RFC 2616.)

Range support

The server must support Range headers and partial content responses (see Section 14.16 of RFC 2616).

- 2 Make two changes to Section 8.5 - Read a Data Object (Non-CDMI Content Type):
 - Add a Request Headers section to document the mandatory header for a 206 response. **(Added)**

Header	Type	Value	Mandatory/Optional
Range	Header String	A valid ranges-specifier (see RFC 2616 Section 14.35.1)	Optional

- Change text in the "204 No Content" line from "Data object may not be complete, retry later" to "Data object exists but has no content". **(Changed)**

- CHANGES**
- 1 Removed all text from 5.13 and replaced it with the following (note the changes in paragraph 1):
 A conformant implementation of CDMI shall also be a conformant implementation of RFC 2616 (HTTP 1.1). This standard lists the sections of RFC 2616 that shall be supported.
 Content-Type Negotiation
 A client may optionally supply an Accept header. If the content type of the requested resource is not present in the header, the server must return a 406 (Not Acceptable) status code. (See Section 12 of RFC 2616.)
 Range Support
 The server must support Range headers and partial content responses (see Section 14.16 of RFC 2616).
 - 2 See changes in **red**.

Trac Ticket #156 Label all tables as tables, add period to any "description" text within tables

Description:

- Please add table caption labels to all tables in the specification that currently do not have them.
- Within the table description field, terminate all text with a period.

Kevin Marks to clarify the requirement for this:

- Do each set of message body tables need a table of contents title? (rather than the existing generic labels) No
- We do not plan to have titles for the examples (grey boxes).

CHANGES Added captions to two tables; added periods to all descriptions in tables for consistency.

Trac Ticket #157 Add consistent RFC 2119 wording to specification.

RFC 2119 reference text "keywords are based on...."

- can -> may
- will -> (rephrase the tense of surrounding words) -> shall
- must -> shall
- "are/is required to" -> shall
- "could" -> may

CHANGES Changed key words as specified throughout document.

Trac Ticket #158 Second sentence of domain definition needs to be moved.

Move second sentence to end of the first paragraph in Section 10.1.

- CHANGES** In 10.1, second paragraph, moved the following sentence to the end of the paragraph:
- “If a cloud storage system supports domains, the *cdmi_domains* container shall be present.”

Trac Ticket #159 Where references are used, provide actual reference link to references section entry.

Kevin to add these and send to Marie.

- CHANGES**
- 1 Created multiple cross-references in Chapters 5, 8-11, 16, and Annex A.
 - 2 Added cross-references for [ISO-8601], [REST], and [RFC216].
 - 3 Added hyperlinks to web addresses in Chapter 2 - References.

Note: There are no references to [RFC2119] or [RFC3986]. Also, ISO-3166 is mentioned in standard but is not in Chapter 2 - References.

Trac Ticket #161 Definition corrections

Marie, please make the following changes to the definitions section:

- 1 Correct spelling of "Ressource" in URI definition.
- 2 Remove XSet and XUID definitions, as they are not used.
- 3 Include hyperlink to SNIA online dictionary for the footnote text "SNIA Online Dictionary".
- 4 Remove "tm" from XAM for its definition.
- 5 Make REST definition match the online SNIA dictionary (please check others as well)

- CHANGES** In Chapter 4 - Terms, made all changes as referenced above; checked all definitions in SNIA dictionary and made appropriate changes.

Trac Ticket #162 Section 5 use of "we"

In the sentences in Section 5 where the word "we" is used, please change to use "the standard" instead (with proper surrounding tense).

- CHANGES** In Chapter 5 - Overview of Cloud Storage, changed all instances of "we". In some instances, replaced "we" with "this standard" and corrected verb tense, and in other places, changed "we" to a passive-voice construction. For example: changed:

In this model, we abstract the underlying storage space exposed by these interfaces using the notion of a container.

to:

In this model, the underlying storage space is abstracted and exposed by these interfaces using the notion of a container.

Trac Ticket #163 Mis-italicized words

The convention of italics is for variables, field names, and book titles. Please find all use of italics that do not conform to the convention and un-italicize them.

- CHANGES**
- 1 Removed italics from words not conforming to convention.
 - 2 Italicized all field names.

Trac Ticket #164 Expand first use of acronyms.

Find the first occurrence in the text of each acronym in the definition section where it appears in the section text. Expand the acronym for that first use and provide a link back to the definition for that first occurrence.

- CHANGES** Made changes throughout document as follows:
- 1 Spelled out the first occurrence of **each acronym in each chapter** and linked the acronym to the definition in Chapter 4 - Terms.
 - 2 Created a link for the first use of **each term in each chapter** and linked the term to the definition in Chapter 4 - Terms.
 - 3 Alphabetized the definitions in Chapter 4 - Terms.

Trac Ticket #165 Figure 6 should show Domain instead of Accounting.

Use updated diagram.

- CHANGES** Updated Figure 6 - CDMI Interface Model.

Trac Ticket #166 Change use of "specification" to "standard."

The document uses the term "specification" in several places. Please substitute the word "standard" in all places.

- CHANGES** Changed "specification" to "standard" throughout document **except** where context would prove usage to be inaccurate. Also did NOT change:
- X-CDMI-*Specification*-Version to X-CDMI-*Standard*-Version

- *cdmi_scope_specification* to *cdmi_scope_standard*.
- *cdmi_query_specification* to *cdmi_query_standard*.

Trac Ticket #167 Change the second to last sentence in Section 5.7.

Change the second to last sentence in Section 5.7 to read:

Typically, cloud implementations will offer a subset of CDMI and describe what is implemented via the capabilities.

CHANGES Replaced second sentence. Paragraph now reads:

The CDMI standard uses RESTful principles in the interface design where possible. Typically, cloud implementations will offer a subset of CDMI and describe what is implemented via the capabilities. For more information on the REST principles, please see [RESTful Web].

Trac Ticket #169 Table 4, Section 7.2 should forward reference the various chapters.

Table 4, Section 7.2 should forward reference the various chapters for further information on each object type. For example, DataObject should forward reference Chapter 8.

CHANGES Added cross-references to Table 4 - Types of Objects in the Model in Section 7.2.

Trac Ticket #170 Section 8.1 - fix sentence.

Change the sentence:

Each data object has a set of well-defined fields that include a single data stream and standardized and optional metadata ...

to:

Each data object has a set of well-defined fields that include a single value and optional metadata ...

CHANGES As indicated, fixed the second sentence in the first paragraph of Section 8.1 Overview.

Trac Ticket #171 Delete Section 8.1.2.

CHANGES Deleted 8.1.2 Data Object Addressing.

Trac Ticket #172 capabilitiesobject+json*Description*

- 1 Section 6.1 refers to a content type of application/vnd.org.snia.cdmi.**capabilitiesobject**+json
Please change to application/vnd.org.snia.cdmi.**capabilities**+json.
- 2 In trac ticket 153, "vnd.org.snia.cdmi" was changed to "cdmi".
Please review this change in errata documents dated June 8, 2010 to see if these changes were applied correctly to the document.
- 3 Please update all occurrences of CDMI MIME types to correspond to <http://tools.ietf.org/search/draft-cdmi-mediatypes-02>
 - application/cdmi-domain
 - application/cdmi-capability
 - application/cdmi-container
 - application/cdmi-object and
 - application/cdmi-queue

- CHANGES**
- 1 Changed application/vnd.org.snia.cdmi.**capabilitiesobject**+json to application/vnd.org.snia.cdmi.**capabilities**+json.
 - 2 In 7.3 and Chapter 8, changed application/vnd.org.snia.cdmi.dataobject+json to application/cdmi-object.
 - 3 Changed the following strings throughout document:
 - application/cdmi.container+json to application/cdmi-container
 - application/cdmi.dataobject+json to application/cdmi-object
 - application/cdmi.queue+json to application/cdmi-queue
 - application/cdmi.capabilities+json to application/cdmi-capabilities
 - application/cdmi.domain+json to application/cdmi-domain

Trac Ticket #174 In all Message Body Tables, change the first column title.

Change the first column title from "Name" to "Field Name".

- CHANGES** Changed "Name" to "Field Name" in all Request Message Body tables and all Response Message Body tables.

Trac Ticket #175 Remove requirement that breaks vendor extensions.

Description: At the end of message body tables, there is a sentence "Requesting fields not defined in the standard result in a 400 Bad Request HTTP status code."

Remove all occurrences of this sentence throughout the specification.

CHANGES Removed this sentence from Chapters 8, 9, 10, 11, and 12 (five occurrences).

Trac Ticket #176 Clarify error messages in CompletionStatus.

Description:

In the spec text, change:

“or an error message” to “or an error string starting with the value "Error"”

CHANGES Changed throughout document. (12 occurrences)

Trac Ticket #177 Clarify purpose of X-CDMI-Partial.

Description: In the spec text, change:

"true". Indicates that the newly created object is part of a series of writes, and has not yet been fully created. When set, the `_completionStatus` field will be set to "Processing".

to

"true". Indicates that the newly created object is part of a series of writes and the value has not yet been fully populated. When set, the `completionStatus` field will be set to "Processing".

CHANGES Changed in the Request Headers table in Section 8.2 - Create a Data Object (CDMI Content Type).

Trac Ticket #178 Clarify fields specified on creation of Reference.

Description: In the spec text, change:

URI of a CDMI data object that will be pointed to by a reference. No other fields may be specified when creating a reference.

to

URI of a CDMI data object that will be pointed to by a reference. If other fields from this table are supplied when creating a reference, the server shall respond with a 400 Bad Request error response.

CHANGES Made change to Request Message Body tables in 8.2, 9.2, and 9.9.

Trac Ticket #179 Clarify error when incompatible fields specified on creation.

Description: In the spec text, change:

*Only one of these parameters shall be specified in any given operation, and except for *value*, these fields are not persisted.

to

*Only one of these parameters shall be specified in any given operation, and except for *value*, these fields are not persisted. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response.

CHANGES Changed paragraph in Section 9.9.

Note: Did **not** make any changes to the following paragraph:

*Only one of these parameters shall be specified in any given operation and is not persisted.

Trac Ticket #180 Clarify 400 error responses.

Description: In the spec Response Status tables, change:

Invalid parameter of field names in the request

to

Invalid parameter **or** field names in the request

CHANGES Made changes in Chapter 8 - 12.

Trac Ticket #181 Clarify field access when reading objects.

Description: In the spec text, for **read** operations, change:

<fieldname> is the name of a field from the response message body.

to

<fieldname> is the name of a field.

CHANGES Made the specified change in 8.4, 8.5, 9.4, 9.5, 10.3, 11.3, and 12.2.

Trac Ticket #182 Missing completionStatus in Examples

Description:

- 1 Section 6.4, Page 31, "percentageComplete" should be "completionStatus"
- 2 Section 6.5, Page 32, "percentComplete" should be "completionStatus"
- 3 Section 16.1.11, Page 153, "percentComplete" should be "completionStatus"

CHANGES Made the specified changes in 6.4, 6.5, and 16.1.11.

Trac Ticket #183 Clarify N/A Sections.

Description: In the spec text, each time a section consists of only "N/A", replace "N/A" with "None specified."

CHANGES Made the specified changes in Chapters 8-12.

Trac Ticket #184 Zeros in gaps between written ranges

Description: In the spec text, change:

Any resulting gaps between ranges should be treated as all zeros, and included when calculating the size of the value.

to

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.

CHANGES Made specified change in 8.6 - Update a Data Object (CDMI Content Type).

Trac Ticket #206 Section 9.1.3 - Change "parameters" to "fields".

Description: Two occurrences in first sentence.

CHANGES Made the specified changes in 9.1.3.

Trac Ticket #207 Section 9.2 - Change sentence.

Description:

From:

GET does not return any value for the object when completionStatus is not "Complete".

to:

GET does not return any children for the container when completionStatus is not "Complete".

CHANGES Changed sentence in Section 9.2.

Trac Ticket #208 Section 8.2, 9.2, 9.9 and 11.2 - Change sentence.

Description:

From:

If the new container object is a reference of an existing container object, support for the ability to create the reference is indicated by the presence of the "cdmi_create_reference" capability in the parent container.

To:

If the object being created in the parent container is a reference, support for that ability is indicated by the presence of the "cdmi_create_reference" capability in the parent container.

CHANGES Changed the sentence in each "Capabilities" Section of 8.2, 9.2, 9.9, and 11.2.

Trac Ticket #209 Section 8.2, 9.2, and 9.9 - Change sentence.

Description:

1 Section 8.2:

In the Request Body Table for the field: *mimetype*

Add:

This field shall not be included when referencing a data object.

2 Section 9.2:

In the Request Body Table for the fields: *metadata* and *exports*

Add:

This field shall not be included when referencing a data object.

3 Section 9.9:

In the Request Body Table for the field *mimetype*

Add:

This field shall not be included when referencing a data object.

Note: This sentence was already included in bullet 1.

Change:

No other fields may be specified when creating a reference.

Note: This sentence isn't found in Section 9.9 in the description column of the Request Message Body table for *reference*. However, it is found in 11.2 - Create a Queue Object.

To:

The *exports* and *metadata* fields shall not be specified when creating a reference.
(No change)

CHANGES Changed sentences in 8.2 and 9.2; no changes in 9.9 (see comments in red).

Trac Ticket #210 Section 8.2, 9.2, 9.9 and 11.2 - Change sentence.*Description:*

Remove the sentence at the bottom of the Request Message Body table:

*Only one of these parameters shall be specified in any given operation, and except for value, these fields are not persisted.

CHANGES None. See Trac Ticket #179.

Trac Ticket #211 Section 9.2 - bullet consistent with 8.2, 9.9 and 11.2*Description:*

See examples in those sections.

CHANGES Made formatting consistent in Sections 8.2, 9.2, 9.9, and 11.2.

Trac Ticket #212 Clarify metadata behaviour when copying/creating objects.*Description:*

1 In the Section 8.2 Request Message Body, the following text is present:

Metadata for the data object.

- If this field is included when deserializing, serializing, copying, or moving a data object, the value provided shall replace the metadata from the source URI.
- This field shall not be included when referencing a data object.
- If this field is not specified, an empty JSON object ("{}") will be assigned as the field value.

This needs to be updated to the below to cover all of the different options, and make it consistent across different object types.

Metadata for the data object.

- If this field is included when deserializing, serializing, copying, or moving a data object, the value provided in this field shall replace the metadata from the source URI.
- If this field is not included when deserializing, serializing, copying, or moving a data object, the metadata from the source URI shall be used.
- If this field is included when creating a new data object by specifying a value, the value provided in this field shall be used as the metadata.
- If this field is not included when creating a new data object by specifying a value, an empty JSON object ("{}") will be assigned as the field value.
- This field shall not be included when referencing a data object.

- 2 In the Section 9.2 Request Message Body, replace the metadata description with: Metadata for the container.
 - If this field is included when deserializing, serializing, copying, or moving a container, the value provided in this field shall replace the metadata from the source URI.
 - If this field is not included when deserializing, serializing, copying, or moving a container, the metadata from the source URI shall be used.
 - If this field is included when creating a new container by specifying a value, the value provided in this field shall be used as the metadata.
 - If this field is not included when creating a new container by specifying a value, an empty JSON object ("{}") will be assigned as the field value.
 - This field shall not be included when referencing a container.
- 3 In the Section 9.9 Request Message Body, replace the metadata description with: Metadata for the data object.
 - If this field is included when deserializing, serializing, copying, or moving a data object, the value provided in this field shall replace the metadata from the source URI.
 - If this field is not included when deserializing, serializing, copying, or moving a data object, the metadata from the source URI shall be used.
 - If this field is included when creating a new data object by specifying a value, the value provided in this field shall be used as the metadata.
 - If this field is not included when creating a new data object by specifying a value, an empty JSON object ("{}") will be assigned as the field value.
 - This field shall not be included when referencing a data object.
- 4 In the Section 11.2 Request Message Body, replace the metadata description with: Metadata for the queue object.
 - If this field is included when deserializing, serializing, copying, or moving a queue object, the value provided in this field shall replace the metadata from the source URI.
 - If this field is not included when deserializing, serializing, copying, or moving a queue object, the metadata from the source URI shall be used.
 - If this field is included when creating a new queue object by specifying a value, the value provided in this field shall be used as the metadata.
 - If this field is not included when creating a new queue object by specifying a value, an empty JSON object ("{}") will be assigned as the field value.
 - This field shall not be included when referencing a queue object.

CHANGES Changed the descriptions listed in **1**, **2**, **3**, and **4** above affecting Sections 8.2, 9.2, 9.9, and 11.2, respectively.

Trac Ticket #213 Split out Query, Logging, and Notifications from Queues chapter.

Description: To improve the readability of the queues section and add prominence to the query, logging, and notifications sections, these sections (11.1.1, 11.1.2, 11.1.3) should be split into separate chapters. The logging section (11.1.2) should be merged into the logging chapter.

- CHANGES**
- 1 Moved 11.1.1 Notification Queues to 19 Notification Queues
 - 2 Moved 11.1.2 Logging Queues to 17.5 Logging Queues
 - 3 Moved 11.1.3 Query Queues to 20 Query Queues.
 - 4 Fixed all cross-references.
 - 5 Updated the Organization section in the Introduction.

Trac Ticket #214 Add "?" display in container listing for References.

Description: In Section 7.3, replace:

References are visible as children in a container and appear identical to non-referenced objects.

with

References are visible as children in a container and are distinguished from non-references by a trailing "?" character added to the reference name.

- CHANGES** Replaced specified sentence in Section 7.3.

Trac Ticket #215 Remove `cdmi_query_domains` from Table 11.

Description: As the *cdmi_query_specification* allows filtering of results based on domain, we should remove the redundant *cdmi_query_domains* row.

- CHANGES** Removed *cdmi_query_domains* from Table 24 - Required Query JSON (Used to be Table 11).

Trac Ticket #216 Clarify use of term "references" to only mean CDMI references.

Description: Replace

This is allowed to create different access security relationships for groups of user objects (via container ACLs) and to allow references to common user lists.

with:

This is allowed to create different access security relationships for groups of user objects (via container ACLs) and to allow delegation to common user lists.

CHANGES Changed “references” to “delegation” in the specified sentence.

Trac Ticket #217 **Table 11 and 12 title should not include "Required".**

Description:

- 1 Table 11 – Required Query JSON
should be
Table 11 - Query JSON
- 2 Table 12 – Required Value of a Query Result Data Object
should be
Table 12 – Query Result JSON

CHANGES

- 1 Changed wording in Tables 11 and 12 (new Tables 24 and 25) with specified wording.
- 2 Added cross-reference to Table 24 and updated cross-reference to Table 25.

Trac Ticket #218 **Fix content type handling for GET.**

Description:

- 1 Drop Content-Type in all CDMI transactions where there is not a Request Message Body. For all of these situations, make Accept mandatory, and add to the description that it must contain a list of one or more of the five CDMI MIME types. This change needs to be made to 8.4, 9.4, 10.3, 11.3, 12.2
- 2 For domains and capabilities, the Accept list shall only be the respective MIME type.
- 3 Update all examples to conform to the above

This statement "Must contain a list of one or more of the five CDMI MIME types." needs to be individually tailored for each section. This is found in both the Domain section (10.3) and the Capabilities section (12.2) when the ticket states that the Accept type shall be specific in these cases (#2 under Description)

- 4 For the Request Headers in 10.3, the "Accept" row should have a value of just "application/cdmi.domain+json".
- 5 For the Request Headers in 11.3, the "Accept" row should have a value of just "application/cdmi.queue+json"
- 6 For the Request Headers in 12.2, the "Accept" row should have a value of just "application/cdmi.capabilities+json"

CHANGES

- 1 Under **Request Headers** in 8.4, 9.4, 10.3, 11.3, and 12.2:
 - Changed the Accept Header from Optional to Mandatory.
 - Added “Must contain a list of one or more of the five CDMI MIME types” to the description of the Accept Header.

- Deleted the Content-Type row.
Under **Response Headers** in 8.4, 9.4, 10.3, 11.3, and 12.2:
- Deleted the Content-Type row.
- 2 No change.
- 3 Deleted Content-Type entries in all examples in 8.4, 9.4, 10.3, 11.3, and 12.2
- 4 10.3 - No change. The "Accept" row already has a value of "application/cdmf.domain+json".
- 5 11.3 - One change in 3rd Example Request. The "Accept" row in all other cases already has a value of "application/cdmf.queue+json"
- 6 12.2 - No change. The "Accept" row already has a value of "application/cdmf.capabilities+json".

Trac Ticket #219 Fix content type handling for DELETE.

Description:

- 1 Collapse 8.8 and 8.9 into one section without a Content-Type or Accept.
- 2 Collapse 9.7 and 9.8 into one section without a Content-Type or Accept.
- 3 Drop Accept from 10.5.
- 4 Drop Content-Type and Accept from 11.5.
- 5 Update all examples to conform to the above.

Why was Content-Type not dropped from 10.5 (Domain Delete) as it was in the other sections?

- 6 Content-Type can also be dropped from 10.5, as per the other sections.

- CHANGES**
- 1 Merged 8.8 and 8.9. Under **Request Headers** in 8.8, deleted the Accept and Content-Type rows.
 - 2 Merged 9.7 and 9.8. Under **Request Headers** in 9.7, deleted the Accept and Content-Type rows.
 - 3 Under **Request Headers** in 10.5, deleted the Accept row.
 - 4 Under **Request Headers** in 11.5, deleted Content-Type and Accept rows.
 - 5 Updated all examples to reflect changes in 8.8, 9.7, 10.5, and 11.5.
 - 6 Deleted Content-Type from Request Headers and Example Request in 10.5 Delete a Domain (CDMI Content Type).

Trac Ticket #220 Clarify that non-CDMI GET for a container is not specified.

Description: Add some text into Section 9.5 that a Non-CDMI GET of the Container URI without requesting fields is not specified in this standard.

Proposal: Just before the Capabilities heading, add the following sentence:

CDMI does not define the format for a GET of a container when fields are not being requested.

CHANGES Added the specified sentence as a note just before the Capabilities heading in Section 9.5 Read a Container Object (Non-CDMI Content Type).

Trac Ticket #221 Table Header updates for 9.1.1

Description:

- 1 The table in Section 9.1.1 should have Mandatory/Optional as last column, with the value of "Optional" for the existing rows.
- 2 The value of the Type column for the existing rows should be "JSON String"

CHANGES In Section 9.1.1 Container Metadata, added "JSON String" to the Type column and replaced the "Permissions" column with "Mandatory/Optional." Added "Optional" under the Mandatory/Optional column.

Trac Ticket #223 Table Header updates for 10.1.1

Description:

- 1 The table in Section 10.1.1 should have Mandatory/Optional as last column, with the value of "Mandatory" for existing rows.
- 2 The value of the Type column for the existing rows should be "JSON String"

CHANGES In Section 10.1.1 Domain Metadata, added "JSON String" to the Type column and replaced the "Permissions" column with "Mandatory/Optional." Added "Mandatory" under the Mandatory/Optional column.

Trac Ticket #224 Updates to table 5 and 6 in Section 10.1.2 (Domain Summaries)

Description:

- 1 Table 5 in Section 10.1.2 should have Mandatory/Optional as last column, with the value of "Mandatory" for all existing rows.
- 2 The value of the Type column should be "JSON String" for all existing rows.
- 3 The contents of Table 6 should be merged into table 5, as "Optional", and with a type of "JSON String".
- 4 The sentence "Any of the metadata listed in Table 6 may also be present within each domain summary object." should be deleted.
- 5 The title of Table 5 should be renamed to "Contents of Domain Summary Objects"

CHANGES Made all specified changes to Table 5 - Contents of Domain Summary Objects in Section 10.1.2 Domain Summaries.

Trac Ticket #225 Set some Domain Summary Object Fields to be optional.

Description:

In table 5 in Section 10.1.2, change the following fields to be mandatory or optional:

- Mandatory: *cdmi_domainURI*, *cdmi_summary_start*, *cdmi_summary_end*
- Optional: *cdmi_summary_objects*, *cdmi_summary_puts*, *cdmi_summary_gets*, *cdmi_summary_bytes*, *cdmi_summary_writes*, *cdmi_summary_reads*

CHANGES Made all specified changes to Table 5 - Contents of Domain Summary Objects in Section 10.1.2 Domain Summaries.

Trac Ticket #226 Change Objects to Object Hours in Domain Summary.

Description:

Add to Table 5 in Section 10.1.2 the following fields:

cdmi_summary_objects is replaced with the following:

- *cdmi_summary_objecthours*, JSON String, The sum of the time each object belonging to the domain existed during the summary time period, Optional
- *cdmi_summary_objectsmin*, JSON String, The minimum number of objects belonging to the domain during the summary time period, Optional
- *cdmi_summary_objectsmax*, JSON String, The maximum number of objects belonging to the domain during the summary time period, Optional
- *cdmi_summary_objectsaverage*, JSON String, The average number of objects belonging to the domain during the summary time period, Optional

CHANGES Made all specified changes to Table 5 - Contents of Domain Summary Objects in Section 10.1.2 Domain Summaries.

Note: Updated the example that follows Table 5, changing “*cdmi_summary_objects*” to “*cdmi_summary_objecthours*”.

Trac Ticket #227 Change Bytes to Byte Hours in Domain Summary.

Description: Add to Table 5 in Section 10.1.2 the following fields:

cdmi_summary_bytes is replaced with the following:

- *cdmi_summary_bytehours*, JSON String, The sum of the time each byte belonging to the domain existed during the summary time period, Optional

- *cdmi_summary_bytesmin*, JSON String, The minimum number of bytes belonging to the domain during the summary time period, Optional
- *cdmi_summary_bytesmax*, JSON String, The maximum number of bytes belonging to the domain during the summary time period, Optional
- *cdmi_summary_bytesaverage*, JSON String, The average number of bytes belonging to the domain during the summary time period, Optional

CHANGES Made all specified changes to Table 5 - Contents of Domain Summary Objects in Section 10.1.2 Domain Summaries.

Note: Updated the example that follows Table 5, changing “*cdmi_summary_bytes*” to “*cdmi_summary_bytehours*”

Trac Ticket #231 Missing capability for *cdmi_logging*

Description: We are missing a capability for indicating if a system supports *cdmi_logging*. We should add the following row to table 7, below the row for “*cdmi_notification*”:

cdmi_logging, JSON String, A JSON string that, if present and “true”, the cloud storage system supports logging queues.

- CHANGES**
- Added above-referenced text to Table 7 - System-Wide Capabilities in Section 12.1.1 Cloud Storage System-Wide Capabilities.
 - Changed “*cdmi_shapshots*” to “*cdmi_snapshots*” in Table 7.

Trac Ticket #233 Fix or remove ACL sample code

Description: During an internal review, it was suggested that we consider testing or removing the ACL sample code in 16.1.7. Specifically, there were questions around the statement

```
(a.flags | CDMI_ACE_FLAGS_INHERIT_ONLY_ACE != 0)
```

and if this would always evaluate to true, as with the line

```
(CDMI_ACE_FLAGS_NO_PROPAGATE_ACE != 0)
```

Delete the entire 16.1.7 section.

CHANGES Deleted 16.1.7 - ACL Evaluation Utilities.

Trac Ticket #234 3DES_ECB_168 in *cdmi_encryption* section

Description:

In the `cdmi_encryption` section of Table 17, the `3DES_ECB_168` entry says that it is a Cipher Block Chaining (CBC) mode of operation. Either the text should be changed to match the constant, or the constant should be changed to match the text.

Replace the following under "`cdmi_encryption`" in Table 24 (Metadata for CDMI Data Copies):

- “`3DES_ECB_168`” for the three-key Triple DES algorithm, the Cipher Block Chaining (CBC) mode of operation, and a key size of 168 bits

with:

- “`3DES_ECB_168`” for the three-key Triple DES algorithm, the Electronic Code Book (ECB) mode of operation, and a key size of 168 bits
- “`3DES_CBC_168`” for the three-key Triple DES algorithm, the Cipher Block Chaining (CBC) mode of operation, and a key size of 168 bits

The change is actually in Table 18 (not Table 24) in the 1.0d spec.

CHANGES Changed text as specified in Table 18 - Data Systems Metadata .

Trac Ticket #235 Garbage text in 10.1.2 Domain Summaries listing

Description: In the listing of domain summaries paths, in section 10.1.2, there is some garbage text ",Ä¶" between each line.

CHANGES Removed ,Ä¶ in three places in Section 10.1.2.

Trac Ticket #236 Add capability for query regex

Description:

On Wednesday's call, we agreed that a capability for supporting regular expressions as part of query should be added.

The capability "`cdmi_query_regex`" should be added to Table 7, under "`cdmi_query`".

`cdmi_query_regex`, JSON String, A JSON string that, if present and "true", the cloud storage system supports query with regular expressions.

Text to resolve this ticket incorporated into the CDMI Query R4 document as edited at the Tuesday face-to-face.

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi.

This text is ready to integrate into the next draft of the standard.

CHANGES Made the specified changes as presented in 2010-11-11 CDMI Query R6.doc.

- Added [POSIC ERE] to Chapter 2 - References.
- Updated Table 8 in 12.1.1 with cdmi query capability.
- Updated 17.5 Logging Queues
- Replaced all text in Chapter 19 Notifications Queues.
- Replaced all text in Chapter 20 Query Queues, adding 20.1 Scope Specification and 20.2 Results Specification.
- Updated all cross-references to new sections/hyperlinks to URLs.

Trac Ticket #237 Updates to Table 6

Description: Table 6 – "Required Settings for Domain Member Objects" needs minor updates to make it consistent with other tables.

- 1 Change the last column heading to "Mandatory/Optional".
- 2 For the Type column, the following items should have the row value of "JSON String":
 - cdmi_member_enabled
 - cdmi_member_type
 - cdmi_member_name
 - cdmi_member_credentials
 - cdmi_member_principal
 - cdmi_delegation_URI
- 3 For the Type column, the following items should have the row value of "JSON Array of JSON Strings":
 - cdmi_member_privileges
 - cdmi_member_groups

CHANGES Made the following changes to Table 6 - Required Settings for Domain Member Objects in Section 10.1.3 Domain Membership:

- Changed "Read/Write" to "Mandatory/Optional" as specified in step 1 and removed contents of this column.
- Added "JSON String" and "JSON Array of JSON Strings" to the referenced types, as specified in steps 2 and 3.

Trac Ticket #238 Split Table 6 into User and and Delegation tables

Description: Table 6 – "Required Settings for Domain Member Objects" should be split into two different tables, one showing user entry and one showing a delegation entry.

- 1 For the "Table 6 – Required Settings for Domain Member User Object", the following fields should be included
 - cdmi_member_enabled (Mandatory)
 - cdmi_member_type (Mandatory)
 - cdmi_member_name (Mandatory)
 - cdmi_member_credentials (Mandatory)
 - cdmi_member_principal (Mandatory)
 - cdmi_member_privileges (Mandatory)
 - cdmi_member_groups (Optional)
- 2 For the "Table 6 – Required Settings for Domain Member Delegation Object", the following fields should be included
 - cdmi_member_enabled (Mandatory)
 - cdmi_member_type (Mandatory)
 - cdmi_delegation_URI (Mandatory)
- 3 The paragraph below the current Table 6 should be dropped:

"Multiple parameters are involved in the above operations, some of which are mandatory and some of which are optional. A conforming implementation shall support the mandatory parameters and may support the optional parameters."
- 4 For the entry for cdmi_delegation_URI, the sentence:

For "delegation" type member records, this field contains the URI of the domain to which the user evaluation should be delegated.

This should be updated to:

This field contains the URI of an external identity resolution provider (such as LDAP or Active Directory), or the URI of a Domain Membership Container.
- 5 When splitting apart the tables, the text "For "user" type member records, this field..." should be replaced with "This field..."

CHANGES Made the following changes:

- Split Table 6 into two tables as indicated in steps 1 and 2 above. After doing this, I changed the table titles, added a introduction paragraph before Table 7, and fixed all cross-references.
- Removed the paragraph as stated in step 3, updated the sentence in step 4, and replaced the specified text in step 5.
- Fixed a typo in the paragraph following the examples, changing "When an transaction" to "When a transaction". Also removed an unnecessary comma in the same paragraph, fourth line.

Trac Ticket #239 Updates to Query Table 22, 23

Description: The last column of Table 22 and 23 should be updated to be Mandatory/Optional.

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #240 Extra text at end of Section 20

Description: An extra bit of text at the end of section 20 was left in when the section was moved out of the Queue chapter.

The heading and text under "Example Request:" on page 162 should be dropped.

CHANGES Deleted Example Request heading and text at the end of Section 20 Query Queues.

Trac Ticket #241 Update to Query Example JSON

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #242 Consider changing "cdmi_returned_results" to a JSON Array

Description: In the reviewed query changes (see ticket #239), "cdmi_returned_results" is a JSON object, and all string JSON entities have empty values. We should consider changing this to JSON Arrays instead of JSON objects.

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #243 Drop "cdmi_results_metadata" from Table 24

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #247 Formatting updates to Table 19*Description:*

Table 19 should be updated to have headings of:

"Metadata Name", "Type", "Description", and "Mandatory/Optional"

All rows should have a "Mandatory/Optional" value of "Optional", and the Type should be extracted from the Description. This is Table 19 – Billed Values of Data Systems Metadata Elements. This should be the same as Table 18. In addition, instead of "Required or Mandatory/Optional" use the same header name as SMI-S (from Marilyn).

CHANGES Made formatting changes to all tables in Chapter 16 Metadata.

Trac Ticket #248 Credentials should be optional when combined with delegation.*Description:*

- 1 In table 6, `cdmi_member_credentials` is mandatory. This does not allow user member entries to be used in combination with delegations.

If `cdmi_member_credentials` is changed to be optional, and the following text is added to the description, user members can be used in combination with delegations:

"If this field is not present, one or more delegations must be present and shall be used to resolve user credentials."

- 2 `cdmi_member_principal` should also be made optional with the below text added: "If this field is not present, one or more delegations must be present and shall be used to resolve the user principal."

CHANGES

- 1 Changed `cdmi_member_credentials` to Optional and added specified text.
- 2 Changed `cdmi_member_principle` to Optional and added specified text.

Trac Ticket #249 Error in Move Example in 8.2

Description: On Page 31, the example to "PUT to the container URI a move from an existing data object" has "copy" in the JSON, instead of "move".

CHANGES Changed "copy" to "move" in the Example Request on page 31.

Trac Ticket #250 Error in Update Examples in 8.7*Description:*

- 1 On Page 44, the example to "PUT to the data object URI to read the value of the data object" should read "PUT to the data object URI to update the value of the data object", and the example to "PUT to the data object URI to read the first ten bytes of the value of the data object" should read "PUT to the data object URI to update the first ten bytes of the value of the data object"
- 2 Also, in the second example, the content lengths don't match the example text. The text should read:
"PUT to the data object URI to update four bytes within of the value of the data object"

CHANGES Made the changes to the update examples in Section 8.7.

Trac Ticket #251 Add capability for query "contains"

Description: Given the complications of being able to do a search on an arbitrary substring of any metadata field, we should add a capability for the "contains" query operator.

The capability "cdmi_query_contains" should be added to Table 7, under "cdmi_query".

cdmi_query_contains, JSON String, A JSON string that, if present and "true", the cloud storage system supports query with the "contains" operator.

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #252 Section 8.6 is missing <fieldname> variant from synopsis

Description:

In section 8.6, Update a Data Object (CDMI Content Type), we are missing a summary of how to update just a single field within an object.

- 1 We need to add the following line:

```
PUT <root URI>/<ContainerName?>/<DataObjectName?>?<fieldname>
```

The remainder of the section already describes the four fields that can be modified and gives examples.

- 2 Trac wiki formatting introduced extra ? characters that have been incorrectly added to the spec.

```
PUT <root URI>/<ContainerName?>/<DataObjectName?>?<fieldname>
```

Text should only have a single ?, between Data Object Name and fieldname.

- CHANGES**
- 1 Added the specified line to the synopsis in Section 8.6, Update a Data Object (CDMI Content Type).
 - 2 Removed extra ? characters.

Trac Ticket #253 Add capability for HTTP transport

Description:

We have a capability for HTTPS transport, but we should also have a capability for HTTP transport, since it is supported, but not required.

Propose:

cdmi_security_http_transport, JSON String, A JSON string that, if present and "true", the cloud storage system supports HTTP communications.

Add this to the system capabilities table 8 in 12.1.1.

- CHANGES** Added the proposed entries to Table 8 - System-wide Capabilities.

Trac Ticket #254 Figure 6 in 5.8 still refers to "Accounting"

Description: The figure 6 in section 5.8 refers to "Accounting", instead of domains. We should rename this bubble to "Domains".

- CHANGES**
- 1 Changed "Accounting" to "Domains".
 - 2 Changed https://<offering>/accounting" to "https://<offering>/domains.
 - 3 Changed https://<offering>/Capabilities" to "https://<offering>/capabilities" (changed to small caps for consistency).
 - 4 Fixed arrows.

Trac Ticket #255 Missing POST to create new Queue

Description: Currently, there is no way to create a Queue by Object ID. We need to add a new section, 9.10, that is the queue equivalent of section 9.8.

Text to resolve this ticket incorporated into the CDMI Queue Create via POST R2 document as edited at the Tuesday face-to-face.

Updated text in 2010-11-10 CDMI Query Create via POST R3.doc.

- CHANGES** Added Section 9.10 Create (POST) a New Queue Object (CDMI Content Type) per the text in 2010-11-10 CDMI Query Create via POST R3.doc.

Trac Ticket #257 Mistake in `cdmi_metadata_maxsize` definition

Description:

The definition should read:

A JSON string that, if present, this capability specifies the maximum size in bytes of each user-defined metadata item supported by the cloud storage system. If absent, there is no limit placed on the size of user-defined metadata items.

(Changed "number" to "size" in the last sentence.) This is in Table 8 of 12.1.1; substitute this text.

CHANGES Changed "number" to "size" in definition of `cdmi_metadata_maxsize` in Table 8 - System-Wide Capabilities.

Trac Ticket #259 Missing "Type" information from Table 20

Description: Table 20 - Required Metadata for a Logging Queue is missing the information in the "Type" column for all entries.

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #260 Missing information in tables 15, 21, 22, and 23

Description: Missing type and/or permissions information in Table 15 - Snapshot Parameter of the Container Update Operation; Table 21 - Required Data for a Notification Queue; Table 22 - Required Metadata for a Query Queue; and Table 23 - Query JSON

- 1 Changes for Table 21, 22 and 23 incorporated into text to resolve this ticket incorporated into the CDMI Query R4 document as edited at the Tuesday face-to-face.
- 2 Table 15 should have a value of "Mandatory".

CHANGES

- 1 Updates for changes in Tables 21, 22, and 23 as per 2010-11-11 CDMI Query R6.doc" document in Kavi. See "Trac Ticket #236" on page 39.
- 2 Updated Table 15 with the value of "Mandatory".

Trac Ticket #263 "Content-Type" Response Header label missing from 9.8

Description: The table listing the response headers for section 9.8 is missing the "Content-Type" label.

CHANGES Added "Accept" to "Content-Type" label.

Trac Ticket #264 Add capability for query tags

Description: At Tuesday's face-to-face meeting, we agreed that a capability for supporting tag-based expressions as part of query should be added.

The capability "cdmi_query_tags" should be added to Table 7, under "cdmi_query".

cdmi_query_tags, JSON String, A JSON string that, if present and "true", the cloud storage system supports query with tag matching expressions.

Query text approved at the face-to-face is in the "2010-11-11 CDMI Query R6.doc" document in Kavi. This text is ready to integrate into the next draft of the standard.

CHANGES All updates made. See "Trac Ticket #236" on page 39.

Trac Ticket #265 Capability for "billingsize", but no corresponding Storage System Metadata

Description: There is a capability describing a *cdmi_billingsize* storage system metadata item, but there is no storage system metadata item with this name.

Remove *cdmi_billingsize* capability row from Capabilities Section Table 9.

CHANGES Removed specified row in Chapter 12 from Table 9 - Capabilities for Storage System Metadata.

Trac Ticket #266 Make definitions section conform to ISO requirements.

Description: See Annex C page 47 of [ISO/IEC Directives, Part 2](#).

CHANGES Changed Chapter 4 - Terms to comply with ISO/IEC Directives, Part 2.

Trac Ticket #267 Incorrect _billed data system metadata

Description: In the Table 19 – Billed Values of Data Systems Metadata Elements, there are three incorrect "Metadata Actual Value" entries that are missing the _billed suffix.

- cdmi_retention_period -> cdmi_retention_period_billed
- cdmi_retention_autodelete -> cdmi_retention_autodelete_billed
- cdmi_hold_id -> cdmi_hold_id_billed

CHANGES Updated Table 19 - Billed Values of Data Systems Metadata with changes.

Trac Ticket #268 Missing Owner/Group Storage System Metadata

Description: We are missing the concept of an owner for an object. We need an additional Storage System Metadata item for this.

Proposed addition to "Table 17 – Storage System Metadata" (add above `cdmi_acl`)

`cdmi_owner`, JSON String, The `cdmi_member_name` of the principal that has owner privileges for the object.

- CHANGES**
- 1 Added specified text above `cdmi_acl` to Table 19 - Storage System Metadata in Section 16.3.
 - 2 Added "Optional" under the Requirement heading.

Trac Ticket #272 Replace "text/json" with "application/json"

Description: All instances in the spec where we have "text/json" needs to be replaced with "application/json", which is the currently approved mime type for JSON. Can do a global search and replace.

CHANGES "text/json" changed to "application/json" throughout document.

Trac Ticket #273 Remove `cdmi_export_fcoe` and `cdmi_export_fc` capabilities

Description: The `cdmi_export_fcoe` and `cdmi_export_fc` capabilities refer to exports that are not documented in the specification. Until such time as we complete these sections, we should remove the capabilities.

CHANGES Removed rows from Table 8 in Chapter 12 Capabilities.

Trac Ticket #274 Clarify QueueValues example

Description: Use the QueueValues? description from the "2010-11-10 CDMI Query Create via POST R3.doc" document in all places where we have examples of how the QueueValue? changes with enqueues and dequeues.

CHANGES Made specified changes.

Trac Ticket #275 Update to queue field JSON type to always be an array.

Description: Currently, the spec has queues either having a single item in the *mimetype*, *valuerange* and *value* fields, or having an array of items. This makes things difficult for a client and should have always been an array of items.

For queues, the type should be changed to:

- *mimetype*, JSON Array of JSON Strings
- *valuerange*, JSON Array of JSON Strings
- *value*, JSON Array of JSON Strings

These changes would need to be made to 11.3 and 11.6

CHANGES Made the specified changes in 11.3 and 11.6.

Trac Ticket #276 Chapter title changes for resource operations

Description: As discussed at the face-to-face meetings last week.

- "8 Data Objects" -> "8 Data Object Resource Operations"
- "9 Container Objects" -> "9 Container Object Resource Operations"
- "10 Domain Objects" -> "10 Domain Object Resource Operations"
- "11 Queue Objects" -> "11 Queue Object Resource Operations"
- "12 Capability Objects" -> "12 Capability Object Resource Operation"

CHANGES Changed the chapter titles as specified and Table 1 - Chapter Contents in the Introduction.

Trac Ticket #277 Updates to complete sanitization

Description: As per discussion at last face-to-face:

Need to add Data System Metadata indicating sanitization method (selected from list in *cdmi_sanitization_method* capability).

Need to add Data System Billed metadata for sanitization.

- 1 Add the following row to Table 20 (page 143) following the entry for *cdmi_throughput*:

Column 1: *cdmi_sanitization_method*

Column 2: JSON String

Column 3: If present, this metadata specifies the sanitization method selected from the list in the *cdmi_sanitization_method* capability list. If absent, objects shall not be securely sanitized.

Column 4: Optional

- 2 Add the following row to Table 21 (page 144) following the entry for `cdmi_throughput_billed`:

Column 1: `cdmi_sanitization_billed`

Column 2: JSON String

Column 3: Contains the sanitization method used.

Column 4: Optional

Agree with change.

We should really define how these are listed within the string - comma delimited, etc?

Choice at face-to-face was to make these a JSON Array. See ticket #319 for this change.

CHANGES Made the specified changes.

Trac Ticket #279 **OpenSSL does not support TLS 1.1/1.2**

Description: I've heard some complaints from open source groups that CDMI mandating TLS 1.1/1.2 excludes OpenSSL from being used.

As discussed at the last face-to-face meeting, we were considering adding some language permitting an earlier version of TLS as long as it still supports the mandated cipher suites.

We will need some updated specification text (in Annex A) to review and for Marie to integrate.

- 1 In Section 2.1, add the following reference:
 - "[RFC2246]
 - IETF RFC 2246. The TLS Protocol Version 1.0 - <http://tools.ietf.org/rfc/rfc2246.txt>"
- 2 In Section A.1, replace the following text:
 - "TLS 1.1 shall be implemented by CDMI entities and TLS 1.2 is strongly encouraged. The use of TLS by CDMI entities is optional, but should be used to protect sensitive data."
 - With:
 - "TLS 1.0 shall be implemented by CDMI entities and a more current version of TLS (for example, v1.1 and v1.2) is strongly encouraged. The use of TLS by CDMI entities is optional, but should be used to protect sensitive data."
- 3 Also in Section A.1, replace the following text:
 - "TLS_RSA_WITH_AES_128_CBC_SHA (mandatory for TLS 1.1/1.2)
 - TLS_RSA_WITH_AES_256_CBC_SHA256 (addresses 112-bit security strength requirements)
 - TLS_RSA_WITH_NULL_SHA (for TLS without encryption)"

With:

- "TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA (mandatory for TLS 1.0)
- TLS_RSA_WITH_AES_128_CBC_SHA (mandatory for TLS 1.1/1.2)
- TLS_RSA_WITH_NULL_SHA (for TLS without encryption)"

CHANGES Made the specified changes to 2.1 and A.1.

Trac Ticket #280 Improve English in Capabilities Descriptions

Description: A comment was made at the last face-to-face that the quality of the English in the Descriptions of capabilities (Table 8, etc) could be improved. For example, consider the below capability description:

"A JSON string that, if present and "true", the cloud storage/ computing system supports CIFS exports."

- 1 Given that we have added the Type column, we can re-word this to something along the lines of:
"If present and "true", the cloud storage/computing system supports CIFS exports."
- 2 Delete "/computing" from the phrase "the cloud storage/computing system" in all occurrences of Table 10 (page 112).
- 3 Insert "indicates that" before the phrase "the cloud storage system" in all occurrences of Table 10 (page 112), Table 11 (page 114), and Table 12 (page 115).

CHANGES Made the specified changes.

Trac Ticket #283 Clarify ValueRange for objects with gaps

Description: When there is an "in progress" object that has not yet been committed, we need to clarify what the value range field should return if you have gaps in the object. Once an object has been committed, the value range should return the entire range of the object, including the gaps. See section 8.6 text under "value" that explains what happens to the size of the object when there are gaps.

My proposal (and what my current code does) is to only return the range starting from 0. So if you have an object with a gap from 0 - 100, the returned range would be "". If there was a gap from 50 - 100, the returned range would be "0 - 49", etc.

After discussion in the group, we accepted David's proposal for value range incremental values. David to supply text for this.

Test is as follows in Section 8.4 - Response Message Body:

valuerange - The range of bytes of the value returned in the value field. If the object value has gaps (due to PUTs with non-contiguous value ranges), the value range will indicate the range to the first gap in the object value. The *cdmi_size*

storage system metadata of the data object will indicate the size of the object including gaps."

CHANGES Made the specified changes to the second Message Response Body in 8.4 Read a Data Object (CDMI Content Type).

Trac Ticket #285 Formatting Problems in Annex B

Description: In Annex B, section B.6.1, in the grey boxes, The tabs are missing when there is an object id.

For example:

```
/document.txt "This" /document.txt/versions/
0000706D001006A1D4534CF0DFDC1289 "This"
```

Should be

```
/document.txt "This" /document.txt/versions/
0000706D001006A1D4534CF0DFDC1289 "This"
```

Specifically, the text in the quotes should be tab justified to the right, and line up.

This problem is found with all of the examples in the grey boxes in B.6.1.

CHANGES Added tabs to align text.

Trac Ticket #287 ACL Changes

Description: Alan has a series of recommended changes to the ACL section:

Here are the changes for the ACL section. I took out all the timestamp stuff, basically, having decided it's too much "how" and not enough "what". Also gone is the INHERITED_ACE flag, which I put there because Microsoft uses it. But it doesn't appear in NFSv4, and after some searching through Microsoft literature, it doesn't appear to serve a useful purpose.

Also added some clarifications.

See the attached PDF document for details.

There is still a little bit of text that refers to the now removed acl timestamp:

- 1 In section 16.1.1, the following text should be removed from the last paragraph:

"Similarly, the date shall be represented as a string in [ISO-8601] point-in-time representation, or as an integer (UNIX timestamp format). Implementations shall be able to convert between these formats as necessary. The preferred representation for storage is the integer format, while the preferred representation for display is via strings."

- 2 Also, in section 16.1.6, we need to re-word the first bullet:
"If the ACL timestamp is nonexistent or is older than that of some parent (see foregoing discussion in previous section), use `get_acl(O, P)` to set the physical ACL equal to the logical ACL."
- 3 In section 16.1.4, in the first grey box, add to the end of the list of flags:
"const CDMI_ACE_FLAGS_INHERITED_ACE = 0x00000100;"

- CHANGES**
- Updated Section 16.1 Access Control per markups in CDMI_Spec_Changes_101207.pdf.
 - Updated Sections 16.1.1, 16.1.16, and 16.1.4 per Items 1-3 above, respectively.

Trac Ticket #288 Missing Notification Event Timestamp

Description: During the edits made to the changes in notification queues during the last face-to-face meeting, the timestamp for notifications enqueued into a notification queue was lost.

- 1 We should add this field back in to Table 24 – Required Data for a Notification Queue
In `cdmi_returned_results`, under `cdmi_event_result`:
"cdmi_event_time - Indicates the time of the event that triggered the notification. The time will be formatted as a ISO-8601 time."
- 2 In the example immediately following it, we should add a line below the line:
"cdmi_event_result" : "",
"cdmi_event_time" : "",
- 3 And, in the example of a notification result JSON object, we should add a line below the line:
"cdmi_event_result" : "200 OK",
"cdmi_event_time" : "2010-11-15T13:12:52.342324", See the attached PDF document for details.

- CHANGES** Made the specified changes to Table 24 - Required Data for Notification Queue and the two examples following the table.

Trac Ticket #289 Clarify source of domain when creating object by ID in `/cdmi_objectid/`

Description: We should add text to clarify the source of domain URI when creating an object by ID using `/cdmi_objectid/`.

- 1 We should add the following line to the "domainURI" section of the Request Message Body in sections 9.8 and 9.10:
"If creating an object by ID using `/cdmi_objectid/`, there is no parent container, so the domain must be specified."

- 2 We should also remove the line "POST <root URI>/cdmi_objectid/" from the synopsis of 9.9, since there is no way to specify a domain when doing a non-CDMI put in this space.

Note: Errors 400 (bad parameters) and 403 (Unauthorized) already exist, thus no new error codes are required.

CHANGES Made the specified changes to the Request Message Body sections of 9.8 and 9.10. Also, removed the specified "POST" line in the synopsis of 9.9.

Trac Ticket #290 Clarify in 5.9 that Metadata must be JSON Strings

Description: Somewhere in the last few versions, we lost the note that says that CDMI only uses JSON strings. Specifically, that JSON numeric types are not used and not supported. We should make this explicit, and section 5.9 seems to be a good place to do so.

Proposed wording change:

"User metadata is arbitrarily-defined metadata that is specified by the CDMI client and attached to objects. The namespace used for user metadata is self-administered (such as using the reverse domain name) and restricted to not beginning with the prefix "cdmi_"."

to:

"User metadata is arbitrarily-defined JSON strings that are specified by the CDMI client and attached to objects. The namespace used for user metadata is self-administered (such as using the reverse domain name) and restricted to not beginning with the prefix "cdmi_"."

CHANGES Made the specified changes in Section 5.9.

Trac Ticket #291 No way for client to send serialized data direct to CDMI Server

Description: In the spec, if a client wished to restore a serialized CDMI Data Object, Container, etc to a CDMI server, there is no way for the client to be able to send this serialized form to the server. As it is specified currently, the serialized form must already exist as a data object.

Proposed change:

- 1 To Request Message Body table in section 8.2, add after the "reference" field:
deserializevalue,
JSON String,
A data object serialized as specified in chapter 15.,
Optional*
- 2 To Request Message Body table in section 9.2, add after the "reference" field:

- deserializevalue,
JSON String,
A container object serialized as specified in chapter 15.,
Optional*
- 3 To Request Message Body table in section 10.2, add after the "copy" field:
deserializevalue,
JSON String,
A domain object serialized as specified in chapter 15.,
Optional*
 - 4 At the end of this Request Message Body table, add "*Only one of these parameters shall be specified in any given operation and is not persisted."
 - 5 To Request Message Body table in section 11.2, add after the "reference" field:
deserializevalue,
JSON String,
A queue object serialized as specified in chapter 15.,
Optional*
 - 6 This should also be added to the request body table of Section 9.8
deserializevalue,
JSON String,
A data object serialized as specified in chapter 15.,
Optional*
 - 7 This should also be added to the request body table of Section 9.10
deserializevalue,
JSON String,
A queue object serialized as specified in chapter 15.,
Optional*

CHANGES Made the specified changes in Chapters 8, 9, 10, and 11.

Trac Ticket #292 Consider numbering examples

Description: If we numbered our transaction examples, it would be easier to refer to them, i.e., "Example Request:" -> "Example 8-1".

CHANGES Created and applied an autonumber style to each example in chapters 7-12. Changed "Example Request" to "Example" in Table 2 - Interface Format Descriptions.

Trac Ticket #293 Specify how numbers in strings are handled for query

Description: When performing a numeric comparison query on metadata, there needs to be a consistent way to interpret string values as numbers in order to provide interoperability.

New text needs to be added to section 20.

Discussion:

Changed 1 month ago by alexmc

Example on page 157; "cdmi_size" : "> 100000" is incorrect as by definition this is a lexicographic, not numeric, test.

Page 160: The greater than matching expression tests if the value of the field is lexicographically greater than a specified constant value. The greater than test is case sensitive. The leading space character after the ">" and before the constant value is not included in the comparison. If the constant value is greater than the value of the field, the condition is considered to be met. If the constant starts with a "+" or "-" sign, the value of the field is considered to be numeric for the purposes of comparison.

However, the specification of the query is ambiguous; consider a request for

"valuerange" : "> -10"

It's not clear whether this is a test for the upper value in a range or a negative number. Although this might be considered an odd test, see Page 159: "Queries may match on any field within an object that a CDMI system is capable of returning as a result of an object GET."

It's also not possible to express adequate queries against timestamps. Affects queries on cdmi_ctime, cdmi_atime and cdmi_mtime.

Does use of a delimiting space after the <test> and before <string> in the query <field> : "<test> <string>" make <string> a malformed JSON string?

Changed 4 weeks ago by dslik

Item 1: Yes, this should be updated.

Item 2: "> -10" means a numeric comparison against a negative constant number. How should we re-word this to be clearer?

Item 3: All time stamps are in ISO8601, which supports lexical comparisons, so we're good there.

Item 4: It's valid JSON.

Changed 4 weeks ago by alexmc

Item 3: Only if timestamps are normalised. Consider UTC timestamps.

Changed 4 weeks ago by dslik

Item 3: ISO 8601 specifies an optional time zone, and is UTC by default.

Changed 6 days ago by dslik

Spec changes required

- 1 Example on page 157;
"cdmi_size" : "> 100000"
should be:
"cdmi_size" : "> +100000"
- 2 For time, change the paragraph in 10.1.2:
"All time and duration, unless otherwise specified, are in the [ISO-8601] extended representation (YYYY- MM-DDThh:mm:ss,sssss). Components of the time representation with finer granularity may be omitted, if desired."
Should be replaced with:
All time and duration values, unless otherwise specified, are in the [ISO-8601] extended representation (YYYY-MM-DDThh:mm:ss.ssssssZ). The full precision must be specified, the sub-second separator must be a ".", the Z UTC zone indicator must be included, and all timestamps must be in UTC time zone.
- 3 Add new section to chapter 5:
5.14 Time Representations
All date/time values, unless otherwise specified, are in the [ISO-8601] extended representation (YYYY-MM-DDThh:mm:ss.ssssssZ). The full precision must be specified, the sub-second separator must be a ".", the Z UTC zone indicator must be included, and all timestamps must be in UTC time zone. The YYYY-MM-DDT24:00:00.000000Z hour must not be used and represented as YYYY-MM-DDT00:00:00.000000Z.
All time durations, unless otherwise specified, are in the [ISO-8601] start date/end date representation (YYYY-MM-DDThh:mm:ss.ssssssZ/YYYY-MM-DDThh:mm:ss.ssssssZ). The end-date must be equal to or later than the start-date. The full precision must be specified, the sub-second separator must be a ".", the Z UTC zone indicator must be included, and all timestamps must be in UTC time zone. The YYYY-MM-DDT24:00:00.000000Z hour must not be used and represented as YYYY-MM-DDT00:00:00.000000Z.
- 4 For cdmi_retention_period in Table 20:
"Contains an [ISO-8601] time interval specifying the period the object is to be protected by retention. This metadata is the time interval (in either an [ISO-8601] date-duration or an [ISO-8601] date-date) during which the object is under retention. Only the duration or end-date may be altered when updated. If an object is under retention, the object may not be deleted and its value may not be altered. After the retention duration has elapsed, the object may be deleted."
Replace with:
"Contains an [ISO-8601] time interval (as described in section 5.14) during which the object is under retention. Only the end-date may be altered when updated. If an object is under retention, the object may not be deleted and its value may not be altered. After the retention date has passed, the object may be deleted."
- 5 For cdmi_RPO in Table 20:
"Contains the largest acceptable duration in time between an update and when the update may be recovered, in [ISO-8601] duration representation. This metadata is used to indicate the desired backup frequency from the primary copy(s) of the data to the secondary copy(s). It is the maximum acceptable duration between a write

to the primary copy and the backup to the secondary copy during which a failure of the primary copy(s) shall result in data loss."

Replace with:

Contains the largest acceptable duration in time between an update and when the update may be recovered, specified in seconds. This metadata is used to indicate the desired backup frequency from the primary copy(s) of the data to the secondary copy(s). It is the maximum acceptable duration between a write to the primary copy and the backup to the secondary copy during which a failure of the primary copy(s) shall result in data loss.

6 For `cdmi_RTO` in Table 20:

"Contains the largest acceptable duration in time to restore data, in [ISO-8601] duration representation. This metadata is used to indicate the desired maximum acceptable duration to restore the primary copy(s) of the data from a secondary backup copy(s)."

Replace with:

"Contains the largest acceptable duration in time to restore data, specified in seconds. This metadata is used to indicate the desired maximum acceptable duration to restore the primary copy(s) of the data from a secondary backup copy(s)."

7 In table 19, for `cdmi_ctime`, `cdmi_atime` and `cdmi_mtime`, replace reference to ISO-8601 with reference to new section 5.14.

8 For `cdmi_retention_period_billed` in Table 21:

"Contains an [ISO-8601] time interval specifying the period the object is protected by retention."

Replace with:

"Contains an [ISO-8601] time interval (as described in section 5.14) specifying the period the object is protected by retention."

9 For `cdmi_RPO_billed` in Table 21:

"Contains the provided duration in time between an update and when the update may be recovered, in [ISO-8601] duration representation."

Replace with:

Contains the provided duration in seconds between an update and when the update may be recovered."

10 For `cdmi_RTO_billed` in Table 21:

"Contains the provided duration in time between an update and when the update may be recovered, in [ISO-8601] duration representation."

Replace with:

"Contains the provided duration in seconds to restore data."

11 In section 17.2, replace the first bullet:

"A timestamp in [ISO-8601] format"

Replace with:

"A timestamp in [ISO-8601] format (See Section 5.14)"

12 In section 19, Table 24, in the `cdmi_returned_results` table row, the below bullet should be updated:

"cdmi_event_time-Indicates the time of the event that triggered the notification. The time will be formatted in ISO-8601 time."

Replace with:

"cdmi_event_time-Indicates the time of the event that triggered the notification. The time will be formatted in ISO-8601 time (See section 5.14)."

CHANGES Made all specified changes.

Trac Ticket #294 Duplicate Domain Enabled field and Domain Enabled Metadata

Description: We have two places where a domain can be enabled or disabled:

- In the domain object metadata item "cdmi_domain_enabled"
- As an "enabled" field in the domain

We should keep only one of these ways. I recommend that we remove the references to the "enabled" field, since we shouldn't be using fields for functions that can be performed via metadata.

Spec changes required:

- 1 Remove "enabled" row from message request body table of 10.2, 10.4
- 2 Remove "enabled" row from message response body table of 10.2, 10.3
- 3 Remove "enabled" JSON string from message response body in example in 10.2, 10.3, 10.4
- 4 Remove second example from 10.4.

CHANGES Made the specified changes.

Trac Ticket #296 cdmi_security_data_integrity capability not referenced in spec

Description:

The capability `cdmi_security_data_integrity` is not referenced from any locations in the spec other than the capabilities table. We should either better indicate what this means, or remove the capability.

- 1 Eric to propose text. This needs to be clarified for the below system capabilities:
 - `cdmi_security_audit`
 - `cdmi_security_data_integrity`
 - `cdmi_security_encryption`
 - `cdmi_security_http_transport`
 - `cdmi_security_https_transport`
 - `cdmi_security_immutability`
 - `cdmi_security_sanitization`

- 2 Clarify Annex A to refer to the HTTP and HTTPS transport capabilities.
- 3 Consider adding `cdmi_security_access_control` that is enabled when ACLs are supported.

<https://www.snia.org/apps/org/workgroup/cloudtwg/download.php/44445/CDMI-Spec-Changes-Trac-Ticket-296-MAC.docx>

I've marked some changes in this version - we need to be consistent between the tables and the description. I have changed the places where it said "shall be set to false" to "shall not be present" - to be consistent with the tables. I would be OK with changing the tables to add the present and false syntax as well. Just desire consistency.

Also, I always thought that the "optional use of https" was on the client's behalf so that he could make objects available via plain old http (subject to ACLs). Let's discuss this in the TWG meeting - maybe we should not allow server implementations to set `cdmi_security_https_transport` to false...

Approved Changes:

- 1 We updated the word document: <https://www.snia.org/apps/org/workgroup/cloudtwg/download.php/44474/CDMI-Spec-Changes-Trac-Ticket-296-TWGfinal.docx> and approved these changes to be applied to the specification.
- 2 In Section A.1, in the first list of bullets, insert the following bullet before "When HTTP over TLS is implemented, then...":
Although HTTP shall be implemented by all CDMI entities, its use is optional.
- 3 Drop from the word document. "When a CDMI implementation disables the use of HTTP to force more secure communications via HTTPS, the system wide capability of `cdmi_security_http_transport` specified in Table 10 of Section 12.1.1, "Cloud Storage System-Wide Capabilities" shall not be present otherwise it should be set to the default value of "true", indicating HTTP is permitted."

Note: We dropped the `cdmi_security_http_transport` capability, so this later part of the bullet no longer makes sense.

CHANGES Made all changes to Chapters 5, 12, 17, and Annex A per this trac ticket and the Word document "CDMI-Spec-Changes-Trac-Ticket-296-TWGfinal.docx".

Trac Ticket #298 Notification Queues section does not discuss event ordering

Description: As a client, for interoperable operations between systems, I need some guidance of how notification queues will handle ordering of events enqueued into a notification. Currently, in section 19, we don't speak to how ordering is handled.

I propose that we indicate that events for a given object shall be enqueued in timestamp order, but there shall be no ordering guarantees between objects.

Specifically, you are ensured that you won't get a delete notification for object X before a read notification for object X, but you could read object X then read object Y, and get the notifications in the opposite order.

Proposed spec text changes:

"Once the notification queue is created, all subsequent matching events after the queue creation time shall result in notification results being enqueued into the queue."

becomes:

"Once the notification queue is created, all subsequent matching events after the queue creation time shall result in notification results being enqueued into the queue. Events for each object are enqueued in timestamp order, but there are no guarantees of order between objects."

Discussion on call was that putting ordering constraints on notification queues may preclude certain implementations, and that it is thus better not to specify ordering constraints. To ensure that clients understand that this means that out-of-order events in a notification queue are permitted, the proposed spec changes are:

Spec changes:

In section 19, replace the paragraph

"Once the notification queue is created, all subsequent matching events after the queue creation time shall result in notification results being enqueued into the queue."

with

"Once the notification queue is created, all subsequent matching events after the queue creation time shall result in notification results being enqueued into the queue. CDMI does not mandate any specific ordering of events, and clients must be able to handle events that arrive out of order."

CHANGES Replaced the specified paragraph in Chapter 19.

Trac Ticket #299 Negative user feedback on "_billed"

Description: When reviewing our approach of naming the set of data system metadata that contains the storage service that is being provided (as contrasted to the storage service being requested) as *nnn_billed* with end users, we have received feedback that this is confusing to groups that are not providing a paid-for managed service or managing chargebacks within an enterprise.

We may wish to replace "_billed" with a term that is more generic, such as "_actual" or "_provided".

Proposal is to change "_billed" to "_provided".

- 1 Change
 - 16.6 Support for Billed Elements
 - to
 - 16.6 Support for Provided Data System Metadata
- 2 Change

Table 21 – Billed Values of Data Systems Metadata Elements
to

Table 21 – Provided Values of Data Systems Metadata Elements
In Table 21, change all instances of "_billed" to "_provided"

3 In section B4, change

The following billed item for data system metadata is defined (as per Section 16.6, "Support for Billed Elements"):

to

The following provided data system metadata item is defined (as per Section 16.6, "Support for Provided Data System Metadata"):

4 In section B4, change all instances of "_billed" to "_provided"

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes in Items 1-4.

Trac Ticket #300 Doc error in draft 1.0.1f

Description:

Pages 56, 57, 59 example return values incorrect and first 3 entries missing trailing "/".
children Names of the children objects in the container. Child containers end with "/".

Sample from page 56,

```
"childrenrange" : "0-4",
"children" : [
  "red",
  "green",
  "yellow",
  "orange/",
  "purple/"
]
```

Page 101, value should be JSON Array of JSON Strings; example is incorrect.

```
1.0HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "value" : "First"
}
```

Page 106, 107 value and mimetype should be JSON Array of JSON Strings; example is incorrect

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

```

Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0
{
  "mimetype" : "text/plain",
  "value" : "This is the Value of this Data Object"
}

```

Page 121, missing mandatory childrenrange on return.

```

1.0HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "capabilities" : {
    "cdmi_domains" : "true",
    "cdmi_export_nfs" : "true",
    "cdmi_export_webdav" : "true",
    "cdmi_export_iscsi" : "true",
    "cdmi_queues" : "true",
    "cdmi_notification" : "true",
    "cdmi_query" : "true",
    "cdmi_xmlrepresentation" : "true",
    "cdmi_metadata_maxsize" : "4096",
    "cdmi_metadata_maxitems" : "1024"
  },
  "children" : [
    "domain/",
    "container/",
    "dataobject/",
    "queue/"
  ]
}

```

Pages 124, 125 use of typographical open and close double quotes on the examples; "exports" instead of "exports"

Page 138 section 16.1.10 JSON Format for ACLs, acettime is specified as a naked JSON number rather than as a JSON string

```

{
  "cdmi_acl" : [
    {
      "acetype" : "0xnn",
      "identifier" : "<user-or-group-name>",
      "aceflags" : "0xnn",
      "acemask" : "0xnn",
      "acettime" : 12345678
    },
    {
      "acetype" : "0xnn",
      "identifier" : "<user-or-group-name>",
      "aceflags" : "0xnn",
      "acemask" : "0xnn",
      "acettime" : 12345678
    }
  ]
}

```



```
    ]
  }
```

Comments:

Red, Yellow and Green are data objects, and thus do not have trailing slashes. All other errors need to be fixed. I will put together updated JSON examples.

Summary of changes needed:

- Queue examples should reflect type of JSON Array of JSON Strings
- Check examples to ensure that containers, domains, and capabilities include mandatory childrenrange
- Remove typographic quotes from examples

Spec Text Changes for Section 11 (Queues)**1 Section 11.3, Example 1**

Response should be changed to (after running it through jsonlint.com):

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/MyContainer/MyQueue",
  "objectID" : "0000706D00101ADEBC119D1BFE98672A",
  "parentURI" : "/MyContainer/",
  "domainURI" : "/cdmi_domains/MyDomain/",
  "capabilitiesURI" : "/cdmi_capabilities/Queue/",
  "completionStatus" : "Complete",
  "metadata" : {

  },
  "queueValues" : "1-2",
  "mimetype" : [
    "text/plain"
  ],
  "valuerange" : [
    "0-19"
  ],
  "value" : [
    "First Enqueued Value"
  ]
}
```

2 Section 11.3, Example 2

Response should be changed to (after running it through jsonlint.com):

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "queueValues" : "1-2",
  "value" : [
    "First Enqueued Value"
  ]
}
```

3 Section 11.3, Example 3

Response should be changed to (after running it through jsonlint.com):

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "value" : [
    "First"
  ]
}
```

4 Section 11.3, Example 4

Response should be changed to (after running it through jsonlint.com):

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "mimetype" : [
    "text/plain",
    "text/plain"
  ],
  "valuerange" : [
    "0-19",
    "0-20"
  ],
  "value" : [
    "First Enqueued Value",
    "Second Enqueued Value"
  ]
}
```

5 Section 11.6, Request Headers

The "Accept" header should be deleted from this table.

6 Section 11.6, Response Headers

This table should be replaced with "None specified."

7 Section 11.6, Example 1

Request should be changed to (after running it through jsonlint.com):

```
POST /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: application/cdmi-queue
X-CDMI-Specification-Version: 1.0
{
  "mimetype" : [
    "text/plain"
  ],
  "value" : [
    "Value to Enqueue"
  ]
}
```

8 Section 11.6, Example 2

Request should be changed to (after running it through jsonlint.com):

```

POST /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0
{
  "copy" : "/MyContainer/MyDataObject"
}

```

9 Section 11.6, Example 3

Title should be change to:

"POST to the queue URI to transfer twenty values from another queue"

Request should be changed to (after running it through jsonlint.com):

```

POST /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0
{
  "move" : "/MyContainer/FirstQueue;values:20"
}

```

10 Section 11.6, Example 4

Request should be changed to (after running it through jsonlint.com):

```

POST /MyContainer/MyQueue
HTTP/1.1 Host: cloud.example.com
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0
{
  "mimetype" : [
    "text/plain",
    "text/plain"
  ],
  "value" : [
    "First",
    "Second"
  ]
}

```

11 Section 11.7, Request Headers

The "Accept" header should be deleted from this table.

12 All of the example responses in section 11.3 should have the following header line placed above the X-CDMI-Specification-Version header line:

Content-Type: application/cdm-queue

Spec Text Changes for missing childrange field

Section 9.5, Example 2

- Response should be changed to (after running it through jsonlint.com):

```

HTTP/1.1 200 OK
Content-Type: application/json
{

```

```

    "childrenrange" : "0-2",
    "children" : [
        "red",
        "green",
        "yellow"
    ]
}

```

- The example on page 121 is correct, as the childrange field was not requested.

Examples in Section 13.3

Examples in section 13.3 need to have typographical quotes replaced with standard double quotes. Marie should search for other instances.

Spec text changes ready for review at next TWG call.

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made all specified changes.

Trac Ticket #305 Missing cdmi_post_queue capability

Description: We describe a cdmi_post_queue capability in section 9.10 but need to add it to the capability section 12.1.5.

- 1 Add to table 14 under cdmi_post_dataobject:
 - cdmi_post_queue
 - JSON String
 - A JSON string that, if present and "true", the container allows a new queue to be added via POST.
- 2 Marie, make sure that the capability description text is consistent with the other capability descriptions.

CHANGES

- 1 Added cdmi_post_capability to Table 14, as specified.
- 2 Checked all capabilities tables and made description text format consistent, i.e., added "indicates that" where appropriate in Tables 10-16 in Chapter 12 - Capability Objects.

Trac Ticket #306 Section 9.8 and 9.9 heading capitalization

Description: "POST" in section 9.8 and 9.9 should be capitalized.

CHANGES Made the specified changes.

Trac Ticket #307 Missing not variant for starts, ends, contains and tag

Description: We are missing the "!" variants for starts, ends, contains, and tag.

Let's discuss if we should put these in (I think we should), if any new capabilities are needed (I think the existing capabilities are sufficient), and then write some spec text to add them.

Alternate approach is to make the "!" generic, such that it inverts the meaning of any operator. Goal is to make it consistent. Will create spec text to propose "!" variants for starts, ends, contains and tag.

Spec Text Changes to Table 28 of Section 20.1 (Query and Notification Scope)

"field" : "starts constant"

The starts with matching expression tests if the field value starts with a specified constant value. The leading space character after the "starts" and before the constant value is not included in the comparison. The starts with test is case sensitive.

If the constant value is equal to the start of the value of the field, the condition is considered to be met.

"field" : "!starts constant"

The not starts with matching expression tests if the field value does not start with a specified constant value. The leading space character after the "!starts" and before the constant value is not included in the comparison. The not starts with test is case sensitive.

If the constant value is not equal to the start of the value of the field, the condition is considered to be met.

"field" : "ends constant"

The ends with matching expression tests if the field value ends with a specified constant value. The leading space character after the "ends" and before the constant value is not included in the comparison. The ends with test is case sensitive.

If the constant value is equal to the end of the value of the field, the condition is considered to be met.

"field" : "!ends constant"

The not ends with matching expression tests if the field value does not end with a specified constant value. The leading space character after the "!ends" and before the constant value is not included in the comparison. The not ends with test is case sensitive.

If the constant value is not equal to the end of the value of the field, the condition is considered to be met.

"field" : "!contains constant"

The not contains matching expression tests if the field value does not contain a specified constant value. The leading space character after the "!contains" and before the constant value is not included in the comparison. The not contains test is case sensitive.

If the constant value is not found as a substring within the value of the field, the condition is considered to be met. The not contains operator is only supported if the "cdmi_query_contains" capability is present.

"field" : "!tag constant"

The not tag matching expression tests if the field value does not contain a specified constant tag value. The leading space character after the "!tag" and before the constant value is not included in the comparison. The not tag test is not case sensitive.

If the constant value is not found as a tag substring within the value of the field, the condition is considered to be met. Tag substrings start at the beginning of the value or a ",", and end at the next "," or the end of the string. Whitespace before and after "," characters shall be stripped for the purpose of comparisons.

Tag matching expressions are only supported if the "cdmi_query_tags" capability is present.

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

On 3/26/11, made one more change; changed "field" : "contains constant" to "field" : "!contains constant".

Trac Ticket #310 Clarify what happens if query metadata changed after create

Description:

When a user creates a query, the metadata associated with the query indicates the scope, results, etc. for the query. We should specify what happens (or doesn't happen) if the user changes the metadata after the query queue has been created. My suggestion is that the original metadata is used for the query, and any subsequent updates are ignored.

Alternate approach is to say that when a query queue is created, the metadata cannot be updated. The problem with allowing the update is that the metadata can become inconsistent with the query results. Or, that second PUT invalidates all results and restarts the query. Recommended approach is to not permit metadata updates on Query and Notification Queues.

DS to create proposed spec text for review.

Spec changes:

1 In Section 19, change

When creating a notification queue, the metadata described in Table 24 shall be provided.

to

When creating a notification queue, the metadata described in Table 24 shall be provided. Once a notification queue has been created, with the exception of `cdmi_queue_type`, the metadata items in this table cannot be altered. `cdmi_queue_type` can only be removed, indicating to the system that the notification queue shall no longer receive notifications and shall be treated as a regular CDMI queue object.

2 In Section 20, change

When creating a query queue, the metadata described in Table 26, "Required Metadata for a Query Queue" shall be provided.

to

When creating a query queue, the metadata described in Table 26, "Required Metadata for a Query Queue" shall be provided. Once a query queue has been created, with the exception of `cdmi_queue_type`, the metadata items in this table cannot be altered. `cdmi_queue_type` can only be removed, indicating to the system that the query queue shall no longer receive query results and shall be treated as a regular CDMI queue object.

We should indicate what is returned by the server if a client attempts to alter these metadata items.

"Attempts to alter metadata in this table will result in an HTTP 403 Forbidden HTTP status code."

1 Merge this into our proposed spec changes:

When creating a query queue, the metadata described in Table 26, "Required Metadata for a Query Queue" shall be provided. Attempts to alter metadata in this table will result in an HTTP 403 Forbidden HTTP status code. Once a query queue has been created, with the exception of `cdmi_queue_type`, the metadata items in this table cannot be altered. `cdmi_queue_type` can only be removed, indicating to the system that the query queue shall no longer receive query results and shall be treated as a regular CDMI queue object.

2 Also merge into Notification paragraph.

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #311 Consider adding triggering user to Notification Results

Description:

A key piece of information associated with notification about events is who performed the event. Every notification has a corresponding `aclname`, and this should be an

optional field that can be requested. To provide this information, it would be useful to add the ability to request the user information to be included as part of the notification results:

Recommend adding to *cdmi_returned_results* Description in table 24:

cdmi_event_user - Indicates the *aclname* of the user who triggered the event. This field would be left empty for system initiated events.

Should be consistent with what we do with logging.

Recommendation is that we need to be clear about what is a "system initiated event". This should be only things that are not a result of any user-initiated CDMI operations or the processing thereof. DS to propose updated spec text and look at the updated logging proposal.

Add to "cdmi_returned_results" row in Table 24:

cdmi_event_user - Indicates the principal (*acl name*) of the user that caused the event that triggered the notification. If the event was triggered by the system, the name will be left as an empty string.

The logging proposal does not speak to the principal of the user that triggers the log entry. It is anticipated that this will be a common field in many log messages, and that the same principal (*acl name*) will be used for this purpose.

CHANGES Made the specified change.

Trac Ticket #313 Remove *cdmi_xmlrepresentation*

Description: We haven't specified it, so we should remove this capability until we have a proposal for how to handle XML representations.

Approved Change:

A JSON string that, if present and "true", the cloud storage system supports XML representations.

to:

This capability is reserved for future use as the xml representation is not defined in this version of CDMI. This capability shall never be present for CDMI 1.0.x

CHANGES Changed specified paragraph in Table 10 - System-Wide Capabilities and removed referenced line in Examples 12-1 and 12-2.

Trac Ticket #314 Missing Domain Serialize/Deserialize Capabilities

Description:

The `cdmi_serialize_domain` and `cdmi_deserialize_domain` capabilities are missing from Table 15. The following two rows should be added to the end of the table:

- `cdmi_serialize_domain`, JSON String, A JSON string that, if present and "true", the domain and all child domains may be serialized.
- `cdmi_deserialize_domain`, JSON String, A JSON string that, if present and "true", the domain and all child domains may be deserialized.

Serialize domain is already specified in section 8.2 and was just omitted from the capabilities table. Deserialize domain is also missing spec text in the 10.2 to allow a domain to be created from a serialized form.

DS to come back with proposed spec text, and will re-review.

Changes to Spec:

1 Add to end of Table 15:

- `cdmi_serialize_domain`, JSON String, A JSON string that, if present and "true", the domain and all child domains may be serialized.
- `cdmi_deserialize_domain`, JSON String, A JSON string that, if present and "true", the domain and all child domains may be deserialized.

2 Add to capabilities bullet list of Section 10.2:

If the new domain is the destination of a deserialize operation, support for the ability to deserialize the source data object serialization of a domain is indicated by the presence of the "`cdmi_deserialize_domain`" capability in the parent domain.

3 Add to the Request Message Body table of Section 10.2, above the `deserializevalue` row:

`deserialize`, JSON String, URI of a serialized CDMI data object that shall be deserialized to create the new domain, including all child objects inside the source serialized data object., Optional*

Spec text changes ready for review at next TWG call.

Discussed at the 2011-03-09 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #315 Incorrect time example

Description: The time example in section 10.1.2 is incorrect:

"YYYY- MM-DDThh:mm:ss,sssss", should be "YYYY- MM-DDThh:mm:ss.sssss" (should be a period separator, not a comma)

CHANGES Made the specified changes.

Trac Ticket #317 Drop Accept header from 11.6

Description: The "Accept" header row should be dropped from the Request Headers table in section 11.6, as there is no response body returned for this transaction. The Accept header also needs to be dropped from the examples in section 11.6.

The Accept header also needs to be dropped from section 11.7, as there is no response body for this transaction. The Accept header also needs to be dropped from the example in section 11.7

The Content-Type header should also be dropped from the request headers in 11.7 (as per other DELETE requests), as there is no request message body.

Spec text changes ready for review at next TWG call.

Spec Changes:

Change should be made to both 11.6 and 11.7, as there is no response body for both of these operations.

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

To clarify:

- 1 For section 11.6, only remove the "Accept" header from the Request Headers table and the examples.
- 2 For section 11.7, remove both the "Accept" and "Content-Type" headers from the Request Headers table and examples.

CHANGES Made the specified changes.

Trac Ticket #318 Drop YAML Serialization

Description: Unless there is strong objection, I would recommend that we drop YAML serialization from the spec.

Spec changes:

- 1 Remove "cdmi_serialization_yaml" capability from table 10.
- 2 In Section 15.2.1, replace bulleted text with:
 - Recursive JSON for the data object, consistent with the rest of CDMI
 - User and data system metadata for each data object/container
 - Data stream contents for each data object and queue
 - Binary data is represented using escaped JSON strings
 - Typing of data elements is consistent with CDMI JSON representations
- 3 Remove section 15.2.2

Discussed at the 2011-03-16 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made specified changes in Chapter 12 and Chapter 15.

Trac Ticket #319 Capabilities with multiple items should be arrays

Description: Capabilities (such as `cdmi_encryption`, `cdmi_sanitization_method`, etc.) that contain a list of values that can be selected should be a JSON array, not a JSON string.

This includes the following capabilities in table 12.1.3:

- `cdmi_encryption`
- `cdmi_value_hash`
- `cdmi_sanitization_method`

Discussed on April 13th TWG call, and decision is to include in 1.0.1i.

CHANGES Made the specified changes.

Trac Ticket #320 Allowed Data System Metadata values should be in Section 16

Description: Move definition of allowed hashes out of `cdmi_value_hash` row in Table 12, section 12, and into the corresponding `cdmi_value_hash` row of Table 20, section 16. This would be consistent with what is done for `cdmi_encryption` in Table 20.

CHANGES Made the specified changes.

Trac Ticket #322 Errors in JSON Serialization example

Description: There are a number of errors in the JSON serialization example in section 15.2.3:

```
"objectURI" : "/MyContainer",
```

Should be

```
"objectURI" : "/MyContainer/",
```

```
"domainURI" : "/cdmi_domains/MyDomain"
```

Should be

```
"domainURI" : "/cdmi_domains/MyDomain/"
```

(two places)

```
"capabilitiesURI" : "/cdmi_capabilities/Container",
```

Should be

```
"capabilitiesURI" : "/cdmi_capabilities/Container/",
```

Should review other examples to ensure that we have a trailing / for parent, domain, and capabilities.

Updated example JSON for section 15.2.3:

```
{
  "objectURI" : "/MyContainer/",
  "objectID" : "0000706D00101ADEBC119D1BFE98672A",
  "parentURI" : "/",
  "domainURI" : "/cdmi_domains/MyDomain/",
  "capabilitiesURI" : "/cdmi_capabilities/Container/",
  "completionStatus" : "Complete",
  "metadata" : {

  },
  "exports" : {
    "OCCE/iSCSI" : {
      "identifier" : "0000706D00101ADEBC119D1BFE98672A",
      "permissions" : "0000706D00107B85BFE6D20B84D603CA"
    },
    "Network/NFSv4" : {
      "identifier" : "/users",
      "permissions" : "domain"
    }
  },
  "childrenrange" : "0-1",
  "children" : [
    {
      "objectURI" : "/MyContainer/MyDataObject.txt",
      "objectID" : "0000706D00101ADEBC119D1BFE98672A",
      "parentURI" : "/MyContainer/",
      "domainURI" : "/cdmi_domains/MyDomain/",
      "capabilitiesURI" : "/cdmi_capabilities/DataObject/",
      "completionStatus" : "Complete",
      "mimetype" : "text/plain",
      "metadata" : {

      },
      "valuerange" : "0-37",
      "value" : "This is the Value of this Data Object"
    },
    {
      "objectURI" : "/MyContainer/MyQueue",
      "objectID" : "0000706D00101ADEBC119D1BFE98672A",
      "parentURI" : "/MyContainer/",
      "domainURI" : "/cdmi_domains/MyDomain/",
      "capabilitiesURI" : "/cdmi_capabilities/Queue/",
      "completionStatus" : "Complete",
      "metadata" : {
```

```

    },
    "queueValues" : "0-1",
    "valuerange" : [
        "0-2",
        "0-4"
    ],
    "value" : [
        "red",
        "blue"
    ]
  }
]
}

```

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made specified changes in Section 15.2.2 (old 15.2.3).

Trac Ticket #324 Examples missing <CR><LF> line after headers

Description: According to HTTP, there should be an empty line after the headers and before the request/response body.

The examples should be updated to include this.

Instructions for Marie:

In every grey example where there are HTTP headers then a request or response body, there needs to be a single empty line between them.

For example:

```

...
X-CDMI-Specification-Version
{
...
}

```

should be

```

...
X-CDMI-Specification-Version

{
...
}

```

For Non-CDMI examples, it is a little more difficult to identify where the headers end and a body begins:

For example, in section 8.5:

```
HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 37
This is the Value of this Data Object
```

Should be

```
HTTP/1.1 200 OK
Content-Type: text/plain
Content-Length: 37

This is the Value of this Data Object
```

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made specified changes throughout the document.

Trac Ticket #325 Extra header in Example 6.7*Description:*

- Remove the following header line from the example in Section 6.7
Accept: application/cdmi-object
- All DELETE examples should not have an Accept OR Content Type request header.

For example:

```
DELETE /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0
```

should be replaced with:

```
DELETE /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
X-CDMI-Specification-Version: 1.0
```

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Checked all DELETE examples: No changes except for example in 6.7, as shown above.

Trac Ticket #326 Add examples to query results showing all metadata/all fields

Description: It would help clarify the scenarios where a client requests all metadata items or all fields in a query results specification, as described in section 20.1

Propose adding two examples:

For all Metadata:

```
{
  "cdmi_results_specification" : {
    "metadata" : ""
  }
}
```

For all fields:

```
{
  "cdmi_results_specification" : ""
}
```

Proposed spec text:

At the end of section "20.2 Results Specification", add two examples:

If a client desires to have all metadata fields returned for each matching object, the `cdmi_results_specification` below shall be used:

```
{
  "cdmi_results_specification" : {
    "metadata" : ""
  }
}
```

If a client desires to have all fields and metadata returned for each matching object, the `cdmi_results_specification` below shall be used:

```
{
  "cdmi_results_specification" : ""
}
```

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made specified changes.

Trac Ticket #327 `cdmi_returned_results` should be `cdmi_results_specification`

Description: There are a few places in the query and notification section where the value `cdmi_returned_results` is incorrectly used. It should be `cdmi_results_specification`.

Proposed changes:

Search all instances (3) of `cdmi_returned_results` and replace with `cdmi_results_specification`.

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made specified changes (3).

Trac Ticket #328 **Domain membership mismatch cdmi_domain_members != cdmi_domain_membership**

Description: In section 10.1.3 Domain Membership the membership container is called "cdmi_domain_members"; however, the example shows "cdmi_domain_membership". Which one is correct?

Document snippet:

If supported, a domain membership container named "cdmi_domain_members" shall be present under each domain. Like any container, the domain summary container may have standard metadata, such as ACLs that permit access to this information to be restricted.

The example is incorrect. It should be "cdmi_domain_members" throughout the spec.

Spec Changes

- 1 In section 10.2, replace all instances of "cdmi_domain_membership" with "cdmi_domain_members".
- 2 In the capabilities table 12.1.6, change "cdmi_domain_membership" to "cdmi_domain_members".

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #329 **Error in domain membership text.**

Description:

In section 10.1.3, second sentence of the third paragraph, replace:

"Like any container, the domain summary container may have standard metadata, such as ACLs that permit access to this information to be restricted."

with

"Like any container, the domain membership container may have standard metadata, such as ACLs that permit access to this information to be restricted."

CHANGES Made the specified changes to Table 10 - System-Wide Capabilities. Removed lines from the two examples.

Trac Ticket #331 Accommodate Systems that don't report paths by default

Description: Some systems may support objects with a path, but the process of computing a full path is expensive. It has been requested that the spec allow such a system to report by OID instead of by path and to allow a client to explicitly request the path when needed in query results and notification results.

- Change #1 provides a mechanism by which a client in a query or notification can explicitly request just the name (along with the parent) of the object be provided in results.
- Change #2 clarifies that the parentURI can be provided by the CDMI storage server to the client either by path or by ID.
- Change #3 clarifies that for certain query and notification scopes, that the path and object ID form of a URI can be interchangeable.
- Change #4 clarifies how a client in a query or notification can explicitly request that the path be provided in results.

As per the e-mail discussion on the mailing list, a refined proposal is:

- 1 For all types of objects, in the response body table, add a new row above "parentURI" to add the following field:
objectName, JSON String, Name of the Object. If the object has a path, the name is the last part of the path. If the object does not have a path, the name of the object is the Object ID.
- 2 To the "parentURI" field description in the response body table, change "URI for the parent object." to "URI for the parent object. If the object has a path, the parentURI is the URI path to the parent object or the URI path to the parent by Object ID. If the object does not have a path, the parentURI is the URI path to the parent by Object ID.
- 3 At the end of section 20.1 on query scope (used for query and notification), add the paragraph:

"When a URI is used as the constant for the equals and not equals operators, either the URI by path or URI by Object ID can be specified and are considered interchangeable. For example, a query to find all objects belonging to a specific domain, the two below query scopes would be considered identical:

```
{
  [
    "domainURI" : "== /cdmi_domains/MyDomain/"
  ]
}
and
{
  [
    "domainURI" : "== /cdmi_objectid/
0000706D0010171EADF15DE7BC0917D3"
  ]
}
```

Likewise, a query to find all objects with a given parent container would have two equivalent forms:

```

{
  [
    "parentURI" : "== /myContainer/"
  ]
}
and
{
  [
    "domainURI" : "== /cdmi_objectid/
0000706D0010E0981215538EE7D19E5E"
  ]
}
"

```

- 4 To the last paragraph at the end of section 20.2, change the paragraph from "For most common use cases, either the Object ID or Object URI should be requested in the results." to "For most common use cases, either the Object ID or Object URI should be requested in the results. If the Object URI is included in the results, if the object has a path, the path should be returned in this field."

Spec changes

- 1 **A)** In every Request and Response body table that includes a row titled "parentURI", add the following table row above the this row:
- objectName,
JSON String,
Name of the Object. If the object has a path, the name should be the last part of the path. Alternatively, or if the object does not have a path and is only accessible by ID, the name shall be Object ID of the object.
If a vendor chooses to always return the Object ID as the name of an object even if one or more paths exists, the parentURI field shall be set to "/cdmi_objectID/", Mandatory.
- 1 **B)** In every example with a "parentURI" JSON field, add an objectName JSON field above it.

Example 1:

```

HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/MyContainer/MyDataObject.txt",
  "objectID" : "0000706D0010B84FAD185C425D8B537E",
  "parentURI" : "/MyContainer/",
  "domainURI" : "/cdmi_domains/MyDomain/",
  "capabilitiesURI" : "/cdmi_capabilities/DataObject/",
  "completionStatus" : "Complete",
  "mimetype" : "text/plain",
  "metadata" : {
    "cdmi_size" : "37"
  },
}

```

```

    "valuerange" : "0-36",
    "value" : "This is the Value of this Data Object"
  }

```

would be changed to:

```

HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/MyContainer/MyDataObject.txt",
  "objectID" : "0000706D0010B84FAD185C425D8B537E",
  "objectName" : "MyDataObject.txt",
  "parentURI" : "/MyContainer/",
  "domainURI" : "/cdmi_domains/MyDomain/",
  "capabilitiesURI" : "/cdmi_capabilities/DataObject/",
  "completionStatus" : "Complete",
  "mimetype" : "text/plain",
  "metadata" : {
    "cdmi_size" : "37"
  },
  "valuerange" : "0-36",
  "value" : "This is the Value of this Data Object"
}

```

The value for the newly added *objectName* fields should be either the last part of the "objectURI", as shown above.

Example 2:

```

HTTP/1.1 200 OK
Content-Type: application/cdmi-capability
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/cdmi_capabilities/",
  "objectID" : "0000706D00100C435125A61B4C289455",
  "parentURI" : "/",
  "capabilities" : {
    "cdmi_domains" : "true",
    "cdmi_queues" : "true",
    "cdmi_notification" : "true",
    "cdmi_query" : "true",
    "cdmi_metadata_maxsize" : "4096",
    "cdmi_metadata_maxitems" : "1024",
    "cdmi_size" : "true",
    "cdmi_list_children" : "true",
    "cdmi_read_metadata" : "true",
    "cdmi_modify_metadata" : "true",
    "cdmi_create_container" : "true",
    "cdmi_delete_container" : "true"
  },
  "childrenrange" : "0-3",
  "children" : [
    "domain/",
    "container/",
    "dataobject/",
    "queue/"
  ]
}

```

would be changed to:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-capability
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/cdmi_capabilities/",
  "objectID" : "0000706D00100C435125A61B4C289455",
  "objectName" : "cdmi_capabilities/",
  "parentURI" : "/",
  "capabilities" : {
    "cdmi_domains" : "true",
    "cdmi_queues" : "true",
    "cdmi_notification" : "true",
    "cdmi_query" : "true",
    "cdmi_metadata_maxsize" : "4096",
    "cdmi_metadata_maxitems" : "1024",
    "cdmi_size" : "true",
    "cdmi_list_children" : "true",
    "cdmi_read_metadata" : "true",
    "cdmi_modify_metadata" : "true",
    "cdmi_create_container" : "true",
    "cdmi_delete_container" : "true"
  },
  "childrenrange" : "0-3",
  "children" : [
    "domain/",
    "container/",
    "dataobject/",
    "queue/"
  ]
}
```

A quick way to check to see if the the modified examples are correct is to check that the objectName + parentURI = objectURI. This is true for all situations except where a named object is accessed by Object ID (and we don't have any examples of this)

1 C) Add an example of when an object with a path is accessed by ID:

In section 8.4, add the below example after the first example in this section:

Example Request:

GET to the data object URI by ID to read all fields of the data object

```
GET /cdmi_objectid/0000706D0010B84FAD185C425D8B537E
HTTP/1.1 Host: cloud.example.com
Accept: application/cdmi-object
X-CDMI-Specification-Version: 1.0
```

The response looks like:

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0
{
  "objectURI" : "/cdmi_objectid/
0000706D0010B84FAD185C425D8B537E",
  "objectID" : "0000706D0010B84FAD185C425D8B537E",
  "objectName" : "MyDataObject?.txt",
```

```

    "parentURI" : "/MyContainer/",
    "domainURI" : "/cdmi_domains/MyDomain/",
    "capabilitiesURI" : "/cdmi_capabilities/DataObject/",
    "completionStatus" : "Complete",
    "mimetype" : "text/plain",
    "metadata" : {
        "cdmi_size" : "37"
    },
    "valuerange" : "0-36",
    "value" : "This is the Value of this Data Object"
}

```

- 2 In each Response body table that includes a row titled "parentURI", change the Description from "URI for the parent object." to "URI for the parent object. If the object has a path, the parentURI should be the URI path to the parent object. Alternatively, or if the object does not have a path and is only accessible by ID, the parentURI shall be set to "/cdmi_objectid/", and the objectName field shall be set to the Object ID of the object."

- 3 At the end of section 20.1 on query scope (used for query and notification), add the paragraph:

"When a URI is used as the constant for the equals and not equals operators against the "objectURI", "parentURI", "domainURI", and "capabilitiesURI" fields, either a URI by path or URI by Object ID can be specified and are considered interchangeable. For example, a query to find all objects belonging to a specific domain, the two below query scopes would be considered identical:

```

[
  {
    "domainURI" : "== /cdmi_domains/MyDomain/"
  }
]

```

and

```

[
  {
    "domainURI" : "== /cdmi_objectid/
0000706D0010171EADF15DE7BC0917D3/"
  }
]

```

Likewise, a query to find all objects with a given parent container would have two equivalent forms:

```

[
  {
    "parentURI" : "== /myContainer/"
  }
]

```

and

```

[
  {
    "parentURI" : "== /cdmi_objectid/
0000706D0010E0981215538EE7D19E5E/"
  }
]
"

```

- 4 To the last paragraph at the end of section 20.2, change the paragraph from:
- For most common use cases, either the Object ID or Object URI should be requested in the results.
- to
- For most common use cases, either the Object ID or Object URI will be requested in the `cdmi_results_specification`. If the Object URI is included, it is up to the implementation to choose when a Object URI by path or an Object URI by ID should be returned, and both are equally valid. If the Object Name and/or parent URI are included, and paths are supported, the implementation shall return the object name and object path, respectively.

Additional spec changes

- Correction for 1a)

`objectName`,
JSON String,
Name of the Object. If the object has a path, the name should be the last part of the path. Alternatively, or if the object does not have a path and is only accessible by ID, the name shall be the Object ID of the object.

If a vendor chooses to always return the Object ID as the name of an object even if one or more paths exists, the `parentURI` field shall be set to `"/cdmi_objectID/"`, Mandatory.
- Correction for 1b)

A quick way to check to see if the the modified examples are correct is to check that the `parentURI + objectName = objectURI`. This is true for all situations except where a named object is accessed by Object ID (and we don't have any examples of this)

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #333 Clarify non-CDMI object and container creation

Description: When creating objects and containers without an explicit `cdmi-object` or `cdmi-container` content type, I would assume that an empty message body is treated as a container create. However, if a content type of `application/pdf` (for example) is specified with an empty message body, is an object created?

We should discuss and update the appropriate spec sections with the results of the discussion

There are two pieces of information that would permit a CDMI server to differentiate between a non-CDMI create of a data object and the non-CDMI create of a container:

- For a container create, the URI ends with a `"/"`.

- For a container create, there is no "Content-Type" header.

For the non-CDMI create of a data object:

- For a data object create, the URI does not ends with a "/"
- For a data object create, there is a "Content-Type" header.

Should add extra sentence to clarify this.

At end of synopsis, add the below paragraph:

"The presence of a trailing slash at the end of the HTTP PUT URI indicates that the object to be created is a container object."

In section 9.3, at the end of the Synopsis section, add the paragraph:

"The presence of a trailing slash at the end of the HTTP PUT URI indicates that the object to be created is a container object, which distinguishes it from a request to create a data object."

Proposed updated change:

"The presence of a trailing slash at the end of the HTTP PUT URI indicates that a container object is being created, and distinguishes it from a request to create a data object."

Discussed on April 13th TWG call, and decision is to include in 1.0.1i.

CHANGES Made the specified changes.

Trac Ticket #337 Missing capability for references

Description: A capability to indicate if a CDMI system supports references should be added to Table 10:

- cdmi_references
- JSON String
- A JSON string that, if present and "true", the cloud storage system supports references.

Discussed at the 2011-03-16 TWG call and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #338 Containers Synopsis and Examples Incorrect

Description: The operation synopsis and examples in the containers chapter contain numerous errors where the trailing "/" is missing. For example, in section 9.4:

```
GET <root URI>/<ContainerName?>/<TheContainerName?>
GET <root URI>/<ContainerName?>/
  <TheContainerName?>?<fieldname>;<fieldname>;...
GET <root URI>/<ContainerName?>/
  <TheContainerName?>?children:<range>;...
GET <root URI>/<ContainerName?>/
  <TheContainerName?>?metadata:<prefix>;...
```

Should be:

```
GET <root URI>/<ContainerName?>/<TheContainerName?>/
GET <root URI>/<ContainerName?>/<TheContainerName?>/
  ?<fieldname>;<fieldname>;...
GET <root URI>/<ContainerName?>/<TheContainerName?>/
  ?children:<range>;...
GET <root URI>/<ContainerName?>/<TheContainerName?>/
  ?metadata:<prefix>;...
```

Need to identify all the places where this error exists and enumerate the required spec changes.

WARNING: Trac will mangle the text due to their wiki-markup feature. Use the text from the notification e-mail instead.

Spec Changes:

1 Change synopsis of Section 8.7 from

```
PUT <root URI>/<ContainerName>/<DataObjectName>/
to
PUT <root URI>/<ContainerName>/<DataObjectName>
```

2 Change synopsis of Section 9.4 from

```
GET <root URI>/<ContainerName>/<TheContainerName>
GET <root URI>/<ContainerName>/
  <TheContainerName?><fieldname>;<fieldname>;...
GET <root URI>/<ContainerName>/
  <TheContainerName?>children:<range>;...
GET <root URI>/<ContainerName>/
  <TheContainerName?>metadata:<prefix>;...
```

to

```
GET <root URI>/<ContainerName>/<TheContainerName>/
GET <root URI>/<ContainerName>/<TheContainerName>/
  ?<fieldname>;<fieldname>;...
GET <root URI>/<ContainerName>/<TheContainerName>/
  ?children:<range>;...
GET <root URI>/<ContainerName>/<TheContainerName>/
  ?metadata:<prefix>;...
```

3 Change synopsis of Section 9.5 from

```
GET <root URI>/<ContainerName>/<TheContainerName?><fieldname>
```



```

GET <root URI>/<ContainerName>/<TheContainerName>?children:<range>
GET <root URI>/<ContainerName>/<TheContainerName>?metadata:<prefix>
to
GET <root URI>/<ContainerName>/<TheContainerName>/?<fieldname>
GET <root URI>/<ContainerName>/<TheContainerName>/?children:<range>
GET <root URI>/<ContainerName>/<TheContainerName>/?metadata:<prefix>

```

4 Change synopsis of Section 9.7 from

```

DELETE <root URI>/<ContainerName>/<TheContainerName>
to
DELETE <root URI>/<ContainerName>/<TheContainerName>/

```

5 Change synopsis of Section 10.2 from

```

PUT <root URI>/cdmi_domains/<DomainName>/<NewDomainName>
to
PUT <root URI>/cdmi_domains/<DomainName>/<NewDomainName>/

```

6 Change synopsis of Section 10.3 from

```

GET <root URI>/cdmi_domain/<DomainName>/<TheDomainName>
GET <root
URI>/cdmi_domain/<DomainName>/
  <TheDomainName>?<fieldname>;<fieldname>;...
GET <root
URI>/cdmi_domain/<DomainName>/<TheDomainName>?children:<range>;...
GET <root
URI>/cdmi_domain/<DomainName>/<TheDomainName>?metadata:<prefix>;...
to
GET <root URI>/cdmi_domain/<DomainName>/<TheDomainName>/
GET <root
URI>/cdmi_domain/<DomainName>/<TheDomainName>/
  ?<fieldname>;<fieldname>;...
GET <root
URI>/cdmi_domain/<DomainName>/<TheDomainName>/?children:<range>;...
GET <root
URI>/cdmi_domain/<DomainName>/<TheDomainName>/
  ?metadata:<prefix>;...

```

7 Change synopsis of Section 10.4 from

```

PUT <root URI>/cmdi_domains/<DomainName>/<TheDomainName>
PUT <root URI>/cmdi_domains/<DomainName>/<TheDomainName>?metadata
to
PUT <root URI>/cmdi_domains/<DomainName>/<TheDomainName>/
PUT <root URI>/cmdi_domains/<DomainName>/<TheDomainName>/?metadata

```

8 Change synopsis of Section 10.5 from

```

DELETE <root URI>/cdmi_domains/<DomainName>/<TheDomainName>
to
DELETE <root URI>/cdmi_domains/<DomainName>/<TheDomainName>/

```

9 Change synopsis of Section 11.2 from

```

PUT <root URI>/<ContainerName>/<QueueName>/
to
PUT <root URI>/<ContainerName>/<QueueName>

```

10 Change synopsis of Section 11.4 from

```

PUT <root URI>/<ContainerName>/<QueueName>/
PUT <root URI>/<ContainerName>/<QueueName>/?metadata
to
PUT <root URI>/<ContainerName>/<QueueName>
PUT <root URI>/<ContainerName>/<QueueName>?metadata

```

11 Change synopsis of Section 12.2 from

```

GET <root URI>/cdmi_capabilities/<Capability>/<TheCapability>
GET <root
URI>/cdmi_capabilities/<Capability>/
  <TheCapability>?<fieldname>;<fieldname>
GET <root
URI>/cdmi_capabilities/<Capability>/
  <TheCapability>?children:{range}
to
GET <root URI>/cdmi_capabilities/<Capability>/<TheCapability>/
GET <root
URI>/cdmi_capabilities/<Capability>/<TheCapability>/
  ?<fieldname>;<fieldname>
GET <root
URI>/cdmi_capabilities/<Capability>/<TheCapability>/
  ?children:{range}

```

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #339 Location header is inconsistent with HTTP/1.1 RFC

Description: RFC 2616 defines the Location header as:

```

Location = "Location" ":" absoluteURI
Where RFC 2396 defines absoluteURI as:
absoluteURI = scheme ":" ( hier_part | opaque_part )

```

The examples in the CDMI spec (with one exception in Section 7.3 Object References) are relative URLs of the form:

```
Location: /MyContainer/MyObject
```

which is forbidden according to the RFC. The expected form of the Location header is:

```
Location: http://cloud.example.com/MyContainer/MyObject
```

Examples should be updated to be a correct URL. Need to create updated spec text.

Spec changes:

- 1 In all Response Header tables where there is a "Location" row, in the description, change "URI" to "URL".

- 2 In the first example of section 9.8, change the Location header from:

```
Location: /MyContainer/0000706D0010B84FAD185C425D8B537E
```

to

```
Location: http://cloud.example.com/MyContainer/
0000706D0010B84FAD185C425D8B537E
```

- 3 In the second example of section 9.8, change the Location header from:

```
Location: "/cdmi_objectid/0000706D0010B84FAD185C425D8B537E
```

to

```
Location: http://cloud.example.com/cdmi_objectid/
0000706D0010B84FAD185C425D8B537E
```

- 4 In the first example of section 9.9, change the Location header from:

```
Location: /MyContainer/0000706D0010B84FAD185C425D8B537E
```

to

```
Location: http://cloud.example.com/MyContainer/
0000706D0010B84FAD185C425D8B537E
```

- 5 In the second example of section 9.9, change the Location header from:

```
Location: "/cdmi_objectid/0000706D0010B84FAD185C425D8B537E
```

to

```
Location: http://cloud.example.com/cdmi_objectid/
0000706D0010B84FAD185C425D8B537E
```

- 6 In the first example of section 9.10, change the Location header from:

```
Location: /MyContainer/0000706D0010B84FAD185C425D8B537E
```

to

```
Location: http://cloud.example.com/MyContainer/
0000706D0010B84FAD185C425D8B537E
```

Discussed at the 2011-03-16 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #340 Should be “undue” instead of “undo”

Description: In section 5.3, third paragraph, replace "undo" with "undue".

"Fortunately, the SNIA Storage Industry Resource Domain Model (see Figure 4) [SIRDM] gives us a way to minimize this complexity and address the need for cloud storage to remain simple. By using the different types of metadata discussed in that model for a cloud storage interface, an interface may be created that allows offerings to meet the requirements of the data without adding undo complexity to the management of that data."

CHANGES Made the specified changes.

Trac Ticket #341

parentURI in 9.8 object create (POST) to object ID URI example is incorrect

Description: In Section 9.8 of draft 1.0f there is an example of creating an object via POST to the URI /cdmi_objectid/ on page 69. The parentURI in the response is "/MyContainer/".

Given that this object was created via a POST to /cdmi_objectid/, it is not a named object, and thus the parent URI should be "/cdmi_objectid/".

Spec changes

Replace Example Request of second example in Section 9.8 with:

```
{
  "objectURI" : "/cdmi_objectid/
0000706D0010B84FAD185C425D8B537E",
  "objectID" : "0000706D0010B84FAD185C425D8B537E",
  "parentURI" : "/cdmi_objectid/",
  "domainURI" : "/cdmi_domains/MyDomain/",
  "capabilitiesURI" : "/cdmi_capabilities/DataObject/",
  "completionStatus" : "Complete",
  "mimetype" : "text/plain",
  "metadata" : {
    }
  }
}
```

Spec text changes ready for review at next TWG call.

Discussed at the 2011-03-09 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the specified changes.

Trac Ticket #342

Add HTTPS TLS security

Description: In section 5.3, third paragraph, replace "undo" with "undue".

CHANGES Made the specified changes.

Trac Ticket #343 Content-Type header omitted from GET responses

Description: The GET container/domain/queue/capability/etc. responses do not list Content-Type as a mandatory header in the HTTP response (although the GET response examples in Section 6 do include Content-Type).

This seems inconsistent with other responses (e.g., for create operations) where a Content-Type header is included in a response that includes a message body.

Content-Type is a required response header for GETs, as per section 14.17 of RFC 2616.

Kevin, would you be able to locate where this is missing from the HTTP response headers section (and any examples), and indicate what spec text changes should be made?

This applies to the following sections:

- 8.4 Read a Data Object (CDMI Content Type)
 - "Content-Type: application/cdmi-object" should be listed as mandatory in the Response Headers table, and added to all example responses in this section.
- 9.4 Read a Container Object (CDMI Content Type)
 - "Content-Type: application/cdmi-container" should be listed as mandatory in the Response Headers table, and added to all example responses in this section.
- 10.3 Read a Domain Object (CDMI Content Type)
 - "Content-Type: application/cdmi-domain" should be listed as mandatory in the Response Headers table, and added to all example responses in this section.
- 11.3 Read a Queue Object (CDMI Content Type)
 - "Content-Type: application/cdmi-queue" should be listed as mandatory in the Response Headers table, and added to all example responses in this section.
- 12.2 Read a Capabilities Object (CDMI Content Type)
 - "Content-Type: application/cdmi-capability" should be listed as mandatory in the Response Headers table, and added to all example responses in this section.
- 20.2 Results Specification
 - The example response is missing the header "Content-Type: application/cdmi-object"

Spec text changes listed above should be sufficient for updating the document.

CHANGES Changed the Response Headers tables and examples in 8.4, 9.4, 10.3, 11.3, 12.2, and 20.2. Changed the examples in same sections: Example 8-4, 8-5, 8-6, and 8.7; Example 9-3, 9-4, and 9-5; 10-2 and 10-3; 11-2, 11-3, 11-4, and 11.5; 12-1, 12-2, and 12-3; and unnumbered example in 12.2.

Trac Ticket #345 Case sensitivity of object ID hex notation is unclear

Description: It is unclear if CDMI systems should treat the hex string notation of a CDMI object ID as caseless in URIs, queries, etc., i.e. where an OID of 00ab.. is equivalent to an OID of 00AB.., or if the canonical string representation of a CDMI object ID is always in upper case hex (as are all examples in the CDMI spec).

If hex object IDs are caseless, that does raise a few questions, e.g. whether a query expression for an objectID field (a rare use case, but possible) should be case insensitive (as query expressions like == and starts are defined as case sensitive).

Should be upper-case. Will draft text to add to spec.

In section 5.11, change:

- The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation should be hex-encoded.

to

- The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation should be upper case hex-encoded.

Note: Alan Y: Prefers case insensitive, does not feel that it places an undue burden on the implementer. This may also simplify client programming for some languages.

Proposed spec change to section 5.11:

- The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation shall be hex-encoded, and shall be case insensitive.

If object ID strings should be treated as case-insensitive, how does that impact a query on objectID? Should implementations treat query expressions (which are defined as case-sensitive) on the objectID field as case-insensitive?

By defining Object ID textual representations as being case insensitive, this implies that for queries involving the objectID field, the constant should be treated as case-insensitive.

Do you think it is worth calling this out in the query section?

Yes, I think it would be worth explicitly noting this in the query section to avoid any ambiguity.

Ticket re-opened for the addition and review of additional spec text to add to the query section.

Add to the end of Section 20.1:

If an Object ID is used in a query scope (such as in the `objectID` field, or as part of a URI in the `objectURI`, `parentURI`, `capabilitiesURI` or `domainURI` fields), the object ID shall be processed such that it is case insensitive.

Revision 2:

If an Object ID is used in a query scope, such as in the `objectID` field, in the `objectName` field when the `objectPath` is set to `"/cdmi_objectid/"`, and as part of a URI compared against the `objectURI`, `parentURI`, `capabilitiesURI` and `domainURI` fields, all object IDs shall be processed such that they are case insensitive.

Discussed at the 2011-03-16 TWG call, and approved for integration into the 1.0.1h version of the spec.

CHANGES Made the following changes in Section 5.11 and Section 20.1:

1 Section 5.11 - changed the following:

The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation should be hex-encoded.

to

The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation shall be hex-encoded and case insensitive.

2 Section 20.1 - Added to end:

If an Object ID is used in a query scope, such as in the *objectID* field, in the *objectName* field, when the `objectPath` is set to `"/cdmi_objectid/"`, and as part of a URI compared against the *objectURI*, *parentURI*, *capabilitiesURI* and *domainURI* fields, all object IDs shall be processed such that they are case insensitive.

Trac Ticket #346 Synopsis and examples should illustrate operations by Object ID

Description: Currently, all of the synopsis and examples show CDMI operations by path. It may be useful to also include operations by Object ID to show that these are also valid ways to perform operations against objects.

We should discuss this further.

Existing text is clear that operations can be performed against the object path and/or object ID.

An alternate option here is to update section 5.10 Object ID:

5.10 Object ID

Every object stored within a CDMI-compliant system shall have a globally unique object identifier (OID) assigned at creation time. The CDMI object ID is a string, but has rules for how it is generated and how it obtains its uniqueness. Each offering that

implements CDMI is able to produce these identifiers without conflicting with other offerings.

Every CDMI system shall allow access to stored objects by object ID, by allowing the object ID to be appended to any provided root container URI. If the data object "MyDataObject?.txt" has an object ID of "00006FFD001001CCE3B2B4F602032653", the below pair of URIs are considered equivalent:

```
http://cloud.example.com/root/MyDataObject.txt
http://cloud.example.com/root/cdmi_objectid/
    00006FFD001001CCE3B2B4F602032653
```

If containers are supported, they shall also be accessible by object ID. If the container "MyContainer?" has an object ID of "00006FFD0010AA33D8CEF9711E0835CA", the below pairs of URIs are considered equivalent:

```
http://cloud.example.com/MyContainer/
http://cloud.example.com/cdmi_objectid/
    00006FFD0010AA33D8CEF9711E0835CA/
http://cloud.example.com/MyContainer/MyDataObject.txt
http://cloud.example.com/cdmi_objectid/
    00006FFD0010AA33D8CEF9711E0835CA/MyDataObject.txt
```

Discussed on the 2011-05-11 TWG call, and approved for addition to the spec.

CHANGES Replaced text in Section 5.10 as specified above.

Trac Ticket #347 Behavior of "move" operation for data, container, and queue objects is unclear.

Description: It is unclear if performing a "move" of an object creates a completely new object (with a new objectID) or simply renames (changes the objectURI) of an existing object.

Under section 8.2 "Create a Data Object" for the "move" operation it states:

URI of a CDMI data object or queue that shall be copied into the new data object, then removing the data object or queue value at the source URI upon the successful completion of the copy.

This implies that a new object is created from a copy of the old object and the old object is deleted, and presumably the new object is assigned a new object ID. If this is a correct interpretation, then under what circumstances is a "cdmi_rename" notification generated?

If this is not a correct interpretation, then the description of the "move" operation for data objects (8.2), containers (9.2), and queues (11.2) needs to be clarified.

Change history

- The existing spec text is unclear. A move for an object should preserve the object ID. We should discuss this at the next TWG call.
- Tejaswini to write proposed spec text changes to clarify this operation.

Proposed spec changes:**1** Section 8.1 create a data object - move

Change

URI of a CDMI data object or queue that shall be copied into the new data object, then removing the data object or queue value at the source URI upon the successful completion of the copy.

to

URI of an existing CDMI Data object (source URI) whose contents shall be moved to create the new data object (destination URI). On successful completion of the "move" operation, the new dataObject exists only at the destination URI. The source data object is removed at the source URI. The ObjectID? of the new dataObject does not change, only the objectURI and objectName (implementor specific) gets changed.

2 Section 9.2 create a container - move

Change

URI of a CDMI container that shall be copied into the new container, including all child objects under the source container., then removing the container at the source URI upon the successful completion of the copy.

to

URI of an existing CDMI container object (source URI) whose contents shall be moved, including all child objects under the source container to create the new container object (destination URI). On successful completion of the "move" operation, the new container object exists only at the destination URI. The container object at the source URI including the child objects will be removed and will not exist. The ObjectID? of the new container Object, including all the child objects does not change , only the objectURI and objectName (implementor specific) gets changed.

3 Section 11.2 create a Queue object

Change

URI of a CDMI queue object that shall be copied into the new queue object. When the copy is successfully completed, the queue object at the source URI is removed.

to

URI of an existing CDMI Queue object (source URI) whose contents shall be moved to create the new Queue object (destination URI). On successful completion of the "move" operation, the new Queue Object exists only at the destination URI. The source Queue object is removed at the source URI. The ObjectID? of the new Queue Object does not change, only the objectURI and objectName (implementor specific) gets changed.

4 Section 19 Table 24 - Required Data for a Notification Queue

Change should be under "Description" for "Metadata Name" value = "cdmi_notification_events"

Change

cdmi_rename - Notifications are generated when an object is renamed.

to

cdmi_rename - Notifications are generated when an object is renamed as part of "move" request message operations.

Two notes (dslik):

- The Object Name would be based on the PUT URI, so it wouldn't be implementor specific.
- We should also note what the system shall do on a failure.

Notes (Tej)

Changed the data object spec description to include the changes suggested by David Slik. Once this is finalised, the description for the rest of the sections can be fixed.

Question: If move or copy operations fail, will any kind of notifications be generated ?

Below is the Change to section 8.1 based on the feedback:

Section 8.1 Create a Data Object

URI of an existing CDMI data object (source URI) whose contents shall be moved to create the new data object (destination URI). On successful completion of the "move" operation, the new data object exists only at the destination URI. The source data object is removed at the source URI. The ObjectID? of the new data object does not change, only the objectURI and objectName (based on the objectName value specified by the PUT URI) gets changed.

If the operation fails, the data object at the source URI is not removed.

Description modified (diff)

For moves across clouds, we need to ensure that the deletion from the source can be performed and is successful.

- 1 Replace description of the "move" row in the Request Message Body table of section 8.2 with:

"URI of an existing local or remote CDMI data object (source URI) that shall be relocated to the URI specified in the PUT. The contents of the object, including the object ID shall be preserved by a move, and the data object at the source URI shall be removed after the data object at the destination has been successfully created.

If there are insufficient permissions to read the data object at the source URI, write the data object at the destination URI, or delete the data object at the source URI, or any of these operations fail, the move shall return a "400 Bad Request" result code, and the source and destination are left unchanged."
- 2 Replace description of the "move" row in the Request Message Body table of section 9.2 with:

"URI of an existing local or remote CDMI data container (source URI) that shall be relocated, along with all child objects, to the URI specified in the PUT. The contents of the container and all children, including the object ID shall be preserved by a move, and the container and all children of the source URI shall be removed after the objects at the destination has been successfully created.

If there are insufficient permissions to read the objects at the source URI, write the objects at the destination URI, or delete the objects at the source URI, or any of these operations fail, the move shall return a "400 Bad Request" result code, and the source and destination are left unchanged."

- 3 Replace description of the "move" row in the Request Message Body table of section 11.2 with:

"URI of an existing local or remote CDMI queue object (source URI) that shall be relocated to the URI specified in the PUT. The contents of the queue object, including the object ID shall be preserved by a move, and the queue object at the source URI shall be removed after the queue object at the destination has been successfully created.

If there are insufficient permissions to read the queue object at the source URI, write the queue object at the destination URI, or delete the queue object at the source URI, or any of these operations fail, the move shall return a "400 Bad Request" result code, and the source and destination are left unchanged."

- 4 Section 19 Table 24 - Required Data for a Notification Queue

Change should be under "Description" for "Metadata Name" value = "cdmi_notification_events"

Change

cdmi_rename - Notifications are generated when an object is renamed.

to

cdmi_rename - Notifications are generated when an object is renamed as part of "move" request message operations.

CHANGES Made the specified changes.

Trac Ticket #349 Possible missing capabilities for named object support

Description: There doesn't appear to be a capability indicating whether a CDMI system supports the creation of named objects.

Or is this intended to be indicated by the "create" vs "post" capabilities? For example:

- cdmi_create_queue: (named) queues can be created using PUT.
- cdmi_post_queue: (unnamed) queues can be created using POST.

So, for example, a CDMI server that supported only unnamed queues would support cdmi_post_queue but not cdmi_create_queue.

If this is the case, the `cdmi_create_{dataobject,queue}` capability descriptions in table 14 should be updated to indicate that these capabilities allow a new object to be added *via PUT*.

* * *

Notes on the proposed spec text changes:

It turns out that we can define a default ACL in section 16.1.6 because the owner is already known. I've added text to this section instead of making the specification of the `cdmi_acl` metadata mandatory.

1 In Section 9.9.1 Synopsis, replace:

"After it is created, the data object shall also be accessible at `<root URI>/cdmi_objectid/<objectID>`."

with

"If created in `"/cdmi_objectid/"`, the data object shall be accessible at `<root URI>/cdmi_objectid/<objectID>`.

If created in a container, the data object shall be accessible as a child of the container with a server-assigned name, and shall also be accessible at `<root URI>/cdmi_objectid/<objectID>`."

2 In Section 9.9.3 Capabilities, replace the section with:

9.9.3 Capabilities

The following capabilities describe the supported operations that may be performed when creating a new data object by ID in `"/cdmi_objectid/"`:

- Support for the ability to create data objects through this operation is indicated by the presence of the `"cdmi_post_dataobject_by_ID"` system capability.
- If the object being created in `"/cdmi_objectid/"` is a reference, support for that ability is indicated by the presence of the `"cdmi_create_reference_by_ID"` system capability.
- If the new data object being created in `"/cdmi_objectid/"` is a copy of an existing data object, support for the ability to copy is indicated by the presence of the `"cdmi_copy_dataobject_by_ID"` system capability.
- If the new data object being created in `"/cdmi_objectid/"` is the destination of a move, support for the ability to move the data object to `"/cdmi_objectid/"` is indicated by the presence of the `"cdmi_object_move_to_ID"` system capability.
- If the new data object being created in `"/cdmi_objectid/"` is the destination of a deserialization operation, support for the ability to deserialize the data object is indicated by the presence of the `"cdmi_deserialize_dataobject_by_ID"` system capability.
- If the new data object being created in `"/cdmi_objectid/"` is the destination of a serialize operation, support for the ability to serialize the data object is indicated by the presence of the `"cdmi_serialize_dataobject_to_ID"`, `"cdmi_serialize_container_to_ID"`, `"cdmi_serialize_domain_to_ID"`, or `"cdmi_serialize_queue_to_ID"` system capabilities.

The following capabilities describe the supported operations that may be performed when creating a new data object by ID in a container:

- Support for the ability to create data objects through this operation is indicated by both the presence of the "cdmi_post_dataobject" and the presence of the "cdmi_create_dataobject" capability in the specified container object.
- If the object being created in the parent container object is a reference, support for that ability is indicated by the presence of the "cdmi_create_reference" capability in the parent container object.
- If the new data object is a copy of an existing data object, support for the ability to copy is indicated by the presence of the "cdmi_copy_dataobject" capability in the parent container object.
- If the new data object is the destination of a move, support for the ability to move the data object is indicated by the presence of the "cdmi_move_dataobject" capability in the parent container object.
- If the new data object is the destination of a deserialize operation, support for the ability to deserialize the the data object is indicated by the presence of the "cdmi_deserialize_dataobject" capability in the parent container object.
- If the new data object is the destination of a serialize operation, support for the ability to serialize the source data object is indicated by the presence of the "cdmi_serialize_dataobject", "cdmi_serialize_container", "cdmi_serialize_domain", or "cdmi_serialize_queue" capabilities in the parent container object.

3 In Section 9.9.9 Examples, replace Example 2 with:

EXAMPLE 2 POST to the object ID URI the data object contents:

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

{
  "mimetype": "text/plain",
  "domainURI": "/cdmi_domains/MyDomain/",
  "value": "This is the Value of this Data Object"
}
```

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": "text/plain",
  "metadata": {
    "cdmi_acl": [
      {
```

```

        "acetype": "CDMI_ACE_ACCESS_ALLOWED_TYPE",
        "identifier": "OWNER@",
        "aceflags": "CDMI_ACE_FLAGS_NONE",
        "acemask": "CDMI_ACE_ALL"
    }
  ]
}

```

- 4 In Section 9.10.1 Synopsis, at the end of the subsection, add:

"The data object shall be accessible as a child of the container with a server-assigned name, and shall also be accessible at <root URI>/cdmi_objectid/<objectID>."
- 5 In Section 9.11.1 Synopsis, replace:

"After it is created, the queue object shall also be accessible at <root URI>/cdmi_objectid/<objectID>."

with

"If created in "/cdmi_objectid/", the queue object shall be accessible at <root URI>/cdmi_objectid/<objectID>."

If created in a container, the queue object shall be accessible as a child of the container with a server-assigned name, and shall also be accessible at <root URI>/cdmi_objectid/<objectID>."
- 6 In Section 9.9.3 Capabilities, replace the section with:

9.9.3 Capabilities

The following capabilities describe the supported operations that may be performed when creating a new queue object by ID in "/cdmi_objectid/":

 - Support for the ability to create queue objects through this operation is indicated by the presence of the "cdmi_post_queueobject_by_ID" system capability.
 - If the object being created in "/cdmi_objectid/" is a reference, support for that ability is indicated by the presence of the "cdmi_create_reference_by_ID" system capability.
 - If the new queue object being created in "/cdmi_objectid/" is a copy of an existing queue object, support for the ability to copy is indicated by the presence of the "cdmi_copy_queueobject_by_ID" system capability.
 - If the new queue object being created in "/cdmi_objectid/" is the destination of a move, support for the ability to move the data object to "/cdmi_objectid/" is indicated by the presence of the "cdmi_object_move_to_ID" system capability.
 - If the new queue object being created in "/cdmi_objectid/" is the destination of a deserialization operation, support for the ability to deserialize the data object is indicated by the presence of the "cdmi_deserialize_queueobject_by_ID" system capability.

The following capabilities describe the supported operations that may be performed when creating a new data object by ID in a container:

 - Support for the ability to create queue objects through this operation is indicated by both the presence of the "cdmi_post_queueobject" and the presence of the "cdmi_create_queueobject" capability in the specified container object.

- If the object being created in the parent container object is a reference, support for that ability is indicated by the presence of the "cdmi_create_reference" capability in the parent container object.
- If the new queue object is a copy of an existing queue object, support for the ability to copy is indicated by the presence of the "cdmi_copy_queueobject" capability in the parent container object.
- If the new queue object is the destination of a move, support for the ability to move the queue object is indicated by the presence of the "cdmi_move_queueobject" capability in the parent container object.
- If the new queue object is the destination of a deserialize operation, support for the ability to deserialize the the queue object is indicated by the presence of the "cdmi_deserialize_queueobject" capability in the parent container object.

7 In Section "12.1.1 Cloud Storage System-Wide Capabilities", Table 102, add the following rows:

cdmi_post_dataobject_by_ID	JSON String	If present and "true", this capability indicates that the system allows a new data object by ID to be added via POST to "/cdmi_objectid/".
cdmi_post_queue_by_ID	JSON String	If present and "true", this capability indicates that the system allows a new queue object by ID to be added via POST to "/cdmi_objectid/".
cdmi_deserialize_dataobject_by_ID	JSON String	If present and "true", this capability indicates that the system allows the deserialization of serialized data objects when creating a new data object by ID via POST to "/cdmi_objectid/".
cdmi_deserialize_queue_by_ID	JSON String	If present and "true", this capability indicates that the system allows the deserialization of serialized queue objects when creating a new queue object by ID via POST to "/cdmi_objectid/".
cdmi_serialize_dataobject_to_ID	JSON String	If present and "true", this capability indicates that the system allows the serialization of data objects when creating a new data object by ID via POST to "/cdmi_objectid/".
cdmi_serialize_domainobject_to_ID	JSON String	If present and "true", this capability indicates that the system allows the serialization of domain objects when creating a new data object by ID via POST to "/cdmi_objectid/".
cdmi_serialize_containerobject_to_ID	JSON String	If present and "true", this capability indicates that the system allows the serialization of container objects when creating a new data object by ID via POST to "/cdmi_objectid/".

cdmi_serialize_queueobject_to_ID	JSON String	If present and "true", this capability indicates that the system allows the serialization of queue objects when creating a new data object by ID via POST to "/cdmi_objectid/".
cdmi_copy_dataobject_by_ID	JSON String	If present and "true", this capability indicates that the system allows the copying an existing data object when creating a new data object by ID via POST to "/cdmi_objectid/".
cdmi_copy_queue_by_ID	JSON String	If present and "true", this capability indicates that the system allows the copying an existing queue object when creating a new queue object by ID via POST to "/cdmi_objectid/".
cdmi_create_reference_by_ID	JSON String	If present and "true", this capability indicates that the system allows the creation of a new reference by ID a new child reference to be created via POST to "/cdmi_objectid/".

8 Add to the end of section "16.1.6 ACL Evaluation":

When an object is created, no ACL is supplied, and an ACL is not inherited from the parent container (or there is no parent container), the server shall place an ACL containing the following ACEs on the object:

```
"cdmi_acl":
[
  {
    "acetype": "CDMI_ACE_ACCESS_ALLOWED_TYPE",
    "identifier": "OWNER@",
    "aceflags": "CDMI_ACE_FLAGS_NONE",
    "acemask": "CDMI_ACE_ALL"
  }
]
```

TWG: Above changes approved, with the modification below, and the following update ACL used in the updated example 2 in 9.9.9:

Add to the end of section "16.1.6 ACL Evaluation":

When an object is created, no ACL is supplied, and an ACL is not inherited from the parent container (or there is no parent container), the server shall place an ACL containing the following ACEs on the object:

```
"cdmi_acl":
[
  {
    "acetype": "ALLOW",
    "identifier": "OWNER@",
    "aceflags": "OBJECT_INHERIT, CONTAINER_INHERIT",
    "acemask": "ALL_PERMS"
  }
]--
```


CHANGES Made the specified changes to 1.0.1m.

I think there are some problems with these changes as defined above and applied in 1.0.1m:

- David's list of changes above specifies changes to "9.9.3 Capabilities" (data objects) twice -- I think the second set of changes was actually intended for "9.11.3 Capabilities" (queue objects). Currently 9.9.3 is now incorrectly listing the queue capabilities for the data object operations, while 9.11.3 is missing the new queue capabilities.
- Most of the queue capabilities referenced in (what should have been) 9.11.3 do not match the capability names defined in 12.1. This includes "cdmi_post_queueobject_by_ID", "cdmi_copy_queueobject_by_ID", "cdmi_deserialize_queueobject_by_ID", "cdmi_post_queueobject", "cdmi_create_queueobject", "cdmi_copy_queueobject", "cdmi_move_queueobject" and "cdmi_deserialize_queueobject", which are defined as "cdmi_post_queue_by_ID", etc. in 12.1.
- The new serialization to ID capabilities for non-data objects should also not have "object" in their names for consistency with the existing non-ID serialization capabilities. "cdmi_serialize_domainobject_to_ID", "cdmi_serialize_containerobject_to_ID", "cdmi_serialize_queueobject_to_ID" should be "cdmi_serialize_domain_to_ID", etc.

CHANGES Made the changes to Version 1.0.1n as specified in the first bullet, as these were already approved by the TWG.

TWG: Change capabilities that refer to actions that can be performed against a queue to say "queue" instead of "queueobject".

- 1 In clause 9.9.3 and 9.9.5, replace all instances of "_queueobject" with "_queue".
- 2 In clause 12.1.1, replace "cdmi_serialize_queueobject_to_ID" with "cdmi_serialize_queue_to_ID"

(kjamieson) - In clause 12.1.1, I think you also need to:

- 1 * replace "cdmi_serialize_containerobject_to_ID" with "cdmi_serialize_container_to_ID"
- 2 * replace "cdmi_serialize_domainobject_to_ID" with "cdmi_serialize_domain_to_ID"

For consistency with the cdmi_serialize_container and cdmi_serialize_domain capabilities, respectively.

Good catch.

Marie: Please incorporate Kevin's fixes above as well as the changes approved by the TWG.

CHANGES Made the specified changes to 1.0.1n.

Trac Ticket #351 Use of "vendor" rather than more inclusive "implementor"

Description: Much text refers to "vendor", including major sections. Can an alternative and more encompassing term (such as "implementor") be used?

Assign to Marie for inclusion in 1.0.1h.

CHANGES Changed "vendor" to "implementor".

Trac Ticket #353 Font size issue in parentURI table row

Description: In the Description column of the newly added parentURI row, there are two different sizes of text.

This should be ready for Marie to fix.

CHANGES Fixed font in parentURI row in Response Message Body tables.

Trac Ticket #354 Example 8-5 Error

Description: In response body, replace:

```
"objectName" : "MyDataObject?.txt" with "objectName" :
  "MyDataObject.txt",
```

CHANGES Removed "?" in Example 8-5.

Trac Ticket #355 Example 8-14 Error

Description: Missing blank line between request headers and request body:

Replace:

```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain
Content-Length: 37
This is the value of this data object
```

with

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

This is the value of this data object

CHANGES Added blank line where indicated.

Trac Ticket #356 Example 8-15 Error

Description: Missing blank line between request headers and request body/fix request body.

Replace:

```
PUT /MyContainer/MyDataObject.txt?mimetype HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain
Content-Length: 41
This is the new value of this data object
with
PUT /MyContainer/MyDataObject.txt?mimetype HTTP/1.1
Host: cloud.example.com
Content-Type: application/json
Content-Length: 41
```

```
{
  "mimetype" : "application/json"
}
```

CHANGES Replaced indicated text.

Trac Ticket #357 Example 9-1 Error

Description:

Replace in response body:

```
"objectURI" : "/MyContainer",
```

with

```
"objectURI" : "/MyContainer/",
```

Replace in response body:

```
"objectName" : "MyContainer?",
```

with

```
"objectName" : "MyContainer/",
```

CHANGES Made the specified changes.

Trac Ticket #358 Example 9-2 Error

Description:

Replace in HTTP Request:

```
PUT /MyContainer HTTP/1.1
```

with

```
PUT /MyContainer/ HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #359 Example 9-4 Error

Description:

Replace in HTTP request:

```
GET /MyContainer?parentURI;children HTTP/1.1
```

with

```
GET /MyContainer/parentURI;children HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #360 Example 9-5 Error

Description:

Replace in HTTP request:

```
GET /MyContainer?childrenrange;children:0-2 HTTP/1.1
```

with

```
GET /MyContainer/?childrenrange;children:0-2 HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #361 Example 9-6 Error

Description:

Replace in HTTP request:

```
GET /MyContainer?parentURI HTTP/1.1
```

with

```
GET /MyContainer/?parentURI HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #362 Example 9-7 Error

Description:

Replace in HTTP request:

```
GET /MyContainer?childrenrange;children:0-2 HTTP/1.1
```

with

```
GET /MyContainer/?childrenrange;children:0-2 HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #363 Example 9-8 Error

Description:

Replace in HTTP request:

```
PUT /MyContainer HTTP/1.1
```

with

```
PUT /MyContainer/ HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #364 Example 9-9 Error

Description:

Replace in HTTP request:

```
PUT /MyContainer?exports HTTP/1.1
```

with

```
PUT /MyContainer/?exports HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #365 Example 9-10 Error*Description:*

Replace in HTTP request:

```
PUT /MyContainer?exports HTTP/1.1
```

with

```
PUT /MyContainer/?exports HTTP/1.1
```

CHANGES Replaced indicated text.**Trac Ticket #366** Example 9-13 Error*Description:*

Add empty line between request headers and request body:

Replace:

```
POST /MyContainer/ HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain
<object contents>
```

with

```
POST /MyContainer/ HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain

<object contents>
```

CHANGES Replaced indicated text.**Trac Ticket #367** Example 9-14 Error*Description:*

Add empty line between request headers and request body:

Replace:

```
POST /cdmi_objectid/ HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain
<object contents>
```

with

```
POST /cdmi_objectid/ HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain

<object contents>
```

CHANGES Replaced indicated text.

Trac Ticket #368 Remove "+json" from MIME Types

Description: In the document, the text "+json" should be removed from Mime types, Content-Types and Accept headers, etc, to be consistent with the proposed RFC for CDMI Mime Types.

CHANGES Removed "+json" from MIME types, Content-Types, and Accept headers.

Trac Ticket #369 Font size issues in examples

Description: Some of the examples (eg, 7-1) have different font sizes for the request and response sections. We should ensure that all of the examples use the same font size.

CHANGES Checked all examples; changed font in Example 7-1 to match others (9 pt.).

Trac Ticket #370 Don't replace "Vendor extensions"

Description:

In 1.0.1h, "Vendor extensions" was unintentionally changed to "Implementor extensions". It should be reverted back to "Vendor extensions". Same with "Implementor extension" -> "Vendor extension"

CHANGES Replaced indicated text.

Trac Ticket #371 A.3.1, reword to eliminate awkward double "implement"

Description: In section A.3.1, Replace:

CDMI implementations shall implement the TLS protocol; however, its use by clients is optional.

with

CDMI servers shall implement the TLS protocol; however, its use by clients is optional.

CHANGES Made the specified changes.

Trac Ticket #372 20.3 - Awkward sentences (implementor implementing)

Description: As a consequence of changing "vendor" to "implementor", we have some awkward sentences that we should fix in section 20.3:

Replace:

An implementor implementing a CDMI offering may extend CDMI query by adding implementor-specific matching expressions. When an implementor adds implementor-specific metadata fields, these fields shall be queried using the standard query queue functionality.

An implementor implementing a CDMI offering may extend CDMI query by allowing the creation of implementor-specific query queues with a type other than "cdmi_query_queue".

with:

An implementor of a CDMI server may extend CDMI query by adding vendor-specific matching expressions. When an implementor adds vendor-specific metadata fields, these fields shall be queried using the standard query queue functionality.

An implementor of a CDMI server may extend CDMI query by allowing the creation of vendor-specific query queues with a type other than "cdmi_query_queue".

CHANGES Made the specified changes.

Trac Ticket #373 Errors in Sanitization Data System Metadata Entries

Description:

- 1 Table 20: The `cdmi_sanitization_method` row in Table 20 should be of type "JSON String", not "JSON Array".
- 2 Table 21: Change "`cdmi_sanitization_provided`" to "`cdmi_sanitization_method_provided`" to be consistent with the naming conventions.
- 3 This row in Table 21 should be of type "JSON String", not "JSON Array".

CHANGES Made the specified changes.

Trac Ticket #374 15.2.2 example JSON formatting

Description: The example JSON in section "15.2.2 Example JSON Canonical Serialized Format" needs to be run through JSONlint to fix formatting.

CHANGES Ran code through JSONlint; replaced text.

Trac Ticket #375 Example 12-3 Error*Description:*

In HTTP Request, replace:

```
GET: /cdmi_capabilities?childrenrange;children:0-1 HTTP/1.1
```

with

```
GET: /cdmi_capabilities/?childrenrange;children:0-1 HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #376 Example 12-2 Error

Description: In HTTP request, replace:

```
GET: /cdmi_capabilities?capabilities;children HTTP/1.1
```

with

```
GET: /cdmi_capabilities/?capabilities;children HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #377 Example 12-1 Error

Description: In HTTP Request, replace:

```
GET: /cdmi_capabilities?childrenrange;children:0-1 HTTP/1.1
```

with

```
GET: /cdmi_capabilities/?childrenrange;children:0-1 HTTP/1.1
```

CHANGES Replaced indicated text.

Trac Ticket #378 Capabilities for deserialize are unclear

Description: The text for the `cdmi_deserialize_*` capabilities are unclear. For example, for containers, it reads, "If present and "true", indicates that the container and all children's contents may be deserialized.", but it should mean that the container or domain permits deserialize operations to be performed against it, as deserialization is always done into "something". Because deserialization capability is a property of the container, domain, or queue that the deserialization is performed into, we should update the descriptions of the capabilities:

Proposed alternate text for TWG review:

- 1 In Table 14, change the description of "cdmi_deserialize_container" to:
"If present and "true", indicates that the container permits the deserialization of serialized containers and associated serialized children into the container."
- 2 In Table 15, change the description of "cdmi_deserialize_domain" to:
"If present and "true", indicates that the domain permits the deserialization of serialized domains and associated serialized children into the domain."
- 3 Move "cdmi_deserialize_queue" from Table 16 to Table 14, and change the description to:
"If present and "true", indicates that the container permits the deserialization of serialized queues into the container."
- 4 Move "cdmi_deserialize_dataobject" from Table 13 to Table 14, and change the description to:
"If present and "true", indicates that the container permits the deserialization of serialized data objects into the container."

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Made the specified changes.

Trac Ticket #380 Example 11-6 Error

Description: In example 11-6, the Accept header needs to be removed since there is no response body.

Remove: Accept: application/cdmi-queue

CHANGES Removed indicated text.

Trac Ticket #381 Font and Size error in Section 11.4 Synopsis

Description: Section 11.4 has different fonts and sizes used within the synopsis.

CHANGES Fixed all font errors throughout all Synopsis sections.

Trac Ticket #382 Example 10-7 Error

Description: Example is missing the "Content-Type" header, as a request body is provided.

- 1 Add above the X-CDMI-Specification-Version header:

```
Content-Type: application/cdmi-domain
```

2 Also in this example, change:

```
DELETE /cdmi_domains/MyDomain
to
DELETE /cdmi_domains/MyDomain/
```

CHANGES Added/replaced indicated text.

Trac Ticket #383 Remove Example 10-6

Description: We removed the "enable" field from Domains several revisions back. This example was missed.

The example should be removed.

CHANGES Deleted Example 10-6.

Trac Ticket #384 Example 10-5 Error

Description: Remove the Accept header from this example, as there is no response body.

1 Remove:

```
Accept: application/cdmi-domain
```

2 Also, in this example, change:

```
PUT /cdmi_domains/myDomain HTTP/1.1
to
PUT /cdmi_domains/myDomain/ HTTP/1.1
```

CHANGES Removed/changed indicated text.

Trac Ticket #385 Replace "cmdi" with "cdmi"

Description: Search and replace to change "cmdi" into "cdmi".

CHANGES Replaced indicated text.

Trac Ticket #386 Example 10-3 Error

Description: Change:

```
GET /MyDomain?childrenrange;children:0-1 HTTP/1.1
```

to
 GET /MyDomain/?childrenrange;children:0-1 HTTP/1.1

CHANGES Replaced indicated text.

Trac Ticket #387 Example 10-4 Error

Description: Change

GET /MyDomain?parentURI;children HTTP/1.1
 to
 GET /MyDomain/?parentURI;children HTTP/1.1

CHANGES Replaced indicated text.

Trac Ticket #388 Example 10-2 Error

Description: Change

GET /MyDomain HTTP/1.1
 to
 GET /MyDomain/ HTTP/1.1

CHANGES Replaced indicated text.

Trac Ticket #389 Example 10-1 Error

Description: Change

PUT /cdmi_domains/MyDomain HTTP/1.1
 to
 PUT /cdmi_domains/MyDomain/ HTTP/1.1

CHANGES Replaced indicated text.

Trac Ticket #390 Incorrect URIs in Diagram in Section 5.8

Description: The diagram in Section 5.8 should have the following URIs:

"https://<offering>" -> "https://<offering>/"
 "https://<offering>/containerA" -> "https://<offering>/containerA/"
 "https://<offering>/containerB" -> "https://<offering>/containerB/"

```
"https://<offering>/domains" -> "https://<offering>/cdmi_domains/"
"https://<offering>/capabilities" -> "https://<offering>/
cdmi_capabilities/"
```

CHANGES Replaced indicated text. Changed “dataobject” to “dataobject” (twice) and “<containerB>” to “containerB”.

Trac Ticket #391 Example 7-1 Font Size inconsistency

Description: The size of the font in the response is different from the size of the font in the request. We should make them consistent.

CHANGES Duplicate ticket (see “Trac Ticket #369” on page 111).

Trac Ticket #393 Fix incorrect description of cdmi_value_hash data system metadata

Description: In the revision 1.0.1g, the text for the cdmi_value_hash data system metadata item is:

This metadata is used to indicate if the object data is to be hashed and indicates the desired hash algorithm and length. Supported algorithm/length values are provided by the cdmi_value_hash capability.

It was changed to:

If present, this capability lists the hash algorithm/ lengths supported. If absent, objects shall not present a hash value as system metadata. Values are in the form of "Algorithm Length", for example, "SHA256". When a CDMI implementation supports hashing, the system-wide capability of “cdmi_security_data_integrity” specified in Table 10 of Section 12.1.1, "Cloud Storage System-Wide Capabilities" shall be set to “true”. Otherwise, it shall not be present, indicating that there is no hashing support.

The description of this data system metadata item should be changed back to:

This metadata is used to indicate if the object data is to be hashed and indicates the desired hash algorithm and length. Supported algorithm/length values are provided by the cdmi_value_hash capability.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Changed the description data system metadata item as specified.

Trac Ticket #394 Example 6.4 Error

Description:

1 Replace:

```
GET /MyContainer HTTP/1.1
```

with

```
GET /MyContainer/ HTTP/1.1
```

2 The Content-Type header should be removed. Replace it with the following line:

```
Accept: */*
```

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Made the specified changes.

Trac Ticket #395 Example 6.5 Error

Description:

Replace the request header:

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

with

```
Accept: application/cdmi-object
```

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Replaced request header in Section 6.5.

Trac Ticket #396 Remove Example 8-15

Description: This example should be removed. Make this the last update, as it will renumber the remaining examples.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Removed Example 8-15.

Trac Ticket #397 Example 9-12 Error

Description: In the example responsel the JSON is shifted one tab-space over to the left. This should be re-aligned.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Fixed code formatting.

Trac Ticket #398 Incorrect MIME Types (".", should be "-")

Description: There are various places in the spec where "." is used as a MIME separator instead of "-".

For example, in section 9.10, the Request Headers table has "application/cdmi.queue". This should be "application/cdmi-queue".

The documented was reviewed and all instances of application/cdmi.something needs to be replaced with application/cdmi-something. Adobe Reader will permit you to search for where these errors are.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Fixed MIME types by replacing "application/cdmi." with "application/cdmi-".

Trac Ticket #399 9.10 Response Headers Table row for Location extra space

Description: In the Response Headers table in section 9.10, in the row for Location, there is an extra space in the URI.

Replace:

```
<root URL>/ <ContainerName?>/<ObjectID>.
```

with

```
<root URL>/<ContainerName?>/<ObjectID>.
```

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Fixed the URI by removing the space.

Trac Ticket #400 Fix 11.4 Synopsis Font Issue

Description: The font in the synopsis under section 11.4 is not consistent.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Corrected the font.

Trac Ticket #401 GET transactions should not have Content-Type header in request headers

Description: Remove the "Content-Type" line from the **request headers** for GET transactions (Not from the response headers).

- 1 To fix this, find each example where the first line starts with GET. In the grey box with the GET statement, remove any lines that start with "Content-Type".

Example:

```
GET /MyContainer/MyDataObject.txt?valuerange;value:0-10 HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0
```

- 2 Remove the "Content-Type" line.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Removed the "Content-Type" line from the request headers of GET transactions.

Trac Ticket #402 Remove "GET:" from examples

Description: "GET:" should be replaced with "GET" in examples 12-1, 12-2 and 12-3.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Replaced "GET:" with "GET".

Trac Ticket #403 Restore removed "Contains" operator to Table 28

Description: During the last edits, the "Contains" query operation was removed. It should be added back in above the "!Contains" row.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Made specified changes.

Trac Ticket #404 Fix example JSON to be enqueued in query/notification

Description: In section 20.2, in the example box following "If an object matched, the result JSON enqueued is as follows:"

Replace:


```
[
  {
    "objectID" : "0000706D0010B84FAD185C425D8B537E",
    "metadata" : {
      "cdmi_size" : "108263"
    }
  }
]
```

with

```
{
  "objectID" : "0000706D0010B84FAD185C425D8B537E",
  "metadata" : {
    "cdmi_size" : "108263"
  }
}
```

WARNING: Track has mangled the JSON - Run through JSONLint first.

Discussed at face-to-face on May 28th, and approved for inclusion in the 1.0.1h public release.

CHANGES Made specified changes after running code through JSONLint.

Trac Ticket #405 Inconsistent Implementor/Implementer

Description: Need to standardize on one spelling of Implementor/Implementer. Marie to decide which one to use.

CHANGES Changed implementer to implementor.

Trac Ticket #406 Inconsistent RFC URLs

Description: The RFC URLs are inconsistent -- some link to tools.ietf.org and some to www.ietf.org. For consistency, www.ietf.org should be used for all RFC URLs.

CHANGES Made the specified changes to three URLs.

Trac Ticket #407 Corrections to References Section

Description:

- Section 2.1, ISO-8601: "ISO 8601:20044" should be "ISO 8601:2004"
- Section 2.1, ITU-T509: "May 2000" should be "March 2000"

- Section 2.1, RFC3986: "http://www.rfc-editor.org/rfc/rfc3986.txt" should be "http://www.ietf.org/rfc/rfc3986.txt" (the PDF hyperlink goes to the latter, which doesn't match the text)
- Section 2.1, RFC3986: RFC 3986 does not appear to be specifically referenced anywhere in the text.
- Section 2.2, PKS12: Remove redundant ":80" from the URL

Discussed on April 13th TWG call, and decision is to include in 1.0.1i.

CHANGES Made the specified changes to the References Section; removed RFC3986, as it is not referenced anywhere in the spec.

Trac Ticket #408 We should add the SNIA Dictionary to the references section.

Discussed on April 13th TWG call, and decision is to include in 1.0.1i.

CHANGES Added the reference to References and Terms.

Trac Ticket #419 Clarification on behavior of updating metadata and exports

Description: During a discussion at the face-to-face meeting, it became apparent that the text describing expected system behavior when updating metadata and exports was not clear.

The proposed change that was discussed and approved at the meeting is:

- In sections 8.6, 9.6, 10.4, and 11.4, replace:
 "Metadata for the data object. If present, this replaces the existing metadata. See Chapter 16, "Metadata" for a further description of metadata."
 with
 "Metadata for the data object. Existing metadata name/value pairs will be replaced if an identically named name/value pair is provided. Metadata name/value pairs that are not already present will be added. See Chapter 16, "Metadata" for a further description of metadata."
- In sections 9.6, replace:
 "A structure for each protocol that is enabled for this container. If an exported protocol is added or altered, the PUT operation only returns after the export operation has completed."
 With
 "A structure for each protocol that is enabled for this container. Existing export objects will be replaced if an identically named export object is provided. Export objects that are not already present will be added. To disable an export, an empty object shall be provided. If an exported protocol is added or altered, the PUT operation only returns after the export operation has completed."

Alternate approach discussed. This would involve:

- 1 PUT /dataobject would replace the entity with new values, including replacing all metadata (if present in the request body).
- 2 PUT /dataobject?metadata would replace all metadata.
- 3 (new) PUT /dataobject?metadata:<elementname> would replace just a single metadata item, preserving other metadata items
- 4 (new) PUT /dataobject?metadata:<elementname1>;metadata:<name2>;... would replace multiple metadata items, preserving metadata items not included in the list of values.

This can be combined with the existing field update syntax:

- 5 PUT /dataobject?mimetype;metadata:<elementname>

Similar syntax would be added for all CDMI objects.

The behaviors for 1 and 2 would be clarified to make it more clear in the spec.

We currently support addressing only the first level of elementnames in the metadata, so only these could be updated individually.

For example:

```
{
  "metadata" : {
    "a" : "test",
    "b" : {
      "c" : "CDMI",
      "d" : "Spec"
    }
  }
}
```

"a" could be replaced individually, "b" could be replaced individually (which would replace both "c" and "d", but "c" or "d" could not be replaced individually.

Next steps:

- Propose spec text for review
- Consider adding example to section 5.

Spec changes:

- 1 In sections 8.6, add to the synopsis:

```
PUT <root URI>/<ContainerName>/
  <DataObjectName>?metadata:<metadataname>;...
```

- 2 In sections 9.6, add to the synopsis:

```
PUT <root URI>/<ContainerName>/<TheContainerName>/
  ?metadata:<metadataname>;...
```

- 3 In sections 10.4, add to the synopsis:

```
PUT <root URI>/cdmi_domains/<DomainName>/<TheDomainName>/
  ?metadata:<metadataname>;...
```

- 4 In sections 11.4, add to the synopsis:

```
PUT <root URI>/<ContainerName>/
    <QueueName>?metadata:<metadataname>;...
```

- 5 In sections 8.6, 9.6, 10.4, and 11.4, replace:

"Metadata for the data object. If present, this replaces the existing metadata. See Chapter 16, "Metadata" for a further description of metadata."

with

"Metadata for the data object. If present, the new metadata specified replaces the existing object metadata. If individual metadata items are specified in the URI, only those items are replaced, with other items being preserved.

See Chapter 16, "Metadata" for a further description of metadata."

- 6 Add the following example to section 8:

Example 8-??

PUT to the data object URI to replace all metadata with new metadata:

```
PUT /MyContainer/MyDataObject.txt?metadata HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0
{
  "metadata" : {
    "colour" : "red",
    "number" : "7"
  }
}
```

The response looks like:

```
HTTP/1.1 200 OK
```

- 7 Modify the request body of example 8-11 to be:

```
{
  "mimetype" : "text/plain",
  "metadata" : {
    "colour" : "blue",
    "length" : "10"
  },
  "value" : "This is the Value of this Data Object"
}
```

- 8 Add the following example to section 8:

Example 8-??

PUT to the data object URI to add a new metadata item while preserving existing metadata:

```
PUT /MyContainer/MyDataObject.txt?metadata:shape HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0
{
  "metadata" : {
```

```

        "shape" : "round"
    }
}

```

The response looks like:

```
HTTP/1.1 200 OK
```

9 Add the following example to section 8:

Example 8-??

PUT to the data object URI to replace just one metadata item with a new value:

```

PUT /MyContainer/MyDataObject.txt?metadata:colour HTTP/1.1
Host: cloud.example.com
Accept: application/cdm-object
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0
{
    "metadata" : {
        "colour" : "green"
    }
}

```

The response looks like:

```
HTTP/1.1 200 OK
```

Discussed on the 2011-05-11 TWG call, and approved for addition to the spec.

Note: Added examples should go in section 8.6, not 8.8.

CHANGES Made the specified changes to items 1-9 as outlined above, except minor changes to #5. In the replacement text, in 9.6, changed “data” to “container”, in 10.4, changed “data” to “domain”, and in 11.4, changed “data” to “queue”.

Trac Ticket #422 Add text that geographic placement restrictions apply to deserialization

Description: We need to clarify that if a user requests via data system metadata a restriction on geographic placement, a create would fail, and a deserialize would fail.

Proposed change:

In section 16.4, Table 20 - Data Systems Metadata, under `cdmi_geographic_placement`, replace:

Contains a list of geopolitical identifiers, each specifying a region where the object is permitted or not permitted to be stored. Geopolitical boundaries are a list of ISO-3166 country codes. A "!" in front of a country code excludes that country from the previous list of geopolitical boundaries. This metadata limits where the data may be placed physically and constrains all cloud movement of the data within the cloud. It does not apply to data once it leaves the cloud. This metadata takes precedent over other metadata, such as `cdmi_data_dispersion`.

With

Contains a list of geopolitical identifiers, each specifying a region where the object is permitted or not permitted to be stored. Geopolitical boundaries are a list of ISO-3166 country codes. A "!" in front of a country code excludes that country from the previous list of geopolitical boundaries.

This metadata limits where the data may be placed physically and constrains all cloud movement of the data within the cloud. If this data system metadata item is supported, and the cloud is not able to guarantee that the data will be stored within the specified geographic placement restrictions, the create or update operation should fail with an HTTP 400 status code. It does not apply to data once it leaves the cloud, or when it is stored on clouds that do not support this data system metadata item. This metadata takes precedent over other metadata, such as `cdmi_data_dispersion`.

Edits resulting from discussion at the April 13th TWG call:

Contains a list of geopolitical identifiers, each specifying a region where the object is permitted or not permitted to be stored. Geopolitical boundaries are a list of ISO-3166 country codes. A "!" in front of a country code excludes that country from the previous list of geopolitical boundaries.

This metadata limits where the object is permitted to be stored. If this data system metadata item is supported, and the cloud is not able to guarantee that the data will be stored within the specified geographic placement restrictions, create or update operations shall fail with an HTTP 403 status code. Specified geographic placement restrictions only apply within the context of a CDMI system that supports this data system metadata item. This metadata takes precedent over other metadata, such as `cdmi_data_dispersion`.

CHANGES Made the specified changes as agreed in April 12 call to Section 16.4, Table 20 - Data Systems Metadata, *under `cdmi_geographic_placement`*.

Trac Ticket #423 Rename fields in versioning vendor extension

Description: Currently, the versioning vendor extension has fields in the form of "com.netapp.nnn". Under the new approach for handling vendor extensions that are to be merged into the spec, the standard "cdmi_nnn" conventions should be used. Update text to reflect this change.

Proposed changes:

Replace all instances of "com.netapp." with "cdmi_" e.g., `com.netapp.versioning` -> `cdmi_versioning`

Discussed on April 13th TWG call, and decision is to include in 1.0.1i.

CHANGES Made the specified changes.

Trac Ticket #424 Description of HTTP status codes 401 and 403 is incorrect

Description: In version 1.0.1h of the spec, the description of the 401 and 403 HTTP status codes does not match the HTTP/1.1 RFC in all Response Status tables that reference these codes:

- The "401 Unauthenticated" HTTP Status code should be "401 Unauthorized".
- The "403 Unauthorized" HTTP Status code should be "403 Forbidden".

Fixes should be made to the response status tables. Just the HTTP Status text needs to be updated. The descriptions can be left as-is.

Table 5 does not require any changes.

CHANGES Made the specified changes.

Trac Ticket #425 Update operations should return 204 No Content instead of 200 OK

Description: Update (HTTP PUT) operations do not return a response message body and so should return a 204 No Content HTTP response instead of 200 OK.

This applies to the following sections:

- Section 8.6, Update a Data Object (CDMI Content Type)
- Section 8.7, Update a Data Object (Non-CDMI Content Type)
- Section 9.6, Update a Container (CDMI Content Type)
- Section 10.4, Update a Domain (CDMI Content Type)
- Section 11.4 Update a Queue Object (CDMI Content Type)

Additionally, because these operations do not return a response message body, the "Accept" header should be removed from the "Request Headers" table and all examples in the above sections.

Accept header should be optional. Need list of changes required to the spec. We should consider clarifying that this does not mean that there is "no content", just that no content is returned.

Spec changes:

- 1 In all places in the spec where we return just "HTTP/1.1 200 OK" as the response, change it to "HTTP/1.1 204 No Content". This change should be made to examples 8-11, 8-12, 8-13, 8-14, 9-8, 9-9, 10-5, 11-6.
- 2 Response Status table in Section 8.6, 8.7, 9.6, 10.4, 11.4 should remove "200 OK" and replace it with the "204 No Content" table row (take from Table 5)

Discussed on the 2011-05-11 TWG call, and approved the change from 200 OK to 204 No Content for addition to the spec.

However, the Accept change needs to be reviewed and have spec text proposed. This will be done in a separate trac ticket.

CHANGES Made the specified changes as outlined in Items 1 & 2 above with the following additions to Item 1: Also changed examples 8-15, and the new examples 8-17, 8-18, and 8-19 (added with Trac ticket #419).

Trac Ticket #427 Capabilities text error in 9.8 through 9.10

Description: Replace "capability the" with "and the" in sections 9.8, 9.9 and 9.10.

CHANGES Made the specified changes.

Trac Ticket #429 Corrections to Chapter 1: Introduction

Description: Minor corrections to Chapter 1: Introduction in revision 1.0.1h:

- Page X, Table 1: Chapters listed under the 'Chapter' column do not match those in the document. Chapter 6 should be 'Common Operations', Chapter 7 should be 'Interface Standard', Chapter 17 should be 'Logging'.
- Table 1: in the Data Object row, the sentence "normative standard for data object" should be changed to be consistent with the other operations section descriptions, which use "normative standard of ..."

CHANGES Made the specified changes.

Trac Ticket #430 Corrections to Chapter 3: Terms

Description: Minor corrections to Chapter 3: Terms in revision 1.0.1h:

- Section 3.0, in the note "...are taken from the SNIA Dictionary, except for the following: CRC" -- CRC actually is defined in the current SNIA Dictionary.
- Section 3.0, the green 'SNIA Dictionary' link is not active when I hover over it, it only works when clicking the first part of the sentence.
- Section 3.1 Access Control List: "a persistent list, commonly composed group of Access Control Entries (ACEs)" -- typo, remove "group"
- Section 3.18, WBEM - I had to look this up, should perhaps be expanded to "Web-Based Enterprise Management (WBEM)"
- Section 3.19, REST: Not clear why this is a separate entry instead of grouping it under "3.18 Representational State Transfer" to be consistent with other definitions that include an acronym.

CHANGES Made the specified changes.

Trac Ticket #431 Corrections to Chapter 4: Conventions

Description: Minor corrections to Chapter 4: Conventions in revision 1.0.1h:

- Section 4.1, Some interfaces also have an additional 'Delayed Completion of Create' section that is not included in this table.
- Section 4.1, Table 2: "The GET, PUT, and POST semantics." -- add "DELETE"
- Section 4.1, Table 2: Request headers uses an example "X-Cloud-Client-Specification-Version", which isn't used in the spec. This should be changed to "X-CDMI-Specification-Version"
- Section 4.2, not sure what 'probably action' is, drop 'probable'
- Section 4.2, some of the typographical conventions (e.g., warning) do not actually appear to be used anywhere in the document.

Proposed changes:

- 1 Section 4.1, Some interfaces also have an additional 'Delayed Completion of Create' section that is not included in this table.
DS: Suggest adding the following line to the table, below the Synopsis row: "Delayed Completion of Create", "For long-running operations, a description of behaviour when the operation does not immediately complete."
- 2 Section 4.1, Table 2: "The GET, PUT, and POST semantics." -- add "DELETE"
DS: Suggest TWG approve change.
- 3 Section 4.1, Table 2: Request headers uses an example "X-Cloud-Client-Specification-Version", which isn't used in the spec. This should be changed to "X-CDMI-Specification-Version"
DS: Suggest TWG approve change.
- 4 Section 4.2, not sure what 'probable action' is, drop 'probable'
DS: Suggest TWG approve change.
- 5 Section 4.2, some of the typographical conventions (e.g., warning) do not actually appear to be used anywhere in the document.
DS: Will be addressed in the changes from Ralph.

Discussed on the 2011-05-11 TWG call, and approved for addition to the spec.

CHANGES Made the specified changes, including removing "WARNING" from the conventions section.

Trac Ticket #432 Corrections to Chapter 5: Overview of Cloud Storage

Minor corrections to Chapter 5: Overview of Cloud Storage in revision 1.0.1h:

- Section 5.3, in the sentence "thin provisioned and billed for, based on actual usage" -- remove the comma
DS: Suggest TWG approve change.
- Section 5.3, last sentence, "they all pretty much support the same set of operations", does not seem appropriate language for a standards doc. How about "they all support a roughly-equivalent set of operations"
DS: Suggest TWG approve change.
- Section 5.4, first sentence, "with very little offering of additional data services for that data", would read better as "with few additional services for that data"
DS: Suggest TWG approve change.
- Section 5.4, in "For more information on support for storage system metadata, see Chapter 16, "Metadata"" -- should link to the specific sub-section for SSMD, Section 16.3, for consistency with the references in the preceding and following paragraphs that link directly to the user/data system metadata subsections.
DS: Suggest TWG approve change.
- Section 5.5, in "of it's parent's data system" -- "data system" should be "data system metadata"
DS: Suggest TWG approve change.
- Section 5.6, Figure 5, there are boxes for 'Information Services (future)' but no mention of what this actually means in the text.
DS: Will need to discuss — Separated out into ticket #478
- Section 5.8, Figure 6, in the Capabilities box, the URI is incorrect, it should be 'https://<offering>/cdmi_capabilities/'
DS: Suggest TWG approve change.
- Section 5.8, Figure 6, in the Domains box, the URI is incorrect, it should be 'https://<offering>/cdmi_domains/'
DS: Suggest TWG approve change.
- Section 5.8, in "The CDMI does not need to be used" -- remove the "The"
DS: Suggest TWG approve change.
- Section 5.9, in "use of the HTTP protocol, such as content-size" -- "content-size" would be clearer as "content-length" to correspond to the actual HTTP header name.
DS: Suggest TWG approve change.
- Section 5.11, in the OID field descriptions, bullet 2, the IANA link is activated when over the word 'and', not over the link, at least in my viewer
DS: Suggest TWG approve change.
- Section 5.14, both first and second paragraphs, '[ISO-8601]' references are not linked or highlighted like other references

DS: Suggest TWG approve change.

- Section 5.14, in "must not be used and represented as" -- this sentence is unclear, suggest changing it to "must not be used and instead should be represented as" use "shall" instead of "should".

Discussed on the 2011-05-11 TWG call, and approved for addition to the spec.

CHANGES Made the specified changes except for changes to 5.6, which will be addressed in Trac Ticket # 478.

Trac Ticket #433 capabilitiesURI is incorrect in example responses

Description: In revision 1.0.1h, the name of the capabilities object in the capabilitiesURI field in most of the response examples has the wrong case. The names should be lower case.

This applies to the following sections:

- Section 6.2, 6.4, 9.2, 9.4, 15.2.2: "/cdmi_capabilities/Container/" should be "/cdmi_capabilities/container/"
- Section 6.3, 6.5, 8.2, 8.4, 9.8, 15.2.2, 16.1.9, 20.1, 20.2: "/cdmi_capabilities/DataObject/" should be "/cdmi_capabilities/dataobject/"
- Section 9.10, 11.2, 11.3, 15.2.2: "/cdmi_capabilities/Queue/" should be "/cdmi_capabilities/queue/"
- Section 10.2: "/cdmi_capabilities/Domain/" should be "/cdmi_capabilities/domain/"

Discussed on the 2011-04-27 TWG call, and approved for incorporation into the next draft of the CDMI specification - 1.0.1i.

CHANGES Made the specified changes.

Trac Ticket #434 Corrections to Chapter 6: Common Operations

Description: Minor corrections to Chapter 6: Common Operations in revision 1.0.1h:

- Section 6.1: The GET example is missing the Accept header, which is mandatory according to request headers specified in section 12.2.
- Section 6.2: The example is missing a trailing '/' on the URI path. It should be '/MyContainer/'; according to synopsis in section 9.2, the / is required.
- Section 6.2: The example PUT has an extra slash after the HTTP version specifier, 'HTTP/1.1/' should be 'HTTP/1.1'.
- Section 6.3: The example PUT has an extra slash after the HTTP version specifier, 'HTTP/1.1/' should be 'HTTP/1.1'.

- Section 6.4: The example request has "Accept: */*" which should be "Accept: application/cdmi-container".
- Section 6.5: The example request has "Accept: application/cdmi_object" which should be "Accept: application/cdmi-object".

Accept: */* is acceptable according to RFC 2616.

All of the other changes need to be applied to the spec.

Discussed on the 2011-04-27 TWG call, and approved for incorporation into the next draft of the CDMI specification.

Note: The second to last change should not be made, as */* is a valid accept type.

CHANGES Made the specified changes, ignoring the change to Section 6.4, bullet 5.

Trac Ticket #435 Object references are not clearly distinguished in container listing

Description

Section 7.3, paragraph 2 of revision 1.0.1h states that "References are visible as children in a container and are distinguished from non-references by a trailing "?" character added to the reference name."

It is not clear how a reference in a container listing is distinguished from a non-reference child object that happens to have a "?" as the last character of its name (a perfectly valid name).

It would also be helpful if the output format was described (and an example provided in) Section 9.4 Read a Container.

Object, Container, Queue and Domain names must be escaped as per URI escaping rules:

reserved = ";" | "/" | "?" | ":" | "@" | "&" | "=" | "+" |

"\$" | ","

See RFC <http://www.ietf.org/rfc/rfc2396.txt>

This makes the "?" at the end of the child name distinguishable from an object name ending with the character "?".

We may wish to add clarifying text around this to the spec.

(kjamieson) - I would recommend adding the clarifying text and referencing RFC 3986 (which obsoletes 2396), as it is not an obvious interpretation that these values should be URI-escaped.

Proposed spec text change:

- 1 Change the "description" column of the "children" row in the table under Response Message Body in section 9.4 from:
 "Names of the children objects in the container. Child containers end with "/"."
 to
 "Names of the children objects in the container. All children names shall have reserved characters escaped according to RFC 3986. For example, the "%" character in a name would be replaced with "%25".
 Children that are containers shall have "/" appended to the child name, and children that are references shall have "?" appended to the child name."
- 2 Add a reference to RFC 3986.
- 3 We need to call out that URI escaping is applied for ALL URIs. Add to section 5.13. (DS to add new ticket for this).

Discussed on the 2011-05-11 TWG call, and approved for addition to the spec.

Added ticket #487 to ensure clarity about URI escaping.

CHANGES Added [RFC 3986] to Chapter 2 - References and updated Chapter 9.4.

Trac Ticket #436 Corrections to Chapter 7: Interface Standard

Description: Minor errors in Chapter 7: Interface Standard in revision 1.0.1h:

- 1 Section 7.1, Table 5, HTTP Status Codes is missing the following HTTP Status codes that are used in the standard:
 - 206 Partial Content
 - 302 Found
 - 304 Not Modified
- 2 Section 7.1, Table 5, 'Used for' description for 200 is incorrect, as it is not used just for GET requests; a more accurate description would be 'The request has succeeded.'
- 3 Section 7.1, Table 5, the result codes 500 Internal Server and 501 Not Implemented are not mentioned in the document anywhere -- it is not clear why they would then need overloading and cannot simply maintain their original meaning from HTTP/1.1.
- 4 Section 7.2, first paragraph has 'The five types...' but only lists four -- queue objects are missing from the list.
- 5 Section 7.1, Table 5, HTTP Status Codes is missing the following HTTP Status codes that are used in the standard:
 - 206 Partial Content
 - 302 Found
 - 304 Not Modified
- 6 DS: Add the following lines to the table:

"206 Partial Content", "A requested range of the data object content was returned in response."

"302 Found", "The URI is a reference to another URI."

304 is a remnant from when we had a custom header specifying that an update should only occur when a given condition was true. We abandoned this in favour of the existing HTTP mechanisms to indicate that the operation should only proceed if the entity exists/does not exist.

Recommendation is to remove all instances where 304 is mentioned, since we don't replicate other parts of the HTTP standard that we don't modify.

Spec change:

- 1 Remove the "304" line from the "Response Status" tables in sections 8.2, 9.2, 9.10, 10.2 and 11.2
- 2 Section 7.1, Table 5, 'Used for' description for 200 is incorrect, as it is not used just for GET requests; a more accurate description would be 'The request has succeeded.'
- 3 Replace "Resource retrieved successfully." with "The request has succeeded."
- 4 Section 7.1, Table 5, the result codes 500 Internal Server and 501 Not Implemented are not mentioned in the document anywhere -- it is not clear why they would then need overloading and cannot simply maintain their original meaning from HTTP/1.1.

Remove these two lines from Table 5.

- 5 Section 7.2, first paragraph has 'The five types...' but only lists four -- queue objects are missing from the list.

Replace:

"The five types of resource objects in the model include data objects, container objects, domain objects, and capability objects (see Table 6)."

with

"The five types of resource objects in the model include data objects, container objects, queue objects, domain objects, and capability objects (see Table 6)."

- 6 We also should remove the hyperlink for "domain", since we have not hyperlinked the other object types, and the concept here is "domain objects", not just "domain".

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the changes specified in Items 1-6 above.

Trac Ticket #437 Corrections to Chapter 8: Data Object Resource Operations

Description: Minor corrections or clarifications needed in Chapter 8: Data Object Resource Operations in revision 1.0.1h:

TWG reviewed these items on the 2011-06-01 call, with the following spec text changes approved:

Subclause 8.1:

- 1 Section 8.1, in "would return first thousand bytes of the value field: http://cloud.example.com/dataobject?value:0-1000" -- this would actually return the first thousand-and-one bytes of the value; a range of "0-999" would return the first thousand bytes.

TWG: The range should be changed to "0-999".

- 2 Section 8.1.2, sentence, "...which is to say, by the timestamps placed on the responses by the server-side implementation...", I'm assuming this means `cdmi_mtime`, as I don't see an explicit timestamp sent in any responses. Or does this refer to the HTTP/1.1 Date header (which isn't shown in any of the Response Headers tables or examples)? This needs clarification as to what is being referred to.

TWG: This section was deliberately vague, as it is referring to eventual consistency recommendations. It could refer to the `mtime`, or to the HTTP timestamp in the response header.

TWG: Decision is to pull this issue out as a separate ticket, and re-review at the next TWG meeting.

Subclause 8.2

- 1 Section 8.2, Request message body table, for `domainURI` mentions that a user needs "cross_domain" privilege, perhaps reference to Table 8 in Domain Membership where this privilege is defined?

TWG: Marie to review to determine compatibility with ISO formatting changes.

- 2 Section 8.2, Response Message Body table, in "objectName shall be set to the Object ID of the object.", remove the extra trailing " character. This comment also applies to Section 8.4.

TWG: Remove double quote.

Subclause 8.3

Section 8.3, Response Status table, 202 Accept and 304 Not Modified result codes are not listed. Shouldn't these codes also be applicable to a non-CDMI PUT?

TWG: 304 is no longer included in the spec. Add 202 to the Response Status table in clause 8.3.

Subclause 8.4

- 1 Section 8.4, Response Message Body table, under "valuerange" there is a typo in "non-contiguous" ("non-contiguous")

TWG: Correct typo.

- 2 Section 8.4, Response Status table, in "404 Not Found A data object was not found at the specified URI" -- to clarify, 404 is returned if *no object* was found at the specified URI, while 406 is returned if a non-data object was found at the specified URI"?

TWG: 406 definition looks good. For 404, replace the description with "A resource was not found at the specified URI". This change should be made in Table 5, and in all Response Status tables that include 404.

- 3 Section 8.4, Example 8-7, The description mentions 'first ten bytes' but request is for the first eleven (range 0-10)

TWG: Change range to "0-9".

Subclause 8.5

- 1 Section 8.5, Response Message Body, formatting of HTTP status codes in text is not using a consistent font, probably other instance in the doc
TWG: Formatting will be reviewed as part of ISO changes.
- 2 Section 8.5, Response Message Body, last bullet, typo 'codeof' should be 'code of', extra space before period
TWG: Make changes.
- 3 Section 8.5, Response Status table, "204 No Content Data object exists but has no content." -- under what circumstances would this status ever be returned? A GET of a data object with a zero-length value should return 200 OK with Content-Length: 0 rather than 204 No Content.
TWG: 204 should be removed from the clause 8.5 Response Status table.
- 4 Section 8.5, Example 8-10, 'first ten bytes', but eleven are returned
TWG: Change description to "first eleven bytes".

Subclause 8.6

- 1 Section 8.6, Synopsis needs to define <fieldname>
TWG: This was removed as part of another ticket. No change needed.
- 2 Section 8.6, Capabilities, does the "cdmi_modify_metadata" capability also indicate the ability to modify a field (other than value), e.g. "mimetype"?
TWG: David's understanding here is that this is controlled by the "cdmi_modify_value" capability. Only the metadata is controlled by the "cdmi_modify_metadata" capability.
Recommended change, replace:
"Support for the ability to modify the value of an existing data object is indicated by the presence of the "cdmi_modify_value" capability in the specified object."
with:
"Support for the ability to modify the value and/or mimetype of an existing data object is indicated by the presence of the "cdmi_modify_value" capability in the specified object."
- 3 Section 8.6, Request Headers, "Accept" header should be removed from this table and all examples as there is no response message body
TWG: Change already made.
- 4 Section 8.6, Request Message Body, if no "domainURI" is specified in the request, shouldn't the object retain its current domain rather than it being reset to the parent domain?
TWG: That is correct - The previous domain should be preserved unless specified.
In the domainURI table row, change the last bullet in the description from:
"* If not specified, the parent domain shall be used."
with:
"* If not specified, the existing domain shall be preserved."

- 5 Section 8.6, Example 8.11, does supplying the empty metadata field essentially delete all user metadata if there was some previously specified (e.g. on create)?

TWG: Yes, if done against the object. No change needed.

Subclause 8.8

- 1 Section 8.8, Capabilities, "cdmi_delete_object" should be "cdmi_delete_dataobject", as per Section 12.1.4

TWG: Make change.

DS will create additional trac ticket as described above, then assign to Marie.

Ticket extracted to new ticket #525. Reassigning remainder of items to Marie.

CHANGES Made the specified changes with the following additions:

- Subclause 8.6 Item 2: Applied change to same text in 8.7.
- Subclause 8.6 Item 4: Applied change to domainURI table row in 8.2.
- Subclause 8.8 Item 1: Applied change to same text in 8.9.

Trac Ticket #438 Clarification needed for paratial data object create/read behavior

Description: In Chapter 8: Data Object Resource Operations in 1.0.1h, the behaviour of partial content create and read operations needs clarification in a few sections, including:

- Section 8.1, if "Byte ranges are specified as per Section 14.35.1 of [RFC2616]" means that all byte range formats supported by HTTP/1.1 (including, for example, multiple ranges), this should be explicitly re-iterated in the description of the <range> field in all applicable read operations sections (e.g., 8.4 Read a Data Object).
- Section 8.3, Request Headers table, the description of "X-CDMI-Partial" for a non-CDMI PUT indicates that the completionStatus field shall be "Processing", but there is no JSON response to this request.
- Section 8.5, Response Headers table, assuming this is conformant to HTTP/1.1, Content-Range header may also be returned if a byte range is requested, and the Content-Type may also be multipart/byteranges if non-contiguous ranges are returned (it may be simpler to refer to Section 14.16 in the HTTP/1.1 RFC for the proper format of a ranged retrieval request/response)
- Section 8.5, according to rfc2616, 10.2.7 a 206 Partial Content response must include the Date header
- Section 8.5, Response Message Body, should clarify what is returned if the value has gaps (e.g., only the bytes up to the first gap, as for a CDMI Content-Type GET?)

- Section 8.7, Request Headers table should list "Content-Range" as an optional header (for a byte range update)

Proposed changes:

- 1 Section 8.1, if "Byte ranges are specified as per Section 14.35.1 of [RFC2616]" means that all byte range formats supported by HTTP/1.1 (including, for example, multiple ranges), this should be explicitly re-iterated in the description of the <range> field in all applicable read operations sections (e.g., 8.4 Read a Data Object).

TWG: No objections to restricting this to inclusive single ranges.

Proposed spec text change is to replace:

"Byte ranges are specified as per Section 14.35.1 of [RFC2616]"

with

"Byte ranges are specified as single inclusive byte ranges as per Section 14.35.1 of [RFC2616]"

- 2 Section 8.3, Request Headers table, the description of "X-CDMI-Partial" for a non-CDMI PUT indicates that the completionStatus field shall be "Processing", but there is no JSON response to this request.

TWG: No change - this is referring to the state of the object, rather than any data returned.

- 3 Section 8.5, Response Headers table, assuming this is conformant to HTTP/1.1, Content-Range header may also be returned if a byte range is requested, and the Content-Type may also be multipart/byteranges if non-contiguous ranges are returned (it may be simpler to refer to Section 14.16 in the HTTP/1.1 RFC for the proper format of a ranged retrieval request/response)

TWG: As per HTTP. No change needed.

- 4 Section 8.5, according to rfc2616, 10.2.7 a 206 Partial Content response must include the Date header

TWG: As per HTTP. No change needed.

- 5 Section 8.5, Response Message Body, should clarify what is returned if the value has gaps (e.g., only the bytes up to the first gap, as for a CDMI Content-Type GET?)

TWG: The specified read behaviour for handling gaps is stated in 8.6, which is the only way that gaps can be created.

We should add clarifying text to describe the read behavior when there are gaps. This would be added to the response body section of 8.5 and to the value section of 8.4.

- In section 8.4, in the response message body JSON, in the description of the "value" row, add the following text:

"When reading a value, zeros shall be returned for any gaps resulting from non-contiguous writes."

- In section 8.5, under the "Response Message Body", add the following bulleted sentence below the second existing sentence:

"When reading a value, zeros shall be returned for any gaps resulting from non-contiguous writes."

- 6 Section 8.7, Request Headers table should list "Content-Range" as an optional header (for a byte range update)

TWG: Content-Range is not currently listed. Add header row as per 8.5.

TWG: In Section 8.5 (and now 8.7), the header should be "Content-Range", not "Range".

Note: Changed 2 weeks ago by kjamieson

Changing "Range" to "Content-Range" in the Request Headers table in Section 8.5 is incorrect -- it should be "Range". "Content-Range" would be returned in the response headers for a byte range retrieval.

Example 8-10 correctly shows a "Range" header in the HTTP GET request and "Content-Range" in the response.

Per HTTP/1.1:

- The "Range" header is used in an HTTP GET request to retrieve a byte range.
- The "Content-Range" header is used in an HTTP request or response with a partial entity-body to indicate where in the full entity-body the partial body appears.

So the usage of Content-Range as a request header in Section 8.7 to update a range of an object is valid.

Changes: Kevin's comment above is correct. We should back out that change to section 8.5. Specifically, in the Request Headers table in 8.5, the header should just be "Range".

- CHANGES**
- June 1: Made the specified changes in Items 1, 5, and 6. No changes recommended for Items 2-4.
 - June 15: Reversed change in Item 6 as per the latest comments (changed Content-Range to Range in in the Request Headers table inSection 8.5). Version 1.0.1j.

Trac Ticket #439 **Serialization capabilities are unclear**

Description: In Section 8.2 of revision 1.0.1h, in the Capabilities subsection, it states that:

“support for the ability to serialize the source data object is indicated by the presence of the "cdmi_serialize_dataobject", "cdmi_serialize_container", "cdmi_serialize_domain", or "cdmi_serialize_queue" capability in the parent container”.

This is inconsistent with Chapter 12, which lists "cdmi_serialize_dataobject", "cdmi_serialize_domain", and "cdmi_serialize_queue" as capabilities of their respective objects, not capabilities of a container. There are no such capabilities listed in Table 14 - Capabilities for Containers.

Which section is correct? Having these as capabilities of the destination parent container seems more appropriate given that the serialization seems to be an operation that is performed by the target of the serialization rather than the source.

TWG: The following changes are approved:

- 1 Move the row "cdmi_serialize_dataobject" from Table 105 to Table 106, above "cdmi_serialize_container".
- 2 Move the row "cdmi_serialize_domain" from Table 107 to Table 106, below "cdmi_serialize_container".
- 3 Move the row "cdmi_serialize_queue" from Table 108 to Table 106, below "cdmi_serialize_container".
- 4 Add a new row "cdmi_modify_deserialize_dataobject" to Table 105, below "cdmi_modify_metadata", with the description, "If present and "true", this capability indicates that the data object permits the deserialization of a serialized data object into the data object as an update."
- 5 Add a new row "cdmi_modify_deserialize_container" to Table 106, below "cdmi_modify_metadata", with the description, "If present and "true", this capability indicates that the container object permits the deserialization of a serialized container object into the container object as an update."
- 6 Add a new row "cdmi_modify_deserialize_domain" to Table 107, below "cdmi_modify_metadata", with the description, "If present and "true", this capability indicates that the domain object permits the deserialization of a serialized domain object into the domain object as an update."
- 7 Add a new row "cdmi_modify_deserialize_queue" to Table 108, below "cdmi_modify_metadata", with the description, "If present and "true", this capability indicates that the queue permits the deserialization of a serialized queue into the queue as an update."

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #441 Use of 304 Not Modified for "already exists" on create is inconsistent with HTTP/1.1 RFC

Description: In revision 1.0.1h, the 304 Not Modified HTTP result code is used to indicate that "the operation conflicts because the object already exists."

This seems like an abuse of the 304 Not Modified status code, which, as used in the HTTP/1.1 RFC, is only defined as a response for a conditional GET operation to indicate that an object has not changed since the last time the client retrieved the object.

It would seem more appropriate to use another status code like 409 Conflict to indicate this error condition.

If the interpretation of this status code is going to be overloaded by CDMI, it should be added to Table 5 HTTP Status Codes in Section 7.1.

This applies to all object create sections, including:

- Sections 8.2, 8.3, 9.8, 9.8: Data Objects

- Sections 9.2, 9.3: Containers
- Sections 9.10, 11.2: Queues
- Section 10.2: Domains

On further reflection, it is not even clear to me that a CDMI server would ever return a "304 Not Modified" ("object exists") response, given that a CDMI Create PUT request is indistinguishable from a CDMI Update PUT request.

Given that the 304 is only referred in the spec in the list of HTTP result codes, I think we can safely remove this, since no CDMI operation would trigger this result. (Only additional HTTP headers not discussed in CDMI would trigger this.)

TWG decision is to remove these lines.

Marie: In each "Response Status" table, remove the table row starting with "304 Not Modified."

CHANGES These were removed in Trac Ticket #436. (See "Trac Ticket #436" on page 133.)

Trac Ticket #442 **cdmi_create_copy and cdmi_create_move container capabilities are referred to but not defined**

Description: Revision 1.0.1h, Section 8.2, subsection Capabilities refers to the "cdmi_create_copy" and "cdmi_create_move" container capabilities as indicating whether or not a data object can be created or moved into the container. These capabilities are not defined in Section 12.1.5 Container Capabilities (or anywhere else in Chapter 12). These capabilities are also referenced in Sections 9.2 and 9.8.

Are copy/move capabilities of the destination parent container or of the source object for the operation? For containers, domains, and queue objects, there are specific capabilities of the source object indicating whether or not it may be copied/moved (e.g., `cdmi_copy_container` and `cdmi_move_container`), but there is no corresponding capability for data objects.

Summary changed from `cdmi_create_copy` and `cdmi_crate_move` container capabilities are referred to but not defined to `cdmi_create_copy` and `cdmi_create_move` container capabilities are referred to but not defined

Expanding on the last comment, the references to the `cdmi_copy_queue/cdmi_move_queue` (Sections 9.10, 11.2), `cdmi_copy_domain/cdmi_move_domain` (Section 10.2), and `cdmi_copy_container/cdmi_move_container` (Section 9.2) capabilities also describe these as capabilities of the parent container of the operation, which is inconsistent with the descriptions of these capabilities in Chapter 12 which describe them as capabilities of the source queue/domain/container to be copied or

[TEJ] This is true. There are couple of things that are not consistent around "copy/move" capabilities. Document indicates that the "cdmi_create_copy" and "cdmi_create_move" capability is needed to move data objects and containers. But these are not defined under the "Container capabilities" table. Instead, "cdmi_move_container" and "cdmi_copy_container" are defined. This is what is in the document:

From Section 8.2 - Capabilities:

- If the new data object is a copy of an existing data object, support for the ability to copy is indicated by the presence of the "cdmi_create_copy" capability in the parent container.
- If the new data object is the destination of a move, support for the ability to move the data object is indicated by the presence of the "cdmi_create_move" capability in the parent container.

From Section 9.2 - Capabilities:

- If the new container object is a copy of an existing container, support for the ability to copy is indicated by the presence of the "cdmi_create_copy" capability in the parent container.
- If the new container is the destination of a move, support for the ability to move the container is indicated by the presence of the "cdmi_create_move" capability in the parent container.

Based on what is defined in table 14 (Capabilities for Containers), the capabilities names use the form: cdmi_<verb indicating the operation>_<resource to which the capability is defined>

On that basis, the capabilities to indicate whether the data object should be copied or moved, should be specified using something like:

- cdmi_copy_dataobject
- cdmi_move_dataobject

Since it seems like there is already a "cdmi_copy_container" and "cdmi_move_container" to indicate whether containers are capable of being copied/moved. Going by the above thought, I have come up with details to be added for the above capability.

- cdmi_move_dataobject, JSON String, If present and "true", indicates that the dataobject may be moved (via PUT) to another URI.
- cdmi_copy_dataobject, JSON String, If present and "true", indicates that the dataobject may be copied (via PUT) to another URI.

Not sure if we should further clarify that the source dataobject/container is being moved. If we change the capability names, then section 8.2 and section 9.2 need to be updated to reflect the change in the capability name. This needs to be discussed.

See tickets #463 and #464 for proposals for capabilities for moving and copying objects.

cdmi_create_copy and cdmi_create_move capabilities have been removed from the capabilities section. However, they still exist in various places in the spec text. This needs to be updated. DS to write updated spec text.

TWG: We have the following system capabilities:

- cdmi_object_move_from_local
- cdmi_object_move_from_remote
- cdmi_object_move_from_ID
- cdmi_object_copy_from_local
- cdmi_object_copy_from_remote

This is viewed as being sufficient, but oddly, are not referenced from anywhere else in the spec. We need to update the text that refers to cdmi_create_copy and cdmi_create_move and update them to refer to these system capabilities, or not mention these at all.

DS to write spec text changes to remove the create_copy and create_move capabilities, and add the above five system capabilities where appropriate.

Proposed Changes

The five referred to system capabilities have already been added to the spec, but there are a few open issues still remaining for this ticket:

- 1 Add to Table 106: "cdmi_move_dataobject", "If present and "true", this capability indicates that a data object may be moved into the container."
- 2 Add to Table 106: "cdmi_copy_dataobject", "If present and "true", this capability indicates that a data object may be copied into the container."
- 3 In Table 106: replace the description text for "cdmi_move_container" with "If present and "true", this capability indicates that a container object may be moved into the container."
- 4 In Table 106: replace the description text for "cdmi_copy_container" with "If present and "true", this capability indicates that a container object may be copied into the container."
- 5 In 8.2.3 and 9.2.3, replace "cdmi_create_copy" with "cdmi_copy_dataobject"
- 6 In 8.2.3 and 9.2.3, replace "cdmi_create_move" with "cdmi_move_dataobject"
- 7 In 9.2.3, replace "cdmi_create_copy" with "cdmi_copy_container"
- 8 In 9.2.2, replace "cdmi_create_move" with "cdmi_move_container"

Ready to discuss.

TWG: Spec text approved.

Will review entire capabilities section and cross-references for consistency and correctness at face-to-face meeting.

CHANGES Made the specified changes.

Trac Ticket #443 Corrections to Chapter 9: Container Object Resource Operations

Description: Minor corrections to Chapter 9: Container Object Resource Operations in revision 1.0.1h:

- Section 9.1, Example text states, "For example, the below URI returns the first two children...", but the uri range is given as "0-2", which would return the first three children
- Section 9.1.2, "*For example, a data object may be accessible via ...*" should be *container object* instead
- Section 9.2, Request Message Body table, 'exports' has text, "...when referencing a data object.", should be "...when referencing a container object."
- Section 9.2, Request Message Body table, 'move' has typo "container.", should be "container,"
- Section 9.2, Request Message Body table, 'reference' has text, "URI of a CDMI data object...", should be, "URI of a CDMI container object..."
- Section 9.2, Request Message Body table, 'reference' specifies what should happen if other "Optional*" fields are supplied, but no other "Optional*" fields do the same. The 400 Bad Request response should be mentioned in one place (e.g., table footer, as in Section 9.8).
- Section 9.4, Example 9-3, is missing container trailing / for container name: request should be 'GET /MyContainer/ HTTP/1.1'; "objectURI" field should be "/MyContainer/"; "objectName" field should be "MyContainer/"
- Section 9.4, Example 9-5, text "...to read children 0..2...", should be "...to read children 0..2 and childrenrange..."
- Section 9.5, Request Message Body section, specifies the response statuses for optional/undefined fields, shouldn't this be in specified in the Response Status table?
- Section 9.5, Request Message Body section and Response Status table have conflicting scenarios for returning a 404, in one case it's if a field is not found and the other if a container is not found at the request URI (the latter seems correct)
- Section 9.5, Example 9-7, text "...to read children 0..2...", should be "...to read children 0..2 and childrenrange..."
- Section 9.6, After "Delayed completion of a snapshot:", the first sentence should be a new paragraph

- Section 9.8, Request Message body table, 'reference' statement that "If other fields from this table ..." should be removed from the field description, as this is not mentioned for any other fields and is already stated in the table footer.
- Section 9.8: The spec erroneously lists an Accept header in the Response Headers table on page 70. This should be a Content-Type header instead, as correctly reflected in the following examples.
- Section 9.8, Example 9-11, response is missing 'Location: ' header-name text before the URL is given.
- Section 9.10, Example 9-15 omits the Content-Type header required as its request body features an empty JSON object
- The section titles are inconsistent, some use "Container Object" (e.g., Section 9.4, "Read a Container Object") while some use just "Container" (e.g., Section 9.6 "Update a Container"). I'd suggest standardizing on "Container Object"

Spec changes

- 1 Section 9.1, Example text states, "For example, the below URI returns the first two children...", but the uri range is given as "0-2", which would return the first three children
DS: Suggest TWG approve change.
Spec text: replace "first two children" with "first three children".
- 2 Section 9.1.2, "For example, a data object may be accessible via ..." should be container object instead
DS: Suggest TWG approve change.
- 3 Section 9.2, Request Message Body table, 'exports' has text, "...when referencing a data object.", should be "...when referencing a container object."
DS: Suggest TWG approve change.
- 4 Section 9.2, Request Message Body table, 'move' has typo "container.," should be "container,"
DS: Suggest TWG approve change.
- 5 Section 9.2, Request Message Body table, 'reference' has text, "URI of a CDMI data object...", should be, "URI of a CDMI container object..."
DS: Suggest TWG approve change.
- 6 Section 9.2, Request Message Body table, 'reference' specifies what should happen if other "Optional*" fields are supplied, but no other "Optional*" fields do the same. The 400 Bad Request response should be mentioned in one place (e.g., table footer, as in Section 9.8).
DS: Recommend adding text at end of the request body tables. We already made this change to 9.8, so we should be consistent:
In sections 8.2, 9.2, 9.10, 10.2, 11.2 and 11.6, replace:
"If present, only one of these parameters shall be specified in any given operation, and these fields are not persisted."
with

"*Only one of these parameters shall be specified in any given operation, and except for value, these fields are not persisted. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response."

- 7 Section 9.4, Example 9-3, is missing container trailing / for container name: request should be 'GET /MyContainer/ HTTP/1.1'; "objectURI" field should be "/MyContainer/"; "objectName" field should be "MyContainer/"

DS: Suggest TWG approve change.

- 8 Section 9.4, Example 9-5, text "...to read children 0..2...", should be "...to read children 0..2 and childrenrange..."

DS: Suggest TWG approve change.

- 9 Section 9.5, Request Message Body section, specifies the response statuses for optional/undefined fields, shouldn't this be in specified in the Response Status table?

This is referring to the Response Message Body section.

DS: Suggest TWG approve change.

Proposed spec text change:

Move the below two lines to the "Response Status:" section, above the table.

"Requesting an optional field that is not present results in a 404 Not Found HTTP status code."

"Requesting an undefined field results in a 400 Bad Request HTTP status code."

- 10 Section 9.5, Request Message Body section and Response Status table have conflicting scenarios for returning a 404, in one case, it's if a field is not found and the other if a container is not found at the request URI (the latter seems correct)

The ?nnn is addressing a sub-entity, so if the sub-entity does not exist, the standard convention is to return a 404.

- 11 Section 9.5, Example 9-7, text "...to read children 0..2...", should be "...to read children 0..2 and childrenrange..."

DS: Suggest TWG approve change.

- 12 Section 9.6, After "Delayed completion of a snapshot:", the first sentence should be a new paragraph .

DS: Suggest TWG approve change.

- 13 Section 9.8, Request Message body table, 'reference' statement that "If other fields from this table ..." should be removed from the field description, as this is not mentioned for any other fields and is already stated in the table footer.

This is more restrictive than the statement in the table footer, which only applies to the starred fields. The restriction in the reference row applies to all other fields in the table.

DS: Recommend leave as-is.

- 14 Section 9.8: The spec erroneously lists an Accept header in the Response Headers table on page 70. This should be a Content-Type header instead, as correctly reflected in the following examples.

DS: Suggest TWG approve change.

- 15** Section 9.8, Example 9-11, response is missing 'Location: ' header-name text before the URL is given.
 DS: Suggest TWG approve change.
 Replace:
`"http://cloud.example.com/MyContainer/
 0000706D0010B84FAD185C425D8B537E"`
 with:
`"Location: http://cloud.example.com/MyContainer/
 0000706D0010B84FAD185C425D8B537E"`
- 16** Section 9.10, Example 9-15 omits the Content-Type header required as its request body features an empty JSON object. Empty JSON object is valid as a request body. Content-Type header should be added.
 DS: Suggest TWG approve change.
 — In Example 9-15, add the following line above the "Accept" line:
`"Content-Type: application/cdmi-queue"`
- 17** The section titles are inconsistent; some use "Container Object" (e.g., Section 9.4, "Read a Container Object") while some use just "Container" (e.g., Section 9.6 "Update a Container"). I'd suggest standardizing on "Container Object"
 DS: Suggest TWG approve change.
 — Change:
 9.2 Create a Container (CDMI Content Type)
 to
 9.2 Create a Container Object (CDMI Content Type)
 — Change:
 9.3 Create a Container (Non-CDMI Content Type)
 to
 9.3 Create a Container Object (Non-CDMI Content Type)
 — Change:
 9.6 Update a Container (CDMI Content Type)
 to
 9.6 Update a Container Object (CDMI Content Type)

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes, **except to Item #10**, where no change was indicated.

Trac Ticket #444 Use of optional fields in object update (PUT) request URI is unclear

Description: The object update (PUT) operations for all objects support optionally specifying the fields to be updated in the URI of the request, for example in the form:

- PUT /object
- PUT /object?metadata
- PUT /object?<fieldname>

This method of invocation seems both:

- unnecessary -- all fields are listed as optional in the request message body specification and a field that is not included in the message body is not changed by the update operation.
- unclear and confusing -- the relationship and behaviour of the request URI versus the request message body is unclear (e.g., what happens when the message body includes fields that are not present in the URI or does not include fields that are present in the URI?)

This behavior needs clarification.

It seems most straightforward for the scope of the update to be exclusively specified by the fields present in the request message body, and to remove support for specifying fieldnames in the request URI of the PUT operation.

This issue applies to the following sections:

- Section 8.6: Update a Data Object
- Section 9.6: Update a Container
- Section 10.4: Update a Domain
- Section 11.4: Update a Queue Object

I agree with this point. Also note pending trac ticket #419, which affects the synopsis and describes what shall happen if metadata items are included in the URI.

WARNING - Remember that Trac will mangle the below text. Use the e-mail notification for making the update.

Assuming that #419 is accepted, the proposed spec text change is:

1 In Section 8.6 Update a Data Object (CDMI Content Type) synopsis, replace:

```
PUT <root URI>/<ContainerName>/<DataObjectName>
PUT <root URI>/<ContainerName>/<DataObjectName>?metadata
PUT <root URI>/<ContainerName>/
  <DataObjectName>?metadata:<metadataname>;...
PUT <root URI>/<ContainerName>/<DataObjectName>?value
PUT <root URI>/<ContainerName>/<DataObjectName?>value:<range>
PUT <root URI>/<ContainerName>/<DataObjectName?><fieldname>
with:
PUT <root URI>/<ContainerName>/<DataObjectName>
```

```
PUT <root URI>/<ContainerName>/
  <DataObjectName>?metadata:<metadataname>;...
PUT <root URI>/<ContainerName>/<DataObjectName>?value:<range>
```

2 In Section 9.6 Update a Container (CDMI Content Type) synopsis, replace:

```
PUT <root URI>/<ContainerName?>/<TheContainerName?>/
PUT <root URI>/<ContainerName?>/<TheContainerName?>/
  ?metadata:<metadataname>;...
PUT <root URI>/<ContainerName?>/<TheContainerName?>/?metadata
```

with:

```
PUT <root URI>/<ContainerName?>/<TheContainerName?>/
PUT <root URI>/<ContainerName?>/<TheContainerName?>/
  ?metadata:<metadataname>;...
```

3 In Section 10.4 Update a Domain (CDMI Content Type) synopsis, replace:

```
PUT <root URI>/cdmi_domains/<DomainName?>/<TheDomainName?>/
PUT <root URI>/cdmi_domains/<DomainName?>/<TheDomainName?>/
  ?metadata
PUT <root URI>/cdmi_domains/<DomainName?>/<TheDomainName?>/
  ?metadata:<metadataname>;...
```

with:

```
PUT <root URI>/cdmi_domains/<DomainName?>/<TheDomainName?>/
PUT <root URI>/cdmi_domains/<DomainName?>/<TheDomainName?>/
  ?metadata:<metadataname>;...
```

4 In Section 11.4 Update a Queue Object (CDMI Content Type) synopsis, replace:

```
PUT <root URI>/<ContainerName?>/<QueueName?>
PUT <root URI>/<ContainerName?>/<QueueName?>?metadata
PUT <root URI>/<ContainerName?>/
  <QueueName?>?metadata:<metadataname>;...
```

with:

```
PUT <root URI>/<ContainerName?>/<QueueName?>
PUT <root URI>/<ContainerName?>/
  <QueueName?>?metadata:<metadataname>;...
```

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #445 Corrections to Chapter 10: Domain Object Resource Operations

Description: Minor corrections and suggestions for Chapter 10: Domain Object Resource Operations in revision 1.0.1h:

Subclause 10.1:

- The section titles are inconsistent, some use "Domain Object" (e.g., Section 10.3, "Read a Domain Object") while some use just "Domain" (e.g., Section 10.4, "Update a Domain"). I'd suggest standardizing on "Domain Object" for all section titles.

TWG: Marie to check with Ralph's comments.

- 2 Section 10.1.2, paragraph 3, '[ISO-8601]' link is not active'
TWG: Marie to check link.
- 3 Section 10.1.2, Table 7, `cdmi_summary_bytesaverage`, typo in description '.during'
TWG: Marie to fix.
- 4 Section 10.1.2, states All time and duration values, unless otherwise specified, are in the [ISO-8601] extended representation (YYYY-MM-DDThh:mm:ss.ssssssZ). -- it should be clearly stated that this does not apply to the directory structure, but to the JSON Metadata elements of the summary object defined in Table 7, and that the directory structures for domain summaries follow a YYYY, YYYY-MM, and YYYY-MM-DD notation

TWG: We added a new section 5.14, so this restriction on time format can be removed from the domains section.
Proposed spec change: Remove paragraph starting with "All time and duration values..." below the example listing of domain summary URIs.
- 5 Section 10.1.2 and Section 10.1.3, Table 7, Table 8 and example are inconsistent with naming convention, '`cdmi_domainURI`', '`cdmi_delegation_URI`', '`cdmi_member_uri`' etc.

TWG: Couldn't find any inconsistencies in Table 7 and the corresponding example.
TWG: In section 10.1.3, in the example of a delegation, "`cdmi_member_uri`" should be "`cdmi_member_URI`".
- 6 Section 10.1.2, Table 7 - the unit of measurement (kilowatt) for `cdmi_summary_kw(hours|min|max|average)` should be listed/explained in the descriptive field

TWG: In Table 7, for "`cdmi_summary_kwhours`", replace "power consumed" with "power consumed (in kilowatt-hours)"
For "`cdmi_summary_kwmin`", replace "power consumed" with "power consumed (in kilowatts)"
For "`cdmi_summary_kwmax`", replace "power consumed" with "power consumed (in kilowatts)"

TWG: From a convention standpoint, with the exception of objectID and *URI, we are using all lower case for field names. We have two choices, we could continue with mixed-case field names, and change the above to `_kWmax`, etc, or we could state that all of the CDMI-defined field names are all lower case.
DS will need to pull this out into another ticket, since there was no response to the above question.
- 7 Section 10.1.3 first example uses '`cdmi_groups`', which is not mentioned in Table 7. This should presumably be "`cdmi_member_groups`"

TWG: The example should use "`cdmi_member_groups`", not "`cdmi_groups`".
- 8 Section 10.1.3 second example uses '`cdmi_member_uri`', which is not mentioned in Table 7 (perhaps should be `cdmi_delegation_URI`? `cdmi_member_uri` seems to fit better)

TWG: The example should use "`cdmi_delegation_URI`", not "`cdmi_member_uri`".
- 9 Section 10.1.3, Table 8, `cdmi_member_credentials` does not mention how the authentication transaction is performed or how the credentials should be transformed prior to storing/comparing them in the domain membership object

(presumably, authentication credentials should not be stored in clear text, but this is not explicitly stated).

TWG: This is intentional. The goal of the spec is to indicate where this information can be stored, but not define how it is used when resolving the credentials of a user to determine their ACL Name.

Subclause 10.2:

- 1 Section 10.2, Capabilities, "cdmi_domain_create" container capability does not exist in Section 12.1.5 Table 14 (and should probably be called "cdmi_create_domain" to be consistent with other create container capabilities)
TWG: This should be "cdmi_create_domain". This capability exists in Table 15, Capabilities for Domains.
Spec change: Replace "cdmi_domain_create" with "cdmi_create_domain".
- 2 Section 10.2, Examples 10-1, 10-2, and 10-3: response objectName, objectURI, domainURI, parentURI should have trailing /, as they are container objects
TWG: Add trailing slashes for objectURI, objectName, parentURI, domainURI and capabilitiesURI to examples 10-1, 10-2 and 10-3.
- 3 Section 10.2, Example 10-1, response has childrenrange set to '1-2', but should be '0-1'
TWG: Fix to be "0-1".

Subclause 10.3:

- 1 Section 10.3, Synopsis, the base URI "cdmi_domain/" should be "cdmi_domains/"
TWG: This should be fixed.
- 2 Section 10.3, Example 10-2, the request URI "/MyDomain/" should be "/cdmi_domains/MyDomain/"
TWG: This should be fixed.

Subclause 10.5:

- 1 Section 10.5, Response Status table suggests returning a 409 conflict if the specified newDomain is invalid or unusable; this code seems inappropriate for the situation and should probably return a 400 Bad Request (the request message body description indicates that the newDomain URI must be valid, so the request is malformed if this is otherwise).
TWG: Create new ticket for consistent use of the 409 result code throughout the spec.
- 2 Section 10.5, Example 10-6, newDomain is missing trailing /, should be "/cdmi_domains/secondDomain/"
TWG: Add missing "/".
- 3 Section 10.5, capabilities refers to "cdmi_domain_delete" capability which does not exist in Chapter 12, should be "cdmi_delete_domain"
TWG: Replace "cdmi_domain_delete" with "cdmi_delete_domain", which is defined in Table 15.
- 4 Section 10.5, Request Headers table is missing Content-type header, which is required since a request message body is present.

TWG: Add Content-Type back in to the Request Headers table. Already present in the example.

- 5 Section 10.5 overloads the DELETE method with content, which contradicts all other usages of DELETE in specification. Although RFC 2616 does not explicitly forbid usage of a request body in DELETE requests, HTTP proxies might remove the body on forwarding, and some HTTP client/server libraries may not support this, so this might be worth reconsidering.

TWG: Pull this out into a separate ticket.

Action items: DS to pull out separate tickets.

TWG: Consensus is to approve other changes that aren't being pulled out into separate tickets. DS will assign to Marie once separate tickets have been created.

Extracted tickets to #526, #527 and #528.

Assigning remainder of times to Marie.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #446 Corrections to Chapter 11: Queue Object Resource Operations

Description: Minor corrections to Chapter 11: Queue Object Resource Operations in revision 1.0.1h:

- 1 Section 11.1 : The sentence "a queue reader GETs objects from the queue, acknowledging the receipt of the last object" implies that a GET will dequeue objects (i.e., read and delete), which is not correct as removal requires a separate DELETE operation. Perhaps restate along the lines of: "a queue reader GETs objects from the queue, and DELETES from the queue to acknowledge receipt of the objects".

TWG: Replace:

"a queue reader GETs objects from the queue, acknowledging the receipt of the last object"

with

"a queue reader GETs value(s) from the queue, and subsequently DELETES the value(s) to acknowledge receipt of the value(s)"

- 2 Section 11.2, Response Message Body table, for queueValues, does the possible overflow need to be taken into account if this count is always incrementing?

TWG: Overflows could happen if an implementer used 32 bit numbers to store the count. We should add a recommendation to ensure sufficient precision to avoid overflows within the lifetime of the objects being managed.

Spec change:

Add the following sentence to the end "queueValues" row description in the response body table in sections 9.10, 11.2 and 11.3:

"Clients should ensure that they use sufficient precision (e.g. 64-bits) when storing these values in an internal representation to avoid overflow."

- 3 Section 11.3, Response Message Body table, for mimetype it states that "If two or more values are requested, the MIME types of the values are returned, each corresponding to the value in the same..." -- this seems like an artifact from an earlier revision when a single value in the response was not encapsulated in a JSON array. It should be simplified to just: "The MIME types of the values are returned, each..." since the format is the same regardless of whether there is one or more than one returned value.

TWG: Replace

"If two or more values are requested, the MIME types of the values are returned, each corresponding to the value in the same..."

with

"The MIME types of the values are returned, each corresponding to the value in the same..."

- 4 Section 11.3: Response Message Body table, "wimplementor" should be "implementor"

TWG: Marie to change.

- 5 Section 11.6, Request Message Body table, the "move" and "copy" descriptions should clarify that this operation can be used to move or copy a subset of values from a queue (e.g., as shown in Example 11-10). It should also clarify whether or not a move/copy source queue URI with no "values:<count>" field moves/copies all values from the queue.

TWG: There is another ticket that captures this. DS to verify, and create separate ticket if not already present.

- 6 Section 11.6, Example 11-10 has a typo in the "move" URI -- "/MyContainer/FirstQueue;values:20" should be "/MyContainer/FirstQueue?values:20"

TWG: Typo should be corrected.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #447 Corrections to Chapter 12: Capability Object Resource Operations

Spec text changes for Ticket #447

12.1

- 1 Section 12.1, paragraph 4, replace:

"Cloud clients may use capabilities to discover what operations are supported. If an operation is attempted to be performed against a CDMI object that does not have a corresponding capability, an HTTP 405 status code shall be returned to the client. All CDMI-compliant cloud storage systems shall implement the ability to list capabilities, but support for all other capabilities is optional."

with

"Cloud clients may use capabilities to discover what operations are supported. If an operation is attempted to be performed against a CDMI object that does not have a corresponding capability, an HTTP 400 status code shall be returned to the client. All CDMI-compliant cloud storage systems shall implement the ability to list capabilities, but support for all other capabilities is optional."

- 2 Section 12.1, paragraph 6, replace:
"The base set of CDMI capabilities are based on the operations defined in the previous subclauses, with additional cloud-specific capabilities added based on use cases for standard cloud storage. The hierarchy of capabilities (see Figure 7) shows the hierarchy of capabilities in an offering and shows how the capabilities-URI links data objects and containers into the capabilities tree."
with
"The base set of CDMI capabilities are based on the operations defined in the previous subclauses, with additional cloud-specific capabilities added based on use cases for standard cloud storage. Figure 7 shows the hierarchy of capabilities in an offering and shows how the capabilitiesURI links data objects and containers into the capabilities tree."
- 3 Section 12.1, figure 7, make the following changes to the diagram:
"mydataobject/" -> "mydataobject"
"myimmutabledataobject/" -> "myimmutabledataobject"
"myqueue/" -> "myqueue"
"gold_container" -> "gold_container/"
"immutable" -> "immutable"

12.1.1

- 1 Section 12.1.1, table 102, add a row below the "cdmi_export_cifs" row:
cdmi_dataobjects,
JSON String,
If present and "true", this capability indicates that the cloud storage system supports data objects.
- 2 Section 12.1.1, table 102, "cdmi_metadata_maxitems" row, replace the description:
"If present, this capability specifies the maximum number of user- defined metadata items supported by the cloud storage system. If absent, there is no limit placed on the number of user-defined metadata items."
with
"If present, this capability specifies the maximum number of user-defined metadata items supported per object. If absent, there is no limit placed on the number of user-defined metadata items."
- 3 Section 12.1.1, table 102, "cdmi_metadata_maxsize" row, replace the description:
"If present, this capability specifies the maximum size in bytes of each user- defined metadata item supported by the cloud storage system. If absent, there is no limit placed on the size of user- defined metadata items."
with
"If present, this capability specifies the maximum size in bytes of each user- defined metadata item supported per object. If absent, there is no limit placed on the size of user- defined metadata items."
- 4 Section 12.1.1, table 102, "cdmi_query_contains" row, replace the description:

"If present and "true", this capability indicates that the cloud storage system supports query with the "contains" expressions."

with

"If present and "true", this capability indicates that the cloud storage system supports query with "contains" expressions."

- 5 Section 12.1.1, table 102, "cdmi_serialization_json" row, replace the description:

"If present and "true", this capability indicates that the cloud storage system supports json as a serialization format."

with

"If present and "true", this capability indicates that the cloud storage system supports JSON as a serialization format."
- 6 Section 12.1.1, table 102, remove the "cdmi_xmlrepresentation" row.
- 7 Section 12.1.1, table 103, remove the second "cdmi_acl" row (the last row).
- 8 Section 12.1.1, table 103, make the following changes to the description of each of the below rows:

"...generate a "size" storage..." -> "...generate a "cdmi_size" storage..."

"...generate a "ctime" storage..." -> "...generate a "cdmi_ctime" storage..."

"...generate a "atime" storage..." -> "...generate a "cdmi_atime" storage..."

"...generate a "mtime" storage..." -> "...generate a "cdmi_mtime" storage..."

"...generate a "acount" storage..." -> "...generate a "cdmi_acount" storage..."

"...generate a "mcount" storage..." -> "...generate a "cdmi_mcount" storage..."
- 9 Section 12.1.1, table 103, remove the "cdmi_hash" row.
- 10 Section 12.1.1, table 104, "cdmi_data_dispersion" row, replace the description:

"If present and "true", this capability indicates that the cloud storage system shall disperse data. If absent, redundancy copies specified shall be ignored."

with

"If present and "true", this capability indicates that the cloud storage system shall disperse data. If absent, the dispersion specified shall be ignored."
- 11 Section 12.1.1, table 104, "cdmi_hash_value" row, replace the description:

"This metadata is used to indicate if the object data is to be hashed and indicates the desired hash algorithm and length. Supported algorithm/length values are provided by the cdmi_value_hash capability."

with

"If present, this capability lists the hash algorithms and lengths supported. If absent, objects shall not be hashed. When a CDMI implementation supports hashing, the system-wide capability of "cdmi_security_data_integrity" specified in Table 102 of 12.1.1 shall be set to "true". Otherwise, it shall not be present, indicating that there is no support for hashing."

12.1.1

- 1 Section 12.1.1, table 106, "cdmi_create_container" row, replace the description:

"If present and "true", this capability indicates that the container allows a new container may be added."

with

"If present and "true", this capability indicates that the container allows a new container to be created via PUT."

- 2 Section 12.1.1, table 106, "cdmi_create_reference" row, replace the description:

"If present and "true", this capability indicates that the container allows a new child reference may be added."

with

"If present and "true", this capability indicates that the container allows a new child reference to be created via PUT."
- 3 Section 12.1.1, table 107, "cdmi_create_container" row, replace the description:

"If present and "true", this capability indicates that the domain allows a new container may be added."

with

"If present and "true", this capability indicates that the domain allows a new container to be created via PUT."
- 4 Section 12.1.1, table 107, "cdmi_copy_domain" row, replace the description:

"If present and "true", this capability indicates that the container may be copied (via PUT) to another URI."

with

"If present and "true", this capability indicates that the domain may be copied (via PUT) to another URI."

12.1.7

- 1 Section 12.1.7, Table 108, "cdmi_read_value" row, remove the extra space after queue. "queue 's" -> "queue's"
- 2 Section 12.1.7, Table 108, "cdmi_modify_value" row, remove the extra space after queue. "queue 's" -> "queue's"
- 3 Section 12.1.7, Table 108, "cdmi_modify_value" row, replace the description:

"If present and "true", this capability indicates that the queue 's value and meta-data may be modified."

with

"If present and "true", this capability indicates that the queue's value may be modified."

Other Changes

- 1 Section 12.2.1, URIs, replace:

"{range}"

with

"<range>"
- 2 Section 12.2.8, Examples 1, 2 and 3, Response Headers, add "Content-Type: application/cdmi-capability" before the X-CDMI-Specification-Version header.
- 3 Globally replace all instances of "cdmi_objectID" with "cdmi_objectid" (to ensure this is all lower case).

- 4 Section 12.2.6, Table 111, Row "capabilities", remove the extra blank line at the top of the description.
- 5 Section 12.2.2, replace:
 - "• All CDMI implementations shall permit clients to read the metadata and contents of all capabilities objects."
 - with
 - "• All CDMI implementations shall permit clients to read all fields of all capabilities objects."
- 6 Section 3, add entries for "RPO" and "RTO".

TWG has approved these changes to the spec, and assigned them to Marie to incorporate.

CHANGES: Made the specified changes to Version 1.0.1k.

Trac Ticket #448 Corrections to Chapter 13: Exported Protocols

Description: Minor corrections to Chapter 13: Exported Protocols in revision 1.0.1h:

- 1 Section 13.0, Figure 9, there is a typo in "CDMI Exported Prtocolcs" -- "Protocols"
- 2 Section 13.0, the green reference link to "OCCI" in the paragraph below Figure 9 does not appear to link to anything.
- 3 Section 13.2, in "(i.e., OCCI/NFSv4)" change "i.e" to "e.g."
- 4 Section 13.2, the example JSON should be split into multiple lines for readability and consistency with other JSON examples in the document.
- 5 Section 13.2, in the example JSON should "permissions" actually be a JSON array of IDs, as stated in the preceding text?
 - TWG: Already is JSON array of IDs.
- 6 Section 13.3.1, in the example JSON, remove the comma from "iqn.2010-01.com.acme:host2," and align the spacing of the two array entries
- 7 Section 13.3.2, the paragraph should start off with "The following code" as opposed to "The follow code"
- 8 Section 13.4, this section is marked incomplete. Is it intended to be completed in version 1.0 of the specification or a later version?
 - TWG: Will be covered in Ralph's comments.
- 9 Section 13.5, in "access through the CDMI" remove "the"

Reviewed on 2011-05-20 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made specified changes except for Items 5 and 8. Validated all JSON examples and made them consistent with other JSON examples.

Trac Ticket #449 Clarify whether `cdmi_snapshots` is a reserved object name

Description: In Sections 9.6 and 14, it describes that snapshots, if supported by the CDMI server, are created under a "cdmi_snapshots" child container of the container receiving the snapshot.

The spec should clarify if and under what conditions "cdmi_snapshots" is a reserved name. E.g., if you cannot create any object named "cdmi_snapshots" under a container, or if this is only applicable when a CDMI server supports snapshots, etc.

If this name is reserved, it (and any other reserved object names, although I did not observe any) should probably be clearly stated somewhere in an earlier section of the spec (Chapter 7?).

TWG: Updated proposal:

Add a new section before 9.1.2 (so this would be a new 9.1.2):

9.1.2 Reserved Container Names

This specification defines reserved container names that shall not be used when creating new containers. These container names are reserved for use by this specification and if an attempt is made to create or delete them, an HTTP 400 Bad Request result code shall be returned to the client.

The reserved container names include:

- `cdmi_objectid`
- `cdmi_domains`
- `cdmi_capabilities`
- `cdmi_snapshots`
- `cdmi_versions`

As additional names may be added in future versions of the specification, server implementations shall prevent the creation of user-defined containers if the container name starts with "cdmi_".

CHANGES Added a new subclause 9.1.2 as specified to Version 1.0.1n.

Trac Ticket #450 Corrections to Chapter 15: Serialization/Deserialization

Description: Minor corrections to Chapter 15: Serialization/Deserialization in revision 1.0.1h:

- Section 15.1, paragraph 2, the sentence "..selected data object, container, and children or queue." has a comma in the wrong place, it should be "...Selected data object, container and children, or queue."

- Section 15.2, pages 134-135, both "cross-domain" and "cross_domain" are used. "cross_domain" is presumably the correct form, as used in the rest of the document.
- Section 15.5.2, the "valuerange" of "This is the Value of this Data Object" should be 0-36 (the string is 37 characters)
- Section 15.5.2, the second "valuerange" in the serialized queue is incorrect -- "0-4" should be "0-3" ("blue" is four characters)

Proposed spec changes:

- 1 Section 15.1, paragraph 2, the sentence "..selected data object, container, and children or queue." has a comma in the wrong place, it should be "...Selected data object, container and children, or queue."
DS: Suggest TWG approve change.
- 2 Section 15.2, pages 134-135, both "cross-domain" and "cross_domain" are used. "cross_domain" is presumably the correct form, as used in the rest of the document.
DS: Suggest TWG approve change.
- 3 Section 15.5.2, the "valuerange" of "This is the Value of this Data Object" should be 0-36 (the string is 37 characters)
DS: Suggest TWG approve change.
- 4 Section 15.5.2, the second "valuerange" in the serialized queue is incorrect -- "0-4" should be "0-3" ("blue" is four characters)
DS: Suggest TWG approve change.

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #451 Corrections to Chapter 16: Metadata - Access Control

Description: Minor corrections to Section 16.1: Access Control in revision 1.0.1h:

- 1 Section 16.1 Access Control should include a reference to the NFSv4 [RFC3530], given this section seems to be largely based on the NFSv4 ACL specification.
TWG Decision: Update text in clause 16.1 to indicate that the ACL implementation is largely taken from RFC 3530.
Replace text in 16.1
"Access control comprises the mechanisms by which various types of access to objects and containers are authorized and permitted or denied. CDMI uses the well-known mechanism of an Access Control List (ACLs). ACLs are lists of permissions-granting or permissions-denying entries called access control entries (ACEs)."
with
"Access control comprises the mechanisms by which various types of access to objects and containers are authorized and permitted or denied. CDMI uses the

well-known mechanism of an Access Control List (ACLs) as defined in the NFSv4 standard [RFC 3530]. ACLs are lists of permissions-granting or permissions-denying entries called access control entries (ACEs)."

- 2 Section 16.1.1: States that "ACEs are composed of five fields." - The "C struct" only contains four members (type, who, flags, access_mask). The fifth field ("acetime") seems to be missing

TWG Decision: "ACEs are composed of five fields." change to "ACEs are composed of four fields."

- 3 Section 16.1.1: The second "typedef char utf8_t;" is not used in the following "C struct" or any other (pseudo-)code on the following pages.

TWG Decision: Verified not used. Can remove.

Replace:

```
typedef unsigned int uint_t;
typedef char utf8_t;
typedef char* utf8str_t;
with
typedef unsigned int uint_t;
typedef char* utf8str_t;
```

- 4 Section 16.1.2, 16.1.4, 16.1.5 -- if the pseudo-code is intended to actually be valid, compilable C, a type is needed ("const uint_t" instead of "const"), and the string form should be commented (prefixed with "// ")

TWG Decision: We haven't reviewed Ralph's comments on this section yet. Changes made could be incorporated by Marie before she processes this section.

Replace:

```
const CDMI_ACE_ACCESS_ALLOW = 0x00000000; "ALLOW"
const CDMI_ACE_ACCESS_DENY = 0x00000001; "DENY"
const CDMI_ACE_SYSTEM_AUDIT = 0x00000002; "AUDIT"
with:
const uint_t CDMI_ACE_ACCESS_ALLOW = 0x00000000; // "ALLOW"
const uint_t CDMI_ACE_ACCESS_DENY = 0x00000001; // "DENY"
const uint_t CDMI_ACE_SYSTEM_AUDIT = 0x00000002; // "AUDIT"
```

- 5 Same changes to be made to sections 16.1.4, 16.1.5

TWG will reconvene to review the remaining items in this ticket when Alan returns.

- 6 Section 16.1.3 ACE Who, "DNS domain" links to the CDMI "domain" terminology definition, which does not apply to the term "domain" in the context of DNS.

TWG: This has been fixed in 1.0.1k. (link removed)

16.1.4

- 1 Section 16.1.4 is "CDMI_ACE_FLAGS_INHERITED_ANCE" listed in error? Its bitwise number does not seem to be sequential with the rest of the list, it is missing a string form, and its purpose is not defined.

TWG: This is used by Microsoft to indicate if the ACE is inherited, and thus can be removed when explicitly set by the client.

We leave guidance on implementation to the NFSv4 spec.

No changes made to spec.

- 2 Section 16.1.4, the sentence "An ACE on which CDMI_ACE_FLAGS_NO_PROPAGATE_ACE is not inherited..." is missing some words ("on which ... is set")

TWG: In section 16.1.4, change:

"An ACE on which CDMI_ACE_FLAGS_NO_PROPAGATE_ACE is not inherited by any objects or subcontainers. It applies only to the object or container on which it is set."

to

"An ACE on which CDMI_ACE_FLAGS_NO_PROPAGATE_ACE is set is not inherited by any objects or subcontainers. It applies only to the object or container on which it is set."

- 3 Section 16.1.4, the sentence "An ACE on which CDMI_ACE_FLAGS_INHERIT_ONLY_ACE child objects or subcontainers." is missing some words ("on which ... is set is inherited by and only applies to...")

TWG: In section 16.1.4, change:

"An ACE on which CDMI_ACE_FLAGS_INHERIT_ONLY_ACE child objects or subcontainers. It is ignored when evaluating access for the object or container on which it is set. This flag is particularly useful for top-level containers, which may specify any kind of default permissions for child objects, but themselves may only be writable by their owners or administrators, for instance."

with

"The CDMI_ACE_FLAGS_INHERIT_ONLY_ACE flag applies to child objects or subcontainers. It is ignored when evaluating access for the object or container on which it is set. This flag is particularly useful for top-level containers, which may specify any kind of default permissions for child objects, but themselves may only be writable by their owners or administrators, for instance."

16.1.5

- 1 Section 16.1.5, "their values are taken from the IETF NFSv4 RFC 3530" might be better worded as "based on", given that the names differ somewhat from NFSv4 and CDMI includes a "SET_RETENTION" bit which is not present in NFSv4.

TWG: Values are taken directly from the RFC. No change required.

A new ticket will be created for the retention mask bits. (#590)

- 2 Section 16.1.5, the definition of RW_ALL = 0x0006006F does not match (READ_ALL|WRITE_ALL) = (0x00020089|0x00040156) = 0x000601DF

TWG: The RW_ALL definition appears to be incorrect.

Replace:

```
"const CDMI_ACE_RW_ALL = 0x0006006F; "RW_ALL"
```

with

```
"const CDMI_ACE_RW_ALL = 0x000601DF; "RW_ALL"
```

- 3 Section 16.1.5, the definition of ALL = 0x1007FFFF excludes "SYNCHRONIZE" permission = 0x00100000 -- is SYNCHRONIZE explicitly excluded, as with WRITE_OWNER? A clarifying comment should be added, if so

TWG:

Spec text changes:

- Delete the lines corresponding to "CDMI_ACE_WRITE_OWNER" and "CDMI_ACE_SYNCHRONIZE" from section 16.1.5.
- Delete the following paragraph from section 16.1.5:

"Note that CDMI_ACE_WRITE_OWNER is not included in CDMI_ACE_WRITE_ALL or CDMI_ACE_ALL. This permission, to change the owner of an object, is generally only given to admins and the object's owner."

16.1.6

- 1 Section 16.1.6, perhaps "PUT and GOTTEN" should be "stored and retrieved"

TWG: Replace:

"As ACLs are metadata, they are PUT and GOTTEN through the metadata field of a PUT or GET request. The syntax is as follows:"

with

"As ACLs are metadata, they are stored and retrieved through the metadata field included in a PUT or GET request. The syntax is as follows:"
- 2 Section 16.1.6, the block following "The syntax is as follows" omits some access bits (e.g., EXECUTE, SYNCHRONIZE, SET_RETENTION). The preceding default ACL example is also inconsistent with this syntax definition (e.g., "ALLOW" instead of "CDMI_ACE_ACCESS_ALLOWED_TYPE")

TWG: We are missing this, and that prevents round trip ACL access. We have removed SYNCHRONIZE, so we need to add EXECUTE and SET_RETENTION. We are also missing the Read/Write Attribute ACE mask flags. Make the following spec changes:

 - In the BNF on Page 186 (Section 16.1.6), change:

"acemaskitem = acereaditem | acewriteitem | aceappenditem | acereadmetaitem | acewritermetaitem | acedeleteitem | acedelfselfitem | acereadaclitem | acewriteraclitem | acewriterowneritem"

to

"acemaskitem = acereaditem | acewriteitem | aceappenditem | acereadmetaitem | acewritermetaitem | acedeleteitem | acedelfselfitem | acereadaclitem | acewriteraclitem | aceexecuteitem | acereadattritem | acewriterattritem | aceretentionitem"
 - In the BNF on Page 186 (Section 16.1.6), replace:

"acewriterowneritem= "CDMI_ACE_WRITE_OWNER" | 0x80000"

with

"aceexecuteitem = "CDMI_ACE_EXECUTE" | 0x00020"
 - In the BNF on Page 186 (Section 16.1.6), at the end of the BNF, add the following lines:

"acereadattritem = "CDMI_ACE_READ_ATTRIBUTES" | 0x00080"

"acewriterattritem= "CDMI_ACE_WRITE_ATTRIBUTES" | 0x00100"

"aceretentionitem= "CDMI_ACE_SET_RETENTION" | 0x10000000"

- 3 Section 16.1.6: EBNF does not define "acetime" which seems to be the mysteriously missing fifth element in the C struct of 16.1.1.

TWG: acetime has been removed.

16.1.7

- 1 Section 16.1.7 in CDMI_ACE_ALL is defined as 0x1007FFFF in Section 16.1.5, which does not equal "0x000FFFFFF" as used here

TWG: In section 16.1.7, replace:

"0xFFFFF

evaluates to 0x000FFFFFF == CDMI_ACE_ALL"

with

"0x1007FFFF

evaluates to 0x1007FFFF == CDMI_ACE_ALL"

- 2 Section 16.1.7 "RW_ALL" | DELETE according to Section 16.1.5 actually evaluates to 0x0006006F | 0x00010000 = 0x0007006F (or 0x000701DF if the previously noted correction to RW_ALL is applied)

TWG: In section 16.1.7, replace:

""RW_ALL" | DELETE

evaluates to 0x0060006F | 0x00100000 == 0x0070006F"

with

""RW_ALL" | DELETE

evaluates to 0x000601DF | 0x00100000 == 0x000701DF"

16.1.9

- 1 Section 16.1.9 should clearly state if the hexadecimal format is the only supported format for JSON representations of ACLs, and the string equivalents of type/flags/mask are not supported. The hexadecimal format should also be padded out to show a 32-bit integer, i.e. 0xnccccccc

TWG: Viewed to be good as is. No changes to spec required.

Desire is to allow hex representations that are shorter than 10 characters, eg, "0x2"

- 2 Section 16.1.9 the JSON example seems to be missing the required "acetime" field

TWG: acetime has been removed. No changed to spec text needed.

- 3 Section 16.1.9 the JSON example has extra-commas after the "acemask" : "0xnn", members of its objects.

TWG: Marie to run JSON in section 16.1.9 through JSON lint (www.jsonlint.com) to get rid of extra commas.

- 4 Section 16.1.9 in the example GET response "/cdmi_capabilities/DataItem/" should be "/cdmi_capabilities/dataobject/", and the valuerange "0-17" should be "0-16"

TWG: Marie to apply above changes to spec text.

- 5 Section 16.1.9 the CDMI response example has an extra-comma after the "acemask" : "0x0020089", member of the JSON object within the "cdmi_acl" array.

TWG: See above comment about running JSON through jsonlint.

CHANGES Made the specified changes to 1.0.1l.

Trac Ticket #452 Corrections to Chapter 16: Metadata

Description: Some minor comments and clarifications for Chapter 16: Metadata in revision 1.0.1h.

Proposed Spec text changes:

16.2

Section 16.2, the description of the "cdmi_metadata_maxitems" capability should clarify that this limit is *per object*.

Already fixed — no change needed.

16.3

- 1 Section 16.3, Table 19 includes a "cdmi_owner" SSMD, but there is no corresponding capability in Section 12.1.2, Table 11.
This SSMD is mandatory to implement, thus there is no capability needed.
- 2 Section 16.3, Table 19 should clarify the definition of "access" for "cdmi_atime", i.e., if this is the same as POSIX semantics, where a read (data/metadata) updates atime but a write does not update atime. The description of cdmi_acount implies that "access" includes writes.
Proposal is to leave spec text as-is.
- 3 Section 16.3, Table 19 "cdmi_mtime" should indicate if metadata/field updates also update the cdmi_mtime
Proposal is to leave spec text as-is.
- 4 Section 16.3, Table 19 "cdmi_acount" and "cdmi_mcount" - what is the maximum envisioned (unsigned) integer size of these numbers and what are the overflow/ wrap-around considerations?
Proposal is to leave spec text as-is.
- 5 Section 16.3, Table 19 "cdmi_acl" should be classified as "JSON Array of JSON Objects" to reflect the mapping of ACL/ACEs
In Table 115, row "cdmi_acl", replace "JSON Array" with "JSON Array of JSON Objects".

16.4

- 1 Section 16.4, Table 20 includes "cdmi_immediate_redundancy", "cdmi_assignedsize", and "cdmi_geographic_placement" DSMD but there are no corresponding capabilities in Section 12.1.3, Table 12.
Add the following rows to table Table 104 - Capabilities for Data System Metadata:
— cdmi_immediate_redundancy
JSON String

If present, this capability specifies the maximum number of immediate redundancy copies that may be specified. If absent, immediate redundancy copies specified shall be ignored.

— `cdmi_assignedsize`

JSON String

If present and "true", this capability indicates that the cloud storage system shall support assigned size for containers.

— `cdmi_geographic_placement`

JSON String

If present and "true", this capability indicates that the cloud storage system shall enforce the geographic restriction of object placement.

- 2 Section 16.4, Table 20 "`cdmi_geographic_placement`" refers to ISO-3166, which is not listed in Chapter 2 References

Marie to add reference.

- 3 Section 16.4, Table 20 "`cdmi_retention_id`" is defined as optional to enable retention, but Section 18.2 CDMI Retention implies that `cdmi_retention_id` is required

As per entry in table 116 Data System Metadata, `cdmi_retention_id` is "It is not required to place an object under retention".

No change to spec text needed.

- 4 Section 16.4, Table 20 "`cdmi_encryption`" description repeats the sentence "This metadata is the desired encryption support that the client is requesting of the cloud."

Already fixed in another ticket.

- 5 Section 16.4, Table 20 "`cdmi_encryption`" is missing a bullet point in front of "`AES_XTS_256`"

Already fixed.

- 6 Section 16.4, Table 20 "`cdmi_value_hash`" description is incorrect: the first sentence describes the capability, not the metadata; it states that the "`cdmi_security_data_integrity`" capability indicates support for this metadata, yet there is a `cdmi_value_hash` capability in Section 12.1.3 Table 12

Already fixed in another ticket.

- 7 Section 16.4, Table 20 the description of "`cdmi_value_hash`" should clarify the value format (the description states "Algorithm Length", but the example is "SHA256" (with no space between the algorithm and length)).

Already fixed in another ticket.

- 8 Section 16.4, Table 20 "`cdmi_throughput`" (and "`cdmi_throughput_provided`" in Table 21) has unclear units, as it states it is both bytes per second and Mbits/second.

In Table 116 - Data System Metadata, `cdmi_throughput` row, remove "(in Mbits/sec)" from the description.

- 9 Section 16.4, Table 20 "`cdmi_throughput`" states that it is the "desired maximum data rate" but as a user of a cloud system I'd rather specify my minimum desired data rate. This also contradicts "`cdmi_latency`" where one specifies the maximum

amount of time willing to wait until the first byte arrives. It might be difficult to guarantee a minimum throughput, but a maximum desired throughput is worthless when it is not achieved most of the time. The use case envisioned for a "maximum data rate" is not clear

This was debated many times in committee. This is the desired behaviour. No change to spec text needed.

16.6

- 1 Section 16.6, Table 21 is missing corresponding "provided" metadata fields for "cdmi_assignedsize_provided", "cdmi_retention_id_provided"

Both of these have no "_provided" metadata items. No change to spec text needed.

- 2 Section 16.6, Table 21 in the "cdmi_hold_id_provided" description "identifier" should be "identifiers"

In Table 117 - Provided Values of Data Systems Metadata Elements, "cdmi_hold_id_provided" row description, replace:

"Contains the user-specified hold identifier for active holds"

with

"Contains the user-specified hold identifiers for active holds"

CHANGES Made the specified changes in 1.0.1m.

Trac Ticket #453 Corrections to Chapter 17: Logging

Description: Minor corrections and suggestions for Chapter 17: Logging in revision 1.0.1h:

- 1 Section 17.5, Table 22: should describe how to specify *all* "cdmi_logging_classes". I'd expect an empty JSON array is to be used for this purpose, similar to cdmi_scope_specification.
- 2 Section 17.5 states that "each enqueued value consists of a JSON array of JSON objects, with each JSON object representing a single log message." -- is this correct? Why is each value in the queue an array of log message JSON objects, and what is the relationship between the set of log messages in a given value? It would seem more sensible for each value in the queue to be a JSON object specifying a *single* log message.
- 3 Section 17.5: in "each JSON object representing a single log message." -- We had this question before (on notification and query queues) about whether the value when reading the queue is actually a JSON object or a JSON-escaped JSON string (it is the latter for notification queues). This should be clarified here as well (preferably with an example).
- 4 Section 17.5, page 155: the reference to Table 21, "Logging Status Metadata" should actually be Table 23
- 5 Section 17.x: There should be a simple example to show creation of and retrieval from a Logging Queue.

Proposed changes against revision 1.0.1m:

- 1 Clause 20.3 - Page 211

Assuming the intent is for the contents of log messages to be free-form and implementation-specific (otherwise I would propose defining a set of standard fields), change "Log messages shall contain a minimum of the following information:" to: Log messages shall contain a minimum of the following information, in a format specified by the implementor:

TWG: Above change accepted.

2 Clause 20.6 - Page 212

In table 121, "cdmi_logging_class" row, below "cdmi_security_logging - Receive logging messages related to security events.", add the following sentence:

Clients may include the desired classes of log messages in the cdmi_logging_class JSON array. If all log messages are desired, an empty JSON array shall be used.

TWG: This would be added after the last bullet in the description for "cdmi_logging_class".

TWG: Above change accepted.

3 In table 121, "cdmi_scope_specification" row, change:

"The scope specification determines the set of objects that operations trigger the generation of log messages." to:

"The scope specification determines the set of objects for which associated log messages shall be enqueued."

TWG: Above change accepted.

4 Change the sentence

"When logging messages are enqueued into a logging queue, each enqueued value consists of a JSON array of JSON objects, with each JSON object representing a single log message."

to:

When logging messages are dequeued from a logging queue, the contents of each queue value shall contain a JSON object, and have a value mimetype of "application/json". This JSON object contains one or more JSON strings or objects, each representing a single log message.

TWG: Above change accepted.

5 Change the sentence beginning with:

"Log messages are only included in a logging queue if the user who created the logging queue is able to access the object associated with the log message, i.e., user has any ACE from 16.1.5.)"

to:

"Log messages are only included in a logging queue if the user who created the logging queue is able to access the object associated with the log message (i.e., user has any ACE from 16.1.5)."

TWG: Above change accepted.

6 In EXAMPLE 2, change the text of the example from:

"If the logging queue was created by the administrator, then all matching objects that the administrator is allowed to read are included in the results. If the notifica-

tion queue was created by user "jdoe", then only logging messages for objects that "jdoe" is allowed to read are included in the results."

with:

"If the logging queue was created by the administrator, then all matching objects without restriction are included in the results. If the logging queue was created by user "jdoe", then only logging messages for objects that "jdoe" is allowed to access are included in the results."

TWG: Above change accepted.

- 7 Below Table 122, delete the sentence "When logging results are stored in a logging queue, each enqueued value shall consist of a JSON object of MIME-type "application/json"."

TWG: Above change accepted.

CHANGES Made the specified changes to Version 1.0.1n EXCEPT for #5 - (cut & paste error) - waiting clarification.

Trac Ticket #454 Corrections to Chapter 18: Retention and Hold Management

Description: Minor corrections to Chapter 18: Retention and Hold Management in revision 1.0.1h:

Proposed Changes

Clause 17.1 - Page 199

Add a period to the end of the sentence:

However, Clause 17 describes what happens when placing either a retention policy and/or a hold on an object.

TWG: Change Approved.

Clause 17.3 - Page 199

- 1 Change "CDMI retention criteria shall be specified by:" to
CDMI retention criteria shall be specified by the following data system metadata:
TWG: Change Approved.
- 2 Change "cdmi_retention_duration" to "cdmi_retention_period"
TWG: Change Approved.
- 3 Above "Figure 10 shows", add the sentence:
When copying objects with a retention policy, retention properties shall not be transferred from the source CDMI object to the destination object, and the destination object shall not have a retention policy.
TWG: Change Approved.
- 4 Convert all dates in retention diagrams and associated text to ISO 8601 style YYYY-MM-DD format.
TWG: Change Approved.

- 5 In the text below Figure 10, change "Retention start date of 4/18/2010" to "Retention start date of 4/28/2010"

TWG: Change Approved.

- 6 Change "A cloud storage system is not responsible for enforcing value changes to the retention duration, as there are valid business reasons to change a retention duration for an object." to:

A cloud storage system shall not prevent metadata changes that increase the retention duration, as there are valid business reasons to change a retention duration for an object.

TWG: Change Approved.

Clause 17.4 - Page 200

- 1 Change "A CDMI client may place an object on hold by adding a hold id to the `cdmi_retention_hold` data system metadata item." to:

A CDMI client may place an object on hold by adding a hold identifier to the `cdmi_hold_id` data system metadata item.

TWG: Change Approved.

- 2 In Figure 12, add "Object deleted on 4/28/2014" to the right of the diagram.

TWG: Change Approved.

Clause 17.5 - Page 201

- 1 Change "(See `cdmi_retention_autodelete` in Table 118." to "(See `cdmi_retention_autodelete` in Table 118.)"

TWG: Change Approved.

- 2 Change "CDMI objects shall be automatically deleted by the system by setting the data system metadata flag, `cdmi_retention_autodelete`, at the retention duration expiration." to

CDMI objects shall be automatically deleted by the system at the retention duration expiration by setting the data system metadata flag `cdmi_retention_autodelete`.

TWG: Change Approved.

- 3 Search and replace all instances of "retention duration" with "retention period"

TWG: Change Approved.

Clause 17.6 - Page 202

- 1 Change

"Equally important is the relative accuracy and integrity of the clock, which determines if retention duration has elapsed, to the clock, which sets the start time property."

to:

"Equally important is the relative accuracy and security of the clock which determines if retention period has elapsed when compared to the clock which sets the start time property."

TWG: Change Approved.

2 Replace Clause 17.2 with:

"CDMI retention, deletion, and hold management affect any CDMI client that creates or deletes CDMI objects, as these disciplines mandate how a cloud storage system manages CDMI objects when they are created and until they are deleted.

CDMI retention management is comprised of three management disciplines: retention, hold, and deletion:

CDMI retention uses retention time criteria to determine the time period during which object deletion from the CDMI-based system is prohibited. No changes to the object are allowed, even after the retention period has expired, except as specified below.

CDMI holds prohibit object deletion and modification until all holds on the object have been released.

A CDMI-based system shall not allow the deletion of a CDMI object before the CDMI retention time criteria are met or while holds exist. Any deletion attempts (e.g., by a CDMI application) shall return an error.

After the CDMI retention time criteria have been met and all holds have been released, CDMI retention and holds shall no longer be a reason to prohibit object deletion.

Once the retention period has started or if holds exist, changes to the object data and metadata shall not be allowed, with the exception of extensions to the retention and hold data system metadata. The retention data system metadata may be added or the retention period extended, and the hold data system metadata may be added or extended with additional holds. Any other attempt to modify the object shall return an error."

TWG: Change Approved.

Clause 17.3 - Page 199

Change "Object metadata may be extended but not modified." to:

"The value of the object data system metadata for the retention period shall not be reduced."

TWG: Change Approved.

Clause 17.4 - Page 201

Change "Object metadata for the retention duration may be extended but not modified." to:

"The value of the object data system metadata for the retention period shall not be reduced, and the value of the object data system metadata for hold identifiers shall not permit holds to be removed. Removing holds is outside the scope of the CDMI international standard."

TWG: Change Approved.

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #455 Minor Changes to Chapter 19: Notification Queues**Proposed spec changes:**

- 1 Section 19, "by checking for the presence of the "cdmi_notifications" -- this is inconsistent with the definition of this capability in Section 12.1.1 as "cdmi_notification"
 TWG: Retain capability name as-is. Find locations in the spec where we refer to "cdmi_notifications" and replace it with "cdmi_notification"
- 2 Section 19, Table 24 under cdmi_notification_events, for "cdmi_create_complete" and "cdmi_modify_complete", are these notifications also generated when an object transitions from "Processing" to "Error"?
 TWG: Replace the below text under cdmi_notification_events:
 "cdmi_create_complete - Notifications are generated when a new object is created and is in the complete completion status. This notification is also generated when a new object being created transitions from "Processing" to "Complete"."
 with
 "cdmi_create_complete - Notifications are generated when a new object is created immediately or when a new object in the process of being created transitions from the "processing" completion status."
- 3 Section 19, Table 24 under cdmi_notification_events, is the "cdmi_copy" event generated for the source object, the destination object, or both? Is this generated at initiation of a copy or only when the copy is complete?
 TWG: Replace the below text under cdmi_notification_events:
 "cdmi_copy - Notifications are generated when an object is copied."
 with
 "cdmi_copy - Notifications are generated for the newly created copied object upon completion of the copy."
- 4 Section 19, Table 24 under cdmi_notification_events, "cdmi_reference" needs clarification -- is this notification only generated when a new reference is created (versus, e.g., accessed)?
 TWG: Replace the below text under cdmi_notification_events:
 "cdmi_reference - Notifications are generated when an object is referenced."
 with
 "cdmi_reference - Notifications are generated when a reference is created."
- 5 Section 19, Table 24, does not describe how to specify that *all* cdmi_notification_events should be returned. Previous discussion with dslik indicated that an empty JSON array is to be used for this purpose.
 TWG: Agreed to define an empty JSON array as meaning all notifications are to be returned.
 Add to end of the description in the cdmi_notification_events row, the following text:
 "Clients may include the desired notification event types in the cdmi_notification_events JSON array. If all notifications events are desired, an empty JSON array shall be used."

- 6 Section 19, Table 25, the purpose of the `cdmi_notification_status` metadata is unclear. The description states "the query is in progress or complete", which does not make sense, as there is no query associated with notification queues. There is also already a "completionStatus" field for queues generally which indicates the completion status of the creation of the queue.

Marie to ignore.

- 7 Section 19.x: There should be a simple example to show creation of a Notification Queue using the mechanism outlined in Section 11.2 (PUT) and Section 9.10 (POST), and an example of retrieving from a queue as outlined in Section 11.3 (GET) showing the values in the queue as JSON-escaped JSON strings.

Will be put into a separate ticket.

It appears that a number of the typos mentioned in this bug report were not actually fixed in revision 1.0.1i. This includes:

- 8 Section 19, Table 24 under `cdmi_scope_specification`, in "set of objects that on which", remove "that"
DS: Recommend that we make this change.
- 9 Section 19, Table 24 under `cdmi_results_specification`, the "[ISO-8601]" reference does not link to the reference.
DS: Recommend that we make this change.
- 10 Section 19, Table 24, links to Section 20.2 in the "Description" column for `cdmi_results_specification` should read "Section 20.2" instead of "Section ". The links actually point to the correct section, though.
DS: Recommend that we make this change
- 11 Section 19, in the metadata example, the "cdmi_notification_events" includes "cdmi_modify", but there is no such event (only "cdmi_modify_processing" and "cdmi_modify_complete")
DS: The example should be updated to refer to "cdmi_modify_complete".
- 12 Section 19, in the metadata example, the criteria for `cdmi_size` should presumably be "> +100000" rather than "> 100000" to perform a numeric comparison
DS: Recommend that we make this change.
- 13 Section 19, in the result example, the `cdmi_event_time` value is missing the UTC timezone indicator (trailing "Z") as required in Section 5.14, so the value should be "2010-11-15T13:12:52.342324Z"
DS: Recommend that we make this change.

CHANGES Made changes as specified in Items 1-5 above to Version 1.0.1i, and 8-13 in 1.0.1j.

Trac Ticket #456 Corrections to Chapter 20: Query Queues

Description: Minor corrections and suggestions for Chapter 20: Query Queues in revision 1.0.1h:

Proposed changes against revision 1.0.1m:**Clause 18**

- 1 Page 203, in the example CDMI GET response, add the following line after "HTTP/1.1 200 OK":
Content-Type: application/cdmi-object
- 2 Page 203, in the example CDMI GET response, change the value of the "objectType" field from "dataobject" to "application/cdmi-object".
- 3 Page 204, in "This approach allows the matching against arbitrarily nested metadata structures", change "allows the matching" to "allows matching"
- 4 Page 208, EXAMPLE 2, change "/myContainer/" to "/MyContainer/" for consistency with all other examples.
(Just make the first "m" a capital letter.)

Clause 19

- 1 Page 210, in "Result specifications are typically used to provide a CDMI client" change "Result specifications" to "Results specifications"
- 2 Page 210, EXAMPLE 1, in "The following results specification requests that the objectID and cdmi_size metadata fields to be returned in the results", change "to be returned" to "be returned"
- 3 Page 210, in "If a client wants to have all fields and metadata", the font-size of "If a client wants to have all" appears to be smaller than the rest of the sentence.
- 4 Page 210, replace:
"If a client wants to have all metadata fields returned for each matching object, the following cdmi_results_specification shall be used:"
with
"To request all metadata items be returned for each matching object, the following cdmi_results_specification shall be used:"
- 5 Page 210, replace:
"If a client wants to have all fields and metadata returned for each matching object, the following cdmi_results_specification shall be used:"
with
"To request all fields and all metadata items be returned for each matching object, the following cdmi_results_specification shall be used:"
- 6 Page 210, move the fragment "The value field is always returned in base 64 encoding when" below the example JSON to the beginning of the line "included in a query result"

Clause 22.1

- 1 Page 219, in "when the query is complete, the query_status metadata of the queue is changed", change "query_status" to "cdmi_query_status"
- 2 Page 219, Table 125, "cdmi_scope_specification" row, in the Description column, before "See Clause 18", add:
To query all objects, specify an empty JSON array.
- 3 Page 220, add a trailing comma to the line:

"cdmi_queue_type" : "cdmi_query_queue",

TWG: All changes approved.

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #457 Changes to Annex A: Transport Security

Description: Minor corrections and suggestions for Annex A: Transport Security in revision 1.0.1h:

- 1 Section A.1, paragraph 4 in "established good security practices avoids" -- "avoids" should be "avoid"
EAH>> Accepted. Change "avoids" to "avoid".
- 2 Section A.1 lists mandatory cipher suites which are repeated in A.3.1.1, but omits TLS_RSA_WITH_AES_128_CBC_SHA256 -- should this be included in A.1 as well? It's not clear if TLS_RSA_WITH_AES_128_CBC_SHA256 is mandatory for TLS 1.2 or just recommended.
EAH>> Rejected. TLS_RSA_WITH_AES_128_CBC_SHA256 is not mandatory; this is the result of downgrading to TLS 1.0.
- 3 Section A.2, paragraph 3 refers to RFC 2617 -- this should be added to Chapter 2, References
EAH>> Accepted. RFC 2617 should be added to Chapter 2, References.
- 4 Section A.3.1.1, the statement "To ensure a minimum level of security and interoperability between implementations" should be extended with ", all CDMI clients and servers shall support" which is misplaced on the first element of the following bullet list.
EAH>> Accepted. Change "To ensure a minimum level of security and interoperability between implementations:" to "To ensure a minimum level of security and interoperability between implementations, all CDMI clients and servers shall support:". In addition, change the first bullet from "All CDMI clients and servers shall support TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA" to "TLS_DHE_DSS_WITH_3DES_EDE_CBC_SHA".
- 5 Section A.3.1.1, TLS_RSA_WITH_NULL_SHA seems ill-advised in usage scenarios employing basic (plain-text) HTTP authentication and should be noted as such.
EAH>> [Important] Accepted. For the TLS_RSA_WITH_NULL_SHA bullet, add the following note: "Note: When this cipher suite is used, HTTP basic authentication shall not be used."
- 6 Section A.3.1.1, paragraph 5 states "TLS_RSA_WITH_AES_128_CBC_SHA...is the mandatory cipher suite for both TLS 1.1 and TLS 1.2" -- this is correct for TLS 1.2, but according to RFC 4346 the only mandatory cipher suite for TLS 1.1 is TLS_RSA_WITH_3DES_EDE_CBC_SHA ({0x000x0A})
EAH>> [Important] Accepted. When we downgraded to TLS 1.0 the cipher suites situation got a bit confused.

Change all occurrences that state "TLS_RSA_WITH_AES_128_CBC_SHA is the mandatory cipher suite for both TLS 1.1 and TLS 1.2" to "TLS_RSA_WITH_AES_128_CBC_SHA is the mandatory cipher suite for TLS 1.2". We can choose to ignore the mandatory cipher suite for TLS 1.1, which is TLS_RSA_WITH_3DES_EDE_CBC_SHA (hexadecimal value {0x000x0A}) or we can add it as another mandatory cipher suite. My personal recommendation is that we don't need yet another 3DES-based cipher suite that won't meet the NIST crypto guidance, so I would not include it as a mandatory cipher suite; however, we might want to include it as an identified optional cipher suite in A.3.1.1 like we did for TLS_RSA_WITH_AES_128_CBC_SHA256.

- 7** Section A.3.1.1, paragraph 7 states "guidance is provided in NIST Special Publication 800-57 and NIST Special Publication 800-131A" -- these publications should be added to Chapter 2, References.
- EAH>> Accepted in principle. NIST Special Publication 800-57 and NIST Special Publication 800-131A should be added to Chapter 2, References. This may create a problem with the CDMI spec in the ISO template.
- TWG: Remove reference to these NIST publications from the specification text.
- 8** Section A.3.1.1, paragraph 8 states that "Implementors are free to include additional cipher suites, but must prefer the mandatory ones in negotiation" -- this doesn't make much sense; given that the mandatory cipher suites must be supported by both client and server, if they must always be preferred it will never be possible for a server to select a different cipher (e.g., a stronger cipher that the client may indicate it supports and should prefer). It would make more sense to state that implementors are free to include additional cipher suites, but must always include the mandatory ones.
- EAH>> Accepted in principle. The issue is that an implementation could include weak cipher suites and then choose one of them over the stronger mandatory cipher suites; however we don't want to preclude the use of stronger cipher suites. Change "Implementors are free to include additional cipher suites, but must prefer the mandatory ones in negotiation" to "Implementors are free to include additional cipher suites, but must prefer the mandatory ones over weaker cipher suites in negotiations."
- TWG: Change "Implementors are free to include additional cipher suites, but must prefer the mandatory ones in negotiation" to "Implementors are free to include additional cipher suites."
- 9** Section A.3.1.2.2 redundantly includes the full bibliographic reference to ITU-T509 and RFC2045, duplicating the information linked to in Chapter 2, References.
- EAH>> Accepted. Full bibliographic reference to ITU-T509 and RFC2045 will be removed from the bullets in Section A.3.1.2.2.
- 10** Section A.3.1.2.2, paragraph 7, in "list per CA root certificate supported" -- remove "supported"
- EAH>> Rejected. Change "list per CA root certificate supported" to "list per CA root certificate shall also be supported"
- 11** Section A.3.1.2.3, the statement "The above requirements may be satisfied via appropriate use of the readily-available OpenSSL toolkit software (www.openssl.org)" sounds like endorsement of a particular toolkit. Could be rephrased to "...appropriate use of readily-available cipher-suites such as or

[OpenSSL] or [GnuTLS]", with the referred toolkits added to Section 2.2 Informative References.

EAH>> Accepted. Change "The above requirements may be satisfied via appropriate use of the readily-available OpenSSL toolkit software (www.openssl.org)." to "The above requirements may be satisfied via appropriate use of readily-available software such as [OpenSSL] or [GnuTLS]." Also not that the referred toolkits need to be added to Section 2.2 Informative References.

TWG: Remove entire line "The above requirements may be satisfied via appropriate use of the readily-available OpenSSL toolkit software (www.openssl.org)."

CHANGES Made the changes specified in Items 1-11 above.

Trac Ticket #458 Corrections to Annex B: Extending the Interface

Description: Minor corrections and suggestions for Annex B: Extending the Interface in revision 1.0.1h:

- 1 Introduction states that "a vendor extension should be written so that the implementor name may be replaced with "cdmi_" when added to the standard", but the following examples use a domain convention syntax for the vendor name, e.g. "netapp.com.<capability>", which is in accordance with 12.1 but the vendor name can't be directly replaced with "cdmi_"
 DS: Replace
 "a vendor extension should be written so that the implementor name may be replaced with "cdmi_" when added to the standard"
 with
 "a vendor extension shall be written so that the implementor prefix can be replaced with "cdmi_" when added to the standard"
- 2 Section B.3, in description of "com.netapp.versioning", "If present and true" should be "If present and "true"" for consistency with Chapter 12.
 DS: This change should be incorporated into the spec.
- 3 Section B.3, clarify if the com.netapp.versions capability specifies only the number of *previous* versions, as per the DSMD defined in B.4 (versus the *total* number of versions, including the current version).
 DS: We should clarify.
 In the spec, in second table in B.3, replace the description:
 "If present, this capability defines the maximum number of versions that may be requested for a given data object."
 with:
 "If present, this capability defines the maximum number of previous versions that may be requested for a given data object."
- 4 Section B.4, typo in title "Versioning Data System Metadaw" ("Metadata")
 DS: This change should be incorporated into the spec.
- 5 Section B.6, "Version objects created by the system are immutable" -- should state that version objects can still be deleted (as per B.6)

DS: This change should be incorporated into the spec.

Replace:

"Version objects created by the system are immutable, and versions shall not have nested versions."

with

"Version objects created by the system are immutable, but may be deleted by a client. Versions shall not have nested versions."

- 6** Section B.6, Figure 15 uses the field name "versionsURI", but the text uses the field name "versions". Which one is correct?

DS: The text "versionURI" should be removed from the diagram. That is a remnant from when we were looking at adding a new field to indicate how to access the versions.

- 7** Section B.6.1, "X-CDMI-PARTIAL" should be "X-CDMI-Partial", as used in the operations descriptions elsewhere in the document

DS: This change should be incorporated into the spec.

- 8** Section B.6.1, the examples in this section need some whitespace between the OID and the value (e.g., in "...DC1289"This")

DS: This change should be incorporated into the spec. Another ticket also covers this issue.

- 9** Section B.6.1, should clarify the behaviour of object IDs and versions, as this is not obvious, e.g. by stating that the OID of the versioned object does not change, but each version version that is created is assigned a new OID

DS: This change should be incorporated into the spec.

Need to write spec text.

- 10** Section B.6.3, the statement "If a limited number of versions are requested in the "com.netapp.versions" data system metadata" implies that there is a way for a CDMI system to indicate that it supports an unlimited number of versions and a way for a client to request this, but how to do so (and if this is even supported) is not described in the earlier B.3 and B.4 sections (short of putting a very large number for "com.netapp.versions"). Similarly, Section B.6 (page 182) also describes a "limited" number of versions with no indication of how this can be specified.

DS: There are a couple of ways this could be accomplished, either with a very large number, or an empty string "".

Need to discuss in the TWG.

- 11** Section B.6.4, "Versions are serialized as children of the data object" -- the serialization format needs to be defined, e.g. do versions appear under a "children" field or under a "versions" field of the data object)

DS: This change should be incorporated into the spec.

Need to write spec text.

- 12** Section B.6.6, "as the version is immutable and may not be deleted" -- B.6 states that versions can be deleted

DS: This change should be incorporated into the spec.

Replace:

"A move operation for a versioned object shall move the object and all versions, unless the move is to a system that does not support versions, in which case it

shall only move the data object. A move operation for a version shall fail, as the version is immutable and may not be deleted."

with

"A move operation for a versioned object shall move the object and all versions, unless the move is to a system that does not support versions, in which case it shall only move the data object. A move operation for a version will result in the deletion of the version being moved."

TWG reviewed and approved all changes except for:

9. Section B.6.1, should clarify the behaviour of object IDs and versions, as this is not obvious, e.g. by stating that the OID of the versioned object does not change, but each version version that is created is assigned a new OID
10. Section B.6.3, the statement "If a limited number of versions are requested in the "com.netapp.versions" data system metadata" implies that there is a way for a CDMI system to indicate that it supports an unlimited number of versions and a way for a client to request this, but how to do so (and if this is even supported) is not described in the earlier B.3 and B.4 sections (short of putting a very large number for "com.netapp.versions"). Similarly, Section B.6 (page 182) also describes a "limited" number of versions with no indication of how this can be specified.

TWG: Additional options: "-1", "unlimited"

This also applies to other capabilities, such as Max Metadata Items.

TWG decision is -1.

Spec text needs to be written.

11. Section B.6.4, "Versions are serialized as children of the data object" -- the serialization format needs to be defined, e.g., do versions appear under a "children" field or under a "versions" field of the data object?

TWG: Don't overload children field. Perhaps create new "versionsrange", "versions".

This has an interesting property that if a cloud that does not support versions deserialized as an object that has versions, then serialized and deserialized it on a third cloud that does support versions, the versions would be preserved.

This also affects the last change, as when moving an object to a system that does not support versions, these fields shall be preserved.

Update "A move operation for a versioned object shall move the object and all versions, unless the move is to a system that does not support versions, in which case it shall only move the data object. A move operation for a version will result in the deletion of the version being moved." to reflect this.

CHANGES Made the specified changes.

Trac Ticket #459 Clarify that `cdmi_size` and `cdmi_hash` cannot be modified

Description: We should add text that indicates that a client cannot set the `cdmi_size` and `cdmi_hash` storage system metadata.

Proposed spec change:

- Add to the `cdmi_size` and `cdmi_hash` descriptions in table 19 on page 146:
This storage system metadata item is computed by the storage system, and any attempts to set or modify will be ignored.
- Refined text to be added to the two Storage System Metadata descriptions:
This storage system metadata item is computed by the storage system, and any attempts to set or modify it will be ignored.

Discussed on the 2011-04-27 TWG call and approved for incorporation into the next draft of the CDMI specification.

CHANGES Made the specified changes.

Trac Ticket #460 Extra incorrect Location headers in examples in section 12

Description: We should remove the incorrect "Location:" headers from the request and response headers in examples 12-1, 12-2 and 12-3

Discussed on the 2011-04-27 TWG call and approved for incorporation into the next draft of the CDMI specification.

CHANGES Made the specified changes.

Trac Ticket #463 Add capabilities for moving objects

Description: We should consider adding to Table 10 system wide capabilities to indicate if a CDMI system supports moving objects, and to what extent.

Proposed spec text:

- `cdmi_object_move_from_local`
JSON String
If present and "true", indicates that the cloud storage system supports moving CDMI objects from URIs within the same storage system.
- `cdmi_object_move_from_remote`
JSON String
If present and "true", indicates that the cloud storage system supports moving CDMI objects from URIs within other CDMI storage systems.
- `cdmi_object_move_from_ID`

JSON String

If present and "true", indicates that the cloud storage system supports moving CDMI objects without a path from a /cdmi_objectID/ URI within the same storage system. This effectively adds a path, allowing the object to be accessed by ID and by path.

- cdmi_object_move_to_ID

JSON String

If present and "true", indicates that the cloud storage system supports moving CDMI objects with a path to a /cdmi_objectID/ URI within the same storage system. This effectively removes the path, leaving the object only accessible by ID.

Reviewed on 2011-05-20 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #464 Add capabilities for copying objects

Description: We should consider adding to Table 10 system wide capabilities to indicate if a CDMI system supports moving copying, and to what extent.

Proposed spec text:

- cdmi_object_copy_from_local

JSON String

If present and "true", indicates that the cloud storage system supports copying CDMI objects from URIs within the same storage system.

- cdmi_object_copy_from_remote

JSON String

If present and "true", indicates that the cloud storage system supports copying CDMI objects from URIs within other CDMI storage systems.

Reviewed on 2011-05-20 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #470 CDMI Object ID in Location header truncated in example 9-12

Description

- 1 The Location header in the response header for example 9-12 should read:

```
Location: http://cloud.example.com/cdmi_objectid/
0000706D0010B84FAD185C425D8B537E
```

(Last character "E" was dropped)

2 Same error in example 9-14. It should read:

Location: `http://cloud.example.com/cdmi_objectid/
0000706D0010B84FAD185C425D8B537E`

Discussed on the 2011-05-11 TWG call, and approved for addition to the spec.

CHANGES Made the specified changes.

Trac Ticket #471 Replace example COIDs with COIDs that use the example enterprise number.

Description: The current example COIDs should be replaced COIDs generated using the official example enterprise number:

32473

Example Enterprise Number for Documentation Use

See [RFC5612]

`iana&iana.org`

Here are some example COIDs that we can use.

Search and replace:

`0000706D00100C435125A61B4C289455 ->
00007E7F0010CEC234AD9E3EBFE9531D`

`0000706D0010D538DEEE8E38399E2815 ->
00007E7F00102E230ED82694DAA975D2`

`0000706D0010734CE0BAEB29DD542B51 ->
00007E7F0010BD1CB8FF1823CF05BEE4`

`0000706D0010B84FAD185C425D8B537E ->
00007E7F0010EB9092B29F6CD6AD6824`

`0000706D00107B85BFE6D20B84D603CA ->
00007E7F00104EB781F900791C70106C`

`0000706D00101ADEBC119D1BFE98672A ->
00007E7F00104BE66AB53A9572F9F51E`

`0000706D0010171EADF15DE7BC0917D3 ->
00007E7F001074C86AD256DA5C67180D`

`0000706D0010E0981215538EE7D19E5E ->
00007E7F001072D79B8D19A6F20F192B`

0000706D0010201DA76AAE16879EB847 ->
00007E7F001054C4DB6185C28C3ACDB3

0000706D001011CCC130940D93F8D97B ->
00007E7F001095ADCED1E7A67E3DB5AA

0000706D00104E935A9B4F95242538FE ->
00007E7F0010A4D51D96B6263E6FEDEC

0000706D001006A1D4534CF0DFDC1289 ->
00007E7F00107E41B2AEBE9F0DF8E1CB

0000706D001089F999BB0B087D30B340 ->
00007E7F0010051EEC0B5DBFE3700E40

0000706D0010CCB1FCDD87D9CA762065 ->
00007E7F0010117E54903F65F9AB846C

We also should update the two COID fragments in Figure 15:

0000706D0010201D... -> 00007E7F0010D71D...

0000706D001011CC... -> 00007E7F0010A4B6...

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES: Made the specified changes; created new graphic to replace existing (no original).

Trac Ticket #472 **OIDs don't match with example flow**

Description: Many of the examples have OIDs that either don't match with the transactions, or use the same OID to refer to different things.

We need to review through all of the examples and make sure that the OIDs used are consistent with the spec.

Proposed spec text to address OIDs, missing "/" for containers and domains (Ticket #630), and missing parentID fields (Ticket #631).

Clause 6

- 1 Clause 6.2, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010DCECC805FB6D195DDBCB",
- 2 Clause 6.3, Example, request body:
Missing ending "/" on request URI.
Replace:
"PUT /MyContainer HTTP/1.1"
with

"PUT /MyContainer/ HTTP/1.1"

- 3 Clause 6.3, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010128E42D87EE34F5A6560",
- 4 Clause 6.4, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",
- 5 Clause 6.5, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010128E42D87EE34F5A6560",
- 6 Clause 6.6, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",

Clause 8

- 1 Clause 8.2.9, Example 1, response body,
Add line below parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",
- 2 Clause 8.2.9, Example 1, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/dataobject",
with
"capabilitiesURI" : "/cdmi_capabilities/dataobject/",
- 3 Clause 8.2.9, Example 2, response body,
Add line below parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",
- 4 Clause 8.2.9, Example 2, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/dataobject",
with
"capabilitiesURI" : "/cdmi_capabilities/dataobject/",
- 5 Clause 8.4.8, Example 1, response body,
Add line below parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",
- 6 Clause 8.4.8, Example 1, response body,
Replace:
"domainURI" : "/cdmi_domains/MyDomain",
with
"domainURI" : "/cdmi_domains/MyDomain/",

- 7 Clause 8.4.8, Example 1, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/dataobject",
with
"capabilitiesURI" : "/cdmi_capabilities/dataobject/",
- 8 Clause 8.4.8, Example 2, response body,
Add line below parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",

Clause 9

- 1 Clause 9.1, Example 2:
Replace:
http://cloud.example.com/container?children
with
http://cloud.example.com/container/?children
- 2 Clause 9.1, Example 3:
Replace:
http://cloud.example.com/container?children:0-2
with
http://cloud.example.com/container/?children:0-2
- 3 Clause 9.1, Example 4:
Replace:
http://cloud.example.com/container?children;metadata
with
http://cloud.example.com/container/?children;metadata
- 4 Clause 9.2.9, Example, request:
Replace:
PUT /MyContainer HTTP/1.1
with
PUT /MyContainer/ HTTP/1.1
- 5 Clause 9.2.9, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010128E42D87EE34F5A6560",
- 6 Clause 9.2.9, Example, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/container",
with
"capabilitiesURI" : "/cdmi_capabilities/container/",
- 7 Clause 9.4.8, Example 1, request,
Replace:

GET /MyContainer HTTP/1.1
with
GET /MyContainer/ HTTP/1.1

- 8 Clause 9.4.8, Example 1, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010128E42D87EE34F5A6560",
- 9 Clause 9.4.8, Example 1, response body,
Replace:
"domainURI" : "/cdmi_domains/MyDomain",
with
"domainURI" : "/cdmi_domains/MyDomain/",
- 10 Clause 9.4.8, Example 1, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/container",
with
"capabilitiesURI" : "/cdmi_capabilities/container/",
- 11 Clause 9.9.9, Example 1, response body,
Add line below parentURI line:
"parentID" : "0000706D0010B84FAD185C425D8B537E",
- 12 Clause 9.9.9, Example 1, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/dataobject",
with
"capabilitiesURI" : "/cdmi_capabilities/dataobject/",
- 13 Clause 9.11.9, Example, response body,
Add line below parentURI line:
"parentID" : "0000706D0010B84FAD185C425D8B537E",
- 14 Clause 9.11.9, Example, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/queue",
with
"capabilitiesURI" : "/cdmi_capabilities/queue/",

Clause 10

- 1 Clause 10.2.8, Example, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010C058374D08B0AC7B3550",
- 2 Clause 10.3.8, Example 1, response body,
Add line below parentURI line:

"parentID" : "00007E7F0010C058374D08B0AC7B3550",

Clause 11

- 1 Clause 11.2.9, Example, response body,
Add line below parentURI line:
"parentID" : "0000706D0010B84FAD185C425D8B537E",
- 2 Clause 11.2.9, Example, response body,
Replace:
"capabilitiesURI" : "/cdmi_capabilities/queue",
with
"capabilitiesURI" : "/cdmi_capabilities/queue/",
- 3 Clause 11.3.8, Example 1, response body,
Add line below parentURI line:
"parentID" : "0000706D0010B84FAD185C425D8B537E",

Clause 12

Clause 12.2.8, Example 1, response body,
Add line below parentURI line:
"parentID" : "00007E7F0010128E42D87EE34F5A6560",

Clause 15

- 1 Clause 15.3.2, Example,
Replace the first objectid line:
"objectID" : "00007E7F00104BE66AB53A9572F9F51E",
with
"objectID" : "00007E7F00102E230ED82694DAA975D2",
- 2 Clause 15.3.2, Example,
Add line below the first parentURI line:
"parentID" : "00007E7F0010128E42D87EE34F5A6560",
Add line below the second parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",
Add line below the third parentURI line:
"parentID" : "00007E7F00102E230ED82694DAA975D2",
- 3 Clause 15.3.2, Example,
Replace the second objectid line:
"objectID" : "00007E7F00104BE66AB53A9572F9F51E",
with
"objectID" : "0000706D0010B84FAD185C425D8B537E",

Clause 18

- 1 Clause 18, first GET results,

Add line below parentURI line:

```
"parentID" : "00007E7F00102E230ED82694DAA975D2",
```

2 Clause 18, Example 2,

Replace the line:

```
"parentURI" : "==" /cdmi_objectid/00007E7F001072D79B8D19A6F20F192B/"
```

with

```
"parentURI" : "==" /cdmi_objectid/0000706D0010B84FAD185C425D8B537E/"
```

Clause 19

Clause 19, first GET results,

Add line below parentURI line:

```
"parentID" : "00007E7F00102E230ED82694DAA975D2",
```

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #473 Tab Indents incorrect in B.6.1 The Current Version

Description: In section B.6.1 The Current Version, in the grey example text, the indents are incorrect, such that the example contents is abutting up against the COIDs in the URIs.

Instead of:

```
/document.txt/versions/0000706D001006A1D4534CF0DFDC1289 "This"
```

It should read as

```
/document.txt/versions/0000706D001006A1D4534CF0DFDC1289 "This"
```

(where it is a tab that separates the COID at the end of the URI from the example value. HTML will replace it with a space.

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #474 Remove pseudo-inheritance references

Description: We eliminated the concept of domains, queues, etc, being inherited from containers back in 0.8 of the spec, so we should remove the statements that say that they are "special classes of containers".

Spec changes:

1 In section 10.1 Overview, replace:

"Domains are a special class of container that are created in the `cdmi_domains` container found in the root URI for the cloud storage system."

with

"Domains are created in the `cdmi_domains` container found in the root URI for the cloud storage system."

2 In section 11.1 Overview, replace:

"Queues are a special class of container object and are used to provide first-in, first-out access when storing and retrieving data."

with

"Queues are a special type of object used to provide first-in, first-out access when storing and retrieving data."

3 In section 12.1 Overview, replace:

"Capability objects are a special class of container object that allow a CDMI client to discover what subset of the CDMI standard is implemented by a CDMI provider."

with

"Capability objects allow a CDMI client to discover what subset of the CDMI standard is implemented by a CDMI provider."

4 In section 11.1 Overview, replace:

"Queues are a special class of container object and are used to provide first-in, first-out access when storing and retrieving data."

with

"Queues provide first-in, first-out access when storing and retrieving data."

Trac Ticket #475 Add capability for the total amount of user metadata supported per object

Description: CDMI 1.0 currently defines the following capabilities for user metadata:

- `cdmi_metadata_maxitems`: The maximum number of user-defined metadata items per object.
- `cdmi_metadata_maxsize`: The maximum size of each user-defined metadata item.

It does not logically follow that a CDMI server will necessarily be able to support `cdmi_metadata_maxitems * cdmi_metadata_maxsize` total metadata per object. For example, we are targeting our system to support 1024 metadata items per object and up to 1MiB per item, but we cannot practically support 1GiB of total metadata per object.

There seems to be a need for a third metadata capability to be added that specifies the maximum total size of user metadata that is supported for each object.

We should discuss this proposal. If accepted, I would propose the following spec text change:

Add to Table 10:

cdmi_metadata_maxtotalsize, JSON String, If present, this capability specifies the maximum size in bytes of user-defined metadata supported by the cloud storage system. If absent, there is no limit placed on the size of user-defined metadata.

TWG approved the following text: Updating to reflect other changes made to capabilities:

Add to Table 102:

cdmi_metadata_maxtotalsize, JSON String, If present, this capability specifies the maximum total size in bytes of all user-defined metadata items per object. If absent, there is no limit placed on the total size of user-defined metadata items.

CHANGES Made the specified changes to Version 1.0.11.

Trac Ticket #476 children/childrenrange should not be returned when creating a container

Description: To act consistently when compared to creating a data object (where the value and valuerange is not returned), when creating a new container, the children and childrenrange fields should not be returned in the response body.

This also solves the problem that when deserializing a large container, you don't want the listing of all the children returned in response to the container create.

Spec change:

In section 9.2, remove the "childrenrange" and "children" rows from the table under Response Message Body.

In example 9-1, in the response, remove the "childrenrange" : "", "children" : [] lines (and don't forget to remove the last "," after the exports object to make it valid JSON).

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #477 Add Object ID example for references

Description: The spec currently provides an example of accessing a reference via path name, but not one for accessing a reference by object ID.

Proposed spec text change:

Add to end of section 7.3:

References by Object ID must always point to a URI that ends with the same Object ID as the request URI.

Example 7-2

```
GET to an Object ID URI, where the URI is a reference:
GET /cdmi_objectid/00006FFD0010AA33D8CEF9711E0835CA HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
X-CDMI-Specification-Version: 1.0
```

The response looks like:

```
HTTP/1.1 302 Found Location: http://archive.example.com/cdmi_objectid/
00006FFD0010AA33D8CEF9711E0835CA
```

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #478 Undefined "Information Services (future) in Section 5.6

Description: Section 5.6, Figure 5, there are boxes for 'Information Services (future)' but no mention of what this actually means in the text.

(Separated out from ticket #432)

We should consider removing this from Figure 5, or adding text describing what this means.

TWG: Approved below addition to end of clause 5.4.

"The SIRDM defines Information Services as services that understand the context of the data. Information Services are thus able to determine the requirements of the data and automatically mark the Data System metadata for that data."

Note for Marie:

SIRDM should be a reference to SIRDM in the bibliography.

CHANGES Made the specified changes to version 1.0.1m.

Trac Ticket #479 JSON encoding of binary data in object value is not well-defined

Marie — The TWG reviewed and approved the following text for Trac ticket #479.

Changes are highlighted in red.

Clause 5

Add section 5.15 Backwards Compatibility:

5.1.15 Backwards Compatibility**5.1.15.1 Value Transfer Encoding**

CDMI version 1.0.1 introduces the concept of value transfer encoding to enable the storage and retrieval of arbitrary binary data via CDMI operations. Data objects created by CDMI 1.0 clients through CDMI operations shall have a value transfer encoding of "utf-8", and data objects created through non-CDMI operations shall have a value transfer encoding of "base64".

Data objects with a value transfer encoding of base 64 shall not have their value field accessible to CDMI 1.0 clients through CDMI operations, and attempts to read the value of these objects shall return an empty value field ("") to these clients. CDMI 1.0 clients can detect this condition when the `cdmi_size` metadata is not 0, and the value field is empty.

Clause 6

In subclause 6.6, in the response:

The following shows the response.

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.1

{
  "objectURI": "/MyContainer/MyDataObject.txt",
  "objectID": "00007E7F0010BD1CB8FF1823CF05BEE4",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "17"
  },
  "valuetransferencoding": "utf-8",
  "valuerange": "0-16",
  "value": "Hello CDMI World!"
}
```

Subclause 8.1

1 In section 8.1:

Individual fields within a data object may be accessed by specifying the field name after a question mark "?" that is appended to the end of the data object URI. Thus, the following URI returns the value field in the response body:

```
http://cloud.example.com/dataobject?value
```

The encoding of the data transported in the data object value field is dependent on the data object "valuetransferencoding" field:

- If the value transfer encoding of the object is set to "utf-8", the data stored in the value of the data object shall be a valid UTF-8 string, and shall be transported as a UTF-8 string in the "value" field.
- If the value transfer encoding of the object is set to "base64", the data stored in the value of the data object can contain arbitrary binary sequences, and shall be transported as a base 64 encoded string in the "value" field.

Specific ranges of the value of a data object may be accessed by specifying a byte range after the value field name. Thus, the following URI returns the first thousand bytes in the value field:

`http://cloud.example.com/dataobject?value:0-999`

Because a byte range of a UTF-8 string is often not a valid UTF-8 string, the response to a range request shall always be transported in the "value" field as a base 64 encoded string. Likewise, when updating a range of bytes within the value of a data object, the contents of the "value" field shall be transported as a base 64 encoded string.

Byte ranges are specified as single inclusive byte ranges as per Section 14.35.1 of RFC 2616.

2 In section 8.1.1:

Data object metadata may also include arbitrary user-supplied metadata and data system metadata, as specified in Clause 16 "Metadata". Metadata shall be stored as a valid UTF-8 string. Binary data stored in user metadata shall be first encoded such that it can be contained in a UTF-8 string, with the use of base 64 encoding recommended.

3 In section 8.1.3:

The representations in this subclause are shown using JSON notation. Both clients and servers shall support UTF-8 JSON representation. The request and response body JSON fields may be specified or returned in any order, with the exception that, if present, the `valuerange` and `value` fields shall appear last and in that order.

Subclause 8.2

1 In section 8.2, Request Message Body, "mimetype" row:

MIME type of the data contained within the value field of the data object.

- This field may be included when creating by value, deserializing, serializing, copying, and moving a data object.
- This field shall be stored as part of the object.
- If this field is not specified, the value of "text/plain" shall be assigned as the field value.
- This field shall not be included when creating a reference.

2 In section 8.2, Request Message Body, "deserializevalue" row:

A data object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648.

- 3 In section 8.2, Request Message Body, above the "value" row, add:
- valuetransferencoding**
JSON String
The value transfer encoding used for the data object value. Two 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.
 - "base64" indicates that the data object may contain arbitrary binary sequences, and 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.
 - This field shall only be included when creating a data object by value,
 - If not specified by the client, the value transfer encoding shall be set to "utf-8" by the server, and
 - This field shall be stored as part of the object.
- Optional**
- 4 In section 8.2, Request Message Body, "value" row:
- The data object value.**
- If this field is not included, an empty JSON String ("") shall be assigned as the field value,
 - If the value transfer encoding field indicates UTF-8 encoding, the value shall be a UTF-8 string escaped using the JSON escaping rules described in RFC 4627, and
 - If the value transfer encoding field indicates base 64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648.
- 5 In section 8.2, Request Message Body, last footnote:
- ^aOnly one of these fields shall be specified in any given operation. Except for value, these fields shall not be stored. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response.
- 6 In section 8.2.9, Example 1:
- EXAMPLE 1 PUT to the container URI the data object name and UTF-8 contents:**
- ```
PUT /MyContainer/MyDataObject.txt HTTP/1.1
Host: cloud.example.com
Accept: application/cdm-object
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.1

{
 "mimetype": "text/plain",
 "metadata": {},
 "value": "This is the Value of this Data Object"
}
```
- The following shows the response.
- ```
HTTP/1.1 201 Created
```

```

Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.1

{
  "objectURI": "/MyContainer/MyDataObject.txt",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37"
  }
}

```

7 New example 8-2:

EXAMPLE 2 PUT to the container URI the data object name and binary 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: 1.0.1

```

```

{
  "mimetype" : "text/plain",
  "metadata" : { },
  "valuetransferencoding" : "base64"
  "value" : "VGhpcyBpcyB0aGUgVmFsdWUgb2YgdGhpcyBEYXRhIE9iamVjdA=="
}

```

The following shows the response.

```

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

```

```

{
  "objectURI": "/MyContainer/MyDataObject.txt",
  "objectID": "0000706D0010374085EF1A5C7018D774",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37"
  }
}

```

Subclause 8.3

- 1 In section 8.3, Request Headers, "Content-Type" row:

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". Otherwise, the "valuetransferencoding" field of the CDMI data object shall be set to "base64".

2 In Section 8.3.8, Example:

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

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

This is the Value of this Data Object

Subclause 8.4

1 In section 8.4, Synopsis:

— <range> is a byte range of the data object value to be returned in the value field.

2 In section 8.4, Response Message Body, "valuerange" row:

The range of bytes of the data object value to be returned in the value field.

- If a specific value range has been requested, the value range field shall correspond to the bytes requested. If the request extends beyond the end of the value, the value range field shall indicate the smaller byte range returned,
- If the object value has gaps (due to PUTs with non-contiguous value ranges), the value range will indicate the range to the first gap in the object value, and
- The cdmi_size storage system metadata of the data object shall always indicate the complete size of the object, including zero-filled gaps.

3 In section 8.4, Response Message Body, above the "value" row:

valuetransferencoding

JSON String

The value transfer encoding used for the data object value. Two 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.
- "base64" indicates that the data object may contain arbitrary binary sequences, and shall be transported as a base 64 encoded string in the "value" field.

Mandatory

4 In section 8.4, Response Message Body, "value" row:

The data object value.

- If the value transfer encoding field indicates UTF-8 encoding, the value field shall contain a UTF-8 string using JSON escaping rules described in RFC 4627,

- If the value transfer encoding field indicates base 64 encoding, the value field shall contain a base 64 encoded string as described in RFC 4648,
- The value field shall only be provided when the "completionStatus" field contains "Complete", and
- When reading a value, zeros shall be returned for any gaps resulting from non-contiguous writes.

5 In Section 8.4.8, Example 1:

The following shows the response.

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0.1
Content-Type: application/cdm-object

{
  "objectURI": "/MyContainer/MyDataObject.txt",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "domainURI": "/cdmi_domains/MyDomain",
  "capabilitiesURI": "/cdmi_capabilities/dataobject",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37"
  },
  "valuerange": "0-36",
  "valuetransferencoding": "utf-8",
  "value": "This is the Value of this Data Object"
}
```

6 In Section 8.4.8, Example 2:

The following shows the response.

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0.1
Content-Type: application/cdm-object

{
  "objectURI": "/cdmi_objectid/0000706D0010B84FAD185C425D8B537E",
  "objectID": "0000706D0010B84FAD185C425D8B537E",
  "objectName": "MyDataObject.txt",
  "parentURI": "/MyContainer/",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/dataobject/",
  "completionStatus": "Complete",
  "mimetype": "text/plain",
  "metadata": {
    "cdmi_size": "37"
  },
  "valuerange": "0-36",
  "valuetransferencoding": "utf-8",
  "value": "This is the Value of this Data Object"
}
```

7 In Section 8.4.8, Example 4:

The following shows the response.

```
HTTP/1.1 200 OK
X-CDMI-Specification-Version: 1.0.1
Content-Type: application/cdm-object
```

```
{
  "valuerange" : "0-10",
  "value" : "VGhpcyBpcyB0aGU="
}
```

Subclause 8.6

1 In Section 8.6, Synopsis:

<range> is a byte range within the data object value to be updated.

2 In section 8.6, Request Message Body, "mimetype" row:

MIME type of the data contained within the value field of the data object. If present, this replaces the existing mimetype.

- This field may be included when updating by value, deserializing, and copying a data object.
- This field shall be stored as part of the object.
- If this field is not specified, the existing value of the mimetype shall be left unchanged.
- This field shall not be included when creating a reference.

3 In section 8.6, Request Message Body, "deserializevalue" row:

A data object serialized as specified in Clause 15 "Serialization/Deserialization" 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.

4 In section 8.6, Request Message Body, above the "value" row, add:

valuetransferencoding

JSON String

The value transfer encoding used for the data object value. Two 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.
- "base64" indicates that the data object may contain arbitrary binary sequences, and 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.
- This field shall only be included when updating a data object by value.
- If this field is not specified, the existing value of "valuetransferencoding" shall be left unchanged.
- This field shall be stored as part of the object.

Optional

- 5 In section 8.6, Request Message Body, "value" row:
This is the new data for the object. If present, this replaces the existing value.
- If the value transfer encoding 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 value transfer encoding field indicates base 64 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 is 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 value transfer encoding field shall be set to "base64".**

- 6 In section 8.6, Request Message Body, last footnote:

^aOnly one of these fields shall be specified in any given operation. Except for value, these fields **shall not be stored**. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response.

- 7 In section 8.6.8, Example 3:

EXAMPLE 3 PUT to the data object URI to update a range of the value

```
PUT /MyContainer/MyDataObject.txt?value:21-24 HTTP/1.1
Host: cloud.example.com
Content-Type: application/cdmis-object
X-CDMI-Specification-Version: 1.0.1
```

```
{
  "value" : "dGhhdA=="
}
```

- 8 In section 8.6.8, Example 3, Add sentence below response:

When updating a value without specifying a value transfer encoding, the client must be aware of the current value transfer encoding of the object. If a client sends a value containing a UTF-8 string to update an existing object with a "valuetransferencoding" of "base64", this shall result in an error being returned. If a client sends a value containing a base 64 string to update an existing object with a "valuetransferencoding" of "utf-8", this shall not generate an error, but results in the literal base 64 character sequence being stored in the data object instead of the expected data encoded in the base 64 string.

Clause 9

- 1 In section 9.1.3:

The representations in this subclause are shown using JSON notation. Both clients and servers shall support UTF-8 JSON representation. The request and response body JSON fields may be specified or returned in any order, with the exception that, if present, **for container objects, the childrenrange and children fields shall appear last and in that order, and for data objects, the valuerange, valuetransferencoding and value fields shall appear last and in that order.**

- 2 In section 9.2, Request Message Body, "deserializevalue" row:
A container object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648.
- 3 In section 9.6, Request Message Body, "deserializevalue" row:
A container object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648. The object ID of the serialized container object shall match the object ID of the destination container object.
- 4 In section 9.9, Request Message Body, "mimetype" row:
MIME type of the data contained within the value field of the data object
 - This field may be included when creating by value, deserializing, serializing, copying, and moving a data object.
 - This field shall be stored as part of the object.
 - If this field is not specified, the value of "text/plain" shall be assigned as the field value.
 - This field shall not be included when creating a reference.
- 5 In section 9.9, Request Message Body, "deserializevalue" row:
A data object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648.
- 6 In section 9.9, Request Message Body, above the "value" row, add:
valuetransferencoding
JSON String
The value transfer encoding used for the data object value. Two 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,
 - "base64" indicates that the data object may contain arbitrary binary sequences, and 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.
 - This field shall only be included when creating a data object by value,
 - If this field is not specified, the value of "utf-8" shall be assigned as the field value, and
 - This field shall be stored as part of the object.
 Optional
- 7 In section 9.9, Request Message Body, "value" row:
JSON-encoded data
 - If this field is not included, an empty JSON String (i.e., "") shall be assigned as the field value,

- If the value transfer encoding field indicates UTF-8 encoding, the value shall be a UTF-8 string escaped using the JSON escaping rules described in RFC 4627, and
- If the value transfer encoding field indicates base 64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648.

8 In section 9.9, Request Message Body, last footnote:

^aOnly one of these fields shall be specified in any given operation. Except for value, these fields shall not be stored. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response.

9 In section 9.10, Request Headers, "Content-Type" row:

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". Otherwise, the "valuetransferencoding" field of the CDMI data object shall be set to "base64".

10 In section 9.10.8, Example 1:

```
POST to the container URI the data object contents
POST /MyContainer/ HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain;charset=utf-8
```

<object contents>

11 In section 9.10.8, Example 2:

```
POST to the container URI the data object contents
POST /cdmi_objectid/ HTTP/1.1
Host: cloud.example.com
Content-Type: text/plain;charset=utf-8
```

<object contents>

12 In section 9.11, Request Message Body, "deserializevalue" row:

A queue object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648.

Clause 10

1 In section 10.2, Request Message Body, "deserializevalue" row:

A domain object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648.

2 In section 10.4, Request Message Body, "deserializevalue" row:

A domain object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648. The object ID of the serialized domain object shall match the object ID of the destination domain object.

Clause 11

Subclause 11.1

1 In section 11.1, Overview:

Queue **objects** provide first-in, first-out access when storing and retrieving data. A queue writer **POSTs data into a queue**, and a queue reader GETs value(s) from the queue, and subsequently deletes the value(s) to acknowledge receipt of the value(s) that it received. Queuing provides a simple mechanism for one or more writers to send data to a single reader in a reliable way. If supported by the cloud storage system, cloud clients create the queue objects by using the same mechanism used to create data objects.

Queue objects are addressed in CDMI in one of the following ways:

- by URI (e.g., `http://cloud.example.com/queueobject`); and
- by object ID (e.g., `http://cloud.example.com/cdmi_objectid/0000706D0010B84FAD185C425D8B537E`).

The first example addresses the queue object by URI, and the second addresses the queue object by object ID. Every queue object has a single, globally-unique object identifier (ID) that remains constant for the life of the object. Each queue object may also have one or more URI addresses that allow the object to be accessed.

Every queue object has a parent object from which the queue object inherits data system metadata that is not explicitly specified in the queue object itself. Thus, the "receipts.queue" queue object stored at the following URI would inherit data system metadata from its parent container, "finance":

```
http://cloud.example.com/finance/receipts.queue
```

Individual fields within a queue object may be accessed by specifying the field name after a question mark "?" that is appended to the end of the data object URI. Thus, the following URI returns the value field containing the oldest queue value in the response body:

```
http://cloud.example.com/queueobject?value
```

The encoding of the data transported in the queue object value field is dependent on the queue object "valuetransferencoding" field:

- If the value transfer encoding of the object is set to "utf-8", the data stored in the value of the queue object shall be a valid UTF-8 string, and shall be transported as a UTF-8 string in the "value" field.
- If the value transfer encoding of the object is set to "base64", the data stored in the value of the queue object can contain arbitrary binary sequences, and shall be transported as a base 64 encoded string in the "value" field.

Specific ranges of the value of a queue object may be accessed by specifying a byte range after the value field name. Thus, the following URI returns the first thousand bytes of the oldest value enqueued:

```
http://cloud.example.com/queueobject?value:0-999
```

Because a byte range of a UTF-8 string is often not a valid UTF-8 string, the response to a range request shall always be transported in the "value" field as a base 64 encoded string.

Byte ranges are specified as single inclusive byte ranges as per Section 14.35.1 of RFC 2616.

2 In section 11.1.3:

The representations in this subclause are shown using JSON notation. Both clients and servers shall support UTF-8 JSON representation. The request and response body JSON fields may be specified or returned in any order, with the exception that, if present, the valuerange and value fields shall appear last and in that order.

Subclause 11.2

1 In section 11.2.5, Request Message Body, "deserializevalue" row:

A queue object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648.

2 In section 11.2.5, Request Message Body, "deserializevalue" row:

^aOnly one of these fields shall be specified in any given operation. Except for value, these fields shall not be stored. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response.

Subclause 11.3

1 In section 11.3.1, Synopsis:

<range> is a byte range of the queue object value to be returned in the value field. If a byte range is requested, the range returned shall be from the oldest queue value.

2 In section 11.3.6, Response Message Body, "mimetype" row:

MIME types for each queue object value.

- The MIME types of the values are returned, each corresponding to the value in the same position in the JSON array, and
- This field shall only be provided when completionStatus is "Complete", and when there are one or more values enqueued.

3 In section 11.3.6, Response Message Body, "valuerange" row:

The range of bytes of the queue object values to be returned in the value field.

- The value ranges of the values are returned, each corresponding to the value in the same position in the JSON array.
- If a specific value range has been requested, the entry in the value range field shall correspond to the bytes requested. If the request extends beyond the end of the value, the value range field shall indicate the smaller byte range returned.
- The valuerange field shall only be provided when the "completionStatus" field contains "Complete".

4 In section 11.3.6, Response Message Body, above the "value" row:

valuetransferencoding

JSON Array of JSON Strings

The value transfer encoding used for each queue object value. Two value transfer encodings are defined:

- "utf-8" indicates that the queue object value contains a valid UTF-8 string, and shall be transported as a UTF-8 string in the "value" field.
- "base64" indicates that the queue object value may contain arbitrary binary sequences, and shall be transported as a base 64 encoded string in the "value" field.
- The value transfer encodings are returned, each corresponding to the value in the same position in the JSON array, and
- The value transfer encoding field shall only be provided when the "completionStatus" field contains "Complete".

Optional

5 In section 11.3.6, Response Message Body, "value" row:

The oldest enqueued queue object values.

- The values in the JSON array are returned in order from oldest to newest.
- If the value transfer encoding 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 value transfer encoding field indicates base 64 encoding, the corresponding value field shall contain a base 64 encoded string as described in RFC 4648.
- The value field shall only be provided when the "completionStatus" field contains "Complete".

6 In section 11.3.8, Example 1:

The response looks like:

```
HTTP/1.1 200 OK
Content-Type: application/cdm-queue
X-CDMI-Specification-Version: 1.0
```

```
{
  "objectURI": "/MyContainer/MyQueue",
  "objectID": "00007E7F00104BE66AB53A9572F9F51E",
  "objectName": "MyQueue",
  "parentURI": "/MyContainer/",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/queue/",
  "completionStatus": "Complete",
  "metadata": {},
  "queueValues": "1-2",
  "mimetype": [
    "text/plain"
  ],
  "valuerange": [
    "0-19"
  ],
  "valuetransferencoding": [
    "utf-8"
  ],
  "value": [
    "First Enqueued Value"
  ]
}
```

```
    ]
  }
```

Subclause 11.4 & 11.6

- 1 In section 11.4.4, Request Message Body, "deserialize" row:
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.
- 2 In section 11.4.4, Request Message Body, "deserializevalue" row:
A queue object serialized as specified in Clause 15 "Serialization/Deserialization" and encoded using base 64 encoding rules described in RFC 4648. The object ID of the serialized **queue** object shall match the object ID of the destination **queue** object.
- 3 In section 11.6.4, Request Message Body, "mimetype" row:
MIME type of the data to be enqueued into the queue object.
 - This field shall be stored as part of the object.
 - If this field is not specified, the value of "text/plain" 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.
- 4 In section 11.6.4, Request Message Body, "copy" row:
URI of a CDMI data object or queue from which the value shall be copied and enqueued.
 - If a URI to a data object is provided, the value, mimetype and valuetransferencoding from the data object is used to enqueue the new item into the queue.
 - If a URI to a queue object is provided, the corresponding value, mimetype and valuetransferencoding of the specified number of enqueued items are copied to the queue.
- 5 In section 11.6.4, Request Message Body, "move" row:
URI of a CDMI data object or queue from which the value shall be moved and enqueued.
 - If a URI to a data object is provided, the value, mimetype and valuetransferencoding from the data object is used to enqueue the new item into the queue, and the data object is atomically deleted.
 - If a URI to a queue object is provided, the corresponding value, mimetype and valuetransferencoding of the specified number of enqueued items are transferred to the queue and atomically removed from the source queue.
- 6 In section 11.6.4, Request Message Body, above the "value" row, add:
valuetransferencoding
JSON Array of JSON Strings
The value transfer encoding used for the queue object value. Two value transfer encodings are defined:

- "utf-8" indicates that the queue value contains a valid UTF-8 string, and shall be transported as a UTF-8 string in the "value" field.
- "base64" indicates that the queue value may contain arbitrary binary sequences, and shall be transported as a base 64 encoded string in the "value" field. Setting the contents of the queue 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 specified, the value of "utf-8" shall be assigned as the field value.
- This field shall be stored as part of the object.

Optional

7 In section 11.6.4, Request Message Body, "value" row:

Data to be enqueued into the queue.

- If the corresponding value transfer encoding 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 value transfer encoding field indicates base 64 encoding, the value shall be first encoded using the base 64 encoding rules described in RFC 4648.

8 In section 11.6.4, Request Message Body, last footnote:

^aOnly one of these fields shall be specified in any given operation. Except for value, these fields **shall not be stored**. If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response.

9 In section 11.6.8, New Example 5:

EXAMPLE 5 POST to the queue URI two new values with base 64 transfer encoding:

```
POST /MyContainer/MyQueue HTTP/1.1
Host: cloud.example.com
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0
```

```
{
  "mimetype": [
    "text/plain",
    "text/plain"
  ],
  "valuetransferencoding": [
    "utf-8",
    "base64"
  ],
  "value": [
    "First",
    "U2Vjb25k"
  ]
}
```

The following shows the response.

HTTP/1.1 201 Created

Clause 15

1 In section 15.1 Exporting Serialized Data:

A canonical encoding of the data is obtained by creating a new data object and specifying that the source for the creation is to serialize a given CDMI data object, container, or queue. On a successful serialization, the result shall be a data object that is created with the serialized data as its value. If a container has an exported block protocol, the serialized data may contain the block-by-block contents of that container along with its metadata as if it were a data object.

The resulting data object that is produced is the canonical representation of the selected data object, container and children, or queue.

- If the source specified is a data object, the canonical format shall contain **all data object fields, including the value, value transfer encoding and metadata.**
- If the source specified is a queue, the **canonical format shall contain all queue fields, including the value and value transfer encoding of enqueued items, along with the metadata of the queue itself.**
- If the source specified is a container, **the canonical format shall contain all container fields, recursively including all children of the container. If a user tries to serialize a container that includes children that the user who is performing the serialization operation does not have permission to read, these objects shall not be included in the resulting serialized object.**

2 In section 15.2 Importing Serialized Data:

Canonical data may be deserialized back into the cloud by creating a new data object, container, or queue and by specifying that the source for the creation is to deserialize a given CDMI data object, **or by specifying the serialized data in base 64 encoding in the "deserializevalue" field.**

The destination may or may not exist previously. If not, a "create" operation is performed. If a container already exists, an "update" operation with serialized **children shall update the container and all children. If the children are not included in a container deserialization, only the container object is updated.** Data objects are recreated as specified in the canonical format, including all metadata and the data object ID. If the serialized data object contains block-by-block data from a container that was exporting a block protocol, the container shall be created with that data as its new value, but the exported block protocol shall be set up with a separate update.

3 In section 15.2.2 Example JSON Canonical Serialized Format:

In this example, a data object and a queue in a container have been selected for serialization.

```
{
  "objectURI": "/MyContainer/",
  "objectID": "00007E7F00104BE66AB53A9572F9F51E",
  "objectName": "MyContainer/",
  "parentURI": "/",
  "domainURI": "/cdmi_domains/MyDomain/",
  "capabilitiesURI": "/cdmi_capabilities/container/",
  "completionStatus": "Complete",
```

```

"metadata": {},
"exports": {
  "OCCE/iSCSI": {
    "identifier": "00007E7F00104BE66AB53A9572F9F51E",
    "permissions": "00007E7F00104EB781F900791C70106C"
  },
  "Network/NFSv4": {
    "identifier": "/users",
    "permissions": "domain"
  }
},
"childrenrange": "0-1",
"children": [
  {
    "objectURI": "/MyContainer/MyDataObject.txt",
    "objectID": "00007E7F00104BE66AB53A9572F9F51E",
    "objectName": "MyDataObject.txt",
    "parentURI": "/MyContainer/",
    "domainURI": "/cdmi_domains/MyDomain/",
    "capabilitiesURI": "/cdmi_capabilities/dataobject/",
    "completionStatus": "Complete",
    "mimetype": "text/plain",
    "metadata": {},
    "valuerange": "0-36",
    "valuetransferencoding": "utf-8",
    "value": "This is the Value of this Data Object"
  },
  {
    "objectURI": "/MyContainer/MyQueue",
    "objectID": "00007E7F00104BE66AB53A9572F9F51E",
    "objectName": "MyQueue",
    "parentURI": "/MyContainer/",
    "domainURI": "/cdmi_domains/MyDomain/",
    "capabilitiesURI": "/cdmi_capabilities/queue/",
    "completionStatus": "Complete",
    "metadata": {},
    "queueValues": "0-1",
    "mimetype": [
      "text/plain",
      "text/plain"
    ],
    "valuetransferencoding": [
      "utf-8",
      "utf-8"
    ],
    "valuerange": [
      "0-2",
      "0-3"
    ],
    "value": [
      "red",
      "blue"
    ]
  }
]
}

```

```
    ]
  }
```

To allow efficient deserialization in stream mode when serializing containers to JSON, the children array shall be the last item in the container JSON object.

Clause 20

- 1 Add to section 20.1 above Table 124:

This approach allows the matching against arbitrarily nested metadata structures. To query the value of objects, the value field is included within the query request. Values are always represented using base 64 encoding in queries. This query is shown as follows:

```
{
  [
    {
      "value": "==" Ymx1ZQ=="
    }
  ]
}
```

Query against the value of objects is optional, and indicated by the presence of the "cdmi_query_value" capability.

- 2 In section 20.2 update:

If a client wants to have all fields and metadata returned for each matching object, the following cdmi_results_specification shall be used:

```
{
  "cdmi_results_specification" : ""
}
```

The value field is always returned in base 64 encoding when included in a query result, where the value transfer encoding field indicates the encoding that should be expected if a GET to read the object is performed.

CHANGES Made all of the specified changes to Version 1.0.11.

Trac Ticket #480 Remove Base64 encoding of cdmi_hash

Description: There is only one place where base64 is referred to in the spec (outside of PEM encoded certificates), and we should change it over to using hex encoding to be consistent with hash representations when displaying certificates, etc.

Proposed change for review:

In Table 19 in section 16.3, in the cdmi_hash row, change the description from:

"The hash of the value of the object, encoded as a base64 string. This metadata field shall be present when the "cdmi_value_hash" data system metadata for the object or a parent object indicates that the value of the object should be hashed."
to

"The hash of the value of the object, hex-encoded as a string. This metadata field shall be present when the "cdmi_value_hash" data system metadata for the object or a parent object indicates that the value of the object should be hashed."

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified change.

Trac Ticket #481 Restrict JSON Representations to UTF-8

Description: It turns out that JSON supports UTF-16 and UTF-32 string encodings in both big and little endian formats.

I propose that we add text stating that CDMI shall use UTF-8 for all representations.

1 Under section 8.1.3 Data Object Representations, replace:

"The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support JSON representation."

with:

"The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support UTF-8 JSON representation."

2 Under section 9.1.3 Container Object Representations, replace:

"The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory fields and may support the optional fields. The parameter fields may be specified or returned in any order. Both clients and servers shall support JSON representation."

with:

"The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory fields and may support the optional fields. The parameter fields may be specified or returned in any order. Both clients and servers shall support UTF-8 JSON representation."

3 Under section 10.1.4 Domain Object Representations, replace:

"The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support JSON representation."

with:

"The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support UTF-8 JSON representation."

- 4 Under section 11.1.3 Queue Object Representations, replace:
 "The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support JSON representation."
 with:
 "The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support UTF-8 JSON representation."
- 5 Add sub-section just above start of section 12.2:
 12.1.8 Capability Object Representations
 "The representations in this section are shown using JSON notation. A conforming implementation shall support the mandatory parameters and may support the optional parameters. The parameter fields may be specified or returned in any order. Both clients and servers shall support UTF-8 JSON representation."

description modified (diff)

summary changed from Restrict JSON Strings to UTF-8 to Restrict JSON Representations to UTF-8

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #482 Stipulate that value/children come last

Description: When processing CDMI operations, both as a client and as a server, it is best for the value/children JSON field to come last in the JSON content presented in the request body/response body.

For example, Data System Metadata may change how one processes the value (location restrictions, synchronous copies, etc.), and you need to know about this before you receive the value to process the transaction efficiently.

I propose that we add text to the standard that stipulates that the valuerange/value pair, if present, and the childrenrange/children, if present, come as the last fields in the JSON.

Reviewed on 2011-05-19 at the face-to-face meeting, and approved. Spec text needed.

Proposed spec text changes:

In 8.1.3 Data Object Representations:

In 9.1.3 Container Object Representations:

In 10.1.4 Domain Object Representations:
In 11.1.3 Queue Object Representations

Replace:

"The parameter fields may be specified or returned in any order."
with

"The request and response body JSON fields may be specified or returned in any order, with the exception of the valuerange and value fields and childrange and children fields always coming last."

Rewording by the TWG - Spec text changes:

In 8.1.3 Data Object Representations:
In 9.1.3 Container Object Representations:
In 10.1.4 Domain Object Representations:
In 11.1.3 Queue Object Representations

Replace:

"The parameter fields may be specified or returned in any order."
with

"The request and response body JSON fields may be specified or returned in any order, with the exception that, if present, the childrenrange, children, valuerange and value fields shall appear last, in that order."

This change does not appear to have been made in draft 1.0.1i to sections 9.1.3 or 10.1.4.

- CHANGES**
- Replaced the specified text in 8.1.3, 10.1.4, and 11.1.3 (version 1.0.1i).
 - Replaced the specified text in 9.1.3 (version 1.0.1j).

All specified changes made in Version 1.0.1j.

Trac Ticket #486 Update operations should not require an Accept header

Description: As discussed in ticket #425, as updates do not return a response body, they should not require an Accept header in the request headers.

Propose the following spec text changes:

- 1 In Section 8.6, Update a Data Object (CDMI Content Type), remove the "Accept" row from the Request Header table.
- 2 In Section 8.7, Update a Data Object (Non-CDMI Content Type), remove the "Accept" row from the Request Header table.
- 3 In Section 9.6, Update a Container (CDMI Content Type), remove the "Accept" row from the Request Header table.
- 4 In Section 10.4, Update a Domain (CDMI Content Type), remove the "Accept" row from the Request Header table.

- 5 In Section 11.4 Update a Queue Object (CDMI Content Type), remove the "Accept" row from the Request Header table.
- 6 In all examples that return "204 No Content", there should not be an Accept header in the request.

Reviewed on 2011-05-19 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes in items 1-6.

Trac Ticket #487 Add section to 5.13 discussing URI escaping

Description: All CDMI fields that are used in URIs must be URI escaped.

Discussed as part of ticket #435.

Proposed spec text: Add to section 5.13:

5.13.3 URI Escaping

Percent escaping of reserved characters specified in RFC 3986 shall be used for all text strings used in URIs. This includes user-supplied metadata names, object names, container names and domain names.

Revised proposed spec text: Add to section 5.13:

5.13.3 URI Escaping

Percent escaping of reserved characters specified in RFC 3986 shall be applied to all text strings used in URIs. This includes user-supplied metadata names, object names and URIs, container names and domain names. This escaping shall be applied to the use of these strings in HTTP request and response URIs and in JSON string representations.

For example, an object URI containing the reserved character "@" would be represented in JSON as:

```
{
  "objectURI": "%40MyContainer/"
}
```

A user-provided metadata name containing the reserved character "@" would be represented in JSON as:

```
{
  "metadata" : {
    "%40name": "comment"
  }
}
```

TWG comments: Danger of double-encoding unless the URI escaping is consistently done immediately before the GET. We need to be very clear about what is already escaped and what isn't.

- 1 Updated proposed spec text: Add to section 5.13:
5.13.3 URI Escaping

Percent escaping of reserved characters specified in RFC 3986 shall be applied to all text strings used in URIs. This includes user-supplied field names, metadata names, object names, container names and domain names when used in URIs. Field names and values contained within the request body and response body shall not be escaped.

For example, a client retrieving a metadata item named "@user" from a container object with the name of "@MyContainer" would perform the following request:

```
GET /%40MyContainer/?objectName;metadata:%40user HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-container
X-CDMI-Specification-Version: 1.0
```

The response shall be:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0

{
  "objectName": "@MyContainer",
  "metadata": {
    "@user": "test"
  }
}
```

Note that we don't need to reserve "/" and "?" in object names, because one cannot create an object with these characters in the name.

Reviewed on 2011-05-20 at the face-to-face meeting, and approved for incorporation into the spec.

- 2 In the 1.0.1i spec, in the third sentence of clause 5.13.3, the text is incorrect: It should be "@MyContainer" without a "?", not "@MyContainer?".

This is most likely due to trac adding characters.

CHANGES Made the specified changes to item #1; reopened ticket and fixed wiki error in item #2.

Trac Ticket #488 Consider "application/octet-stream" as better MIME default

Description: Currently in the spec, if a MIME type is omitted when creating an object, the default MIME type of "text/plain" is used. Given that CDMI can be used to store binary objects, using "application/octet-stream" would a safer default.

Proposed spec change:

Change all instances of the string:

"If this field is not specified, the value of "text/plain" shall be assigned as the field value."

with:

"If this field is not specified, the value of "application/octet-stream" shall be assigned as the field value."

Reviewed on 2011-05-19 at the face-to-face meeting, and reluctantly approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #489 Storage System Metadata cdmi_size only refers to "data items"

Description: The existing description text for the cdmi_size storage system metadata only refers to "data items". We should update this text to refer to data objects and queues.

Also, we should discuss about containers, capabilities and domains.

Proposed spec changes:

In table 19, under cdmi_spec, replace:

"The number of bytes stored in the data item."

with:

"For data objects, the number of bytes in the value field, and for queues the total number of bytes for all enqueued values."

TWG Discussion:

Q. Should this be the size of the value, or the size of the object?

- The HTTP content-length is related to the transaction, rather than the size of the object.
- The valuerange is related to the transaction, and sometimes will reflect the size of the value of the object.
- Nothing currently corresponds to the actual stored size of the object (metadata plus value)

TWG: This should represent a size that you are billed for.

Q. What about containers and domains?

We can now have a generic definition:

Proposed spec text:

In table 19, under the cdmi_size description, replace:

"The number of bytes stored in the data item."

with:

"The number of bytes consumed by the object."

CHANGES Made the specified changes.

Trac Ticket #493 Updates missing copy, serialize, deserializevalue

Description (last modified by dslik) (diff)

Updates should permit operations that preserve the object ID but update the contents of the object. Likewise, we should provide guidance in the update section that operations that result in a new Object ID will be treated as a create.

The following spec text updates should be considered to clarify this:

- 1 In section 8.6 Update a Data Object, add the following rows to the request body JSON table, under the "domainURI" row:

deserialize

JSON String

URI of a serialized CDMI data object that shall be deserialized to update an existing data object. The object ID of the serialized data object must match the object ID of the destination data object, or else the deserialize will be handled as a create (see Section 8.2).

Optional*

copy

JSON String

URI of a CDMI data object or queue that shall be copied into the existing data object.

Optional*

deserializevalue

JSON String

A data object serialized as specified in Chapter 15, "Serialization/Deserialization". The object ID of the serialized data object must match the object ID of the destination data object, or else the deserialize will be handled as a create (see Section 8.2).

Optional*

Note: The * needs to be added to the "Optional" in the "value" row.

Note: The ""Only one of these parameters shall be specified in any given operation, and except for value, these fields are not persisted." field needs to be added to the end of the table.

- 2 In section 9.6 Update a Container Object, add the following rows to the request body JSON table, under the "snapshots" row:

deserialize

JSON String

URI of a serialized CDMI container object that shall be deserialized to update an existing container object. The object ID of the serialized container object must match the object ID of the destination container object, or else the deserialize will be handled as a create (see Section 9.2).

If the serialized container does not contain children, the update is applied only to the container object and any existing children are left as-is. If the serialized container object does contain children, then creates, updates and deletes are recursively applied for each child, depending on the differences between the provided serialized state and the current state of the child.

Optional*

copy

JSON String

URI of a CDMI container object that shall be copied into the existing container object. Only the contents of the container itself shall be copied, not any children of the container.

Optional*

deserializevalue

JSON String

A container object serialized as specified in Chapter 15, "Serialization/Deserialization". The object ID of the serialized container object must match the object ID of the destination container object, or else the deserialize will be handled as a create (see Section 9.2).

If the serialized container does not contain children, the update is applied only to the container object and any existing children are left as-is. If the serialized container object does contain children, then creates, updates and deletes are recursively applied for each child, depending on the differences between the provided serialized state and the current state of the children.

Optional*

Note: The * needs to be added to the "Optional" in the "value" row.

Note: The "*Only one of these parameters shall be specified in any given operation, and except for value, these fields are not persisted." field needs to be added to the end of the table.

- 3 In section 10.4 Update a Domain, add the following rows to the request body JSON table, under the "metadata" row:

copy

JSON String

URI of a CDMI domain object that shall be copied into the existing domain object. Only the metadata and membership of the domain shall be copied, not any sub-domains of the domain.

Optional*

deserialize

JSON String

URI of a serialized CDMI domain object that shall be deserialized to update an existing domain object. The object ID of the serialized domain object must match

the object ID of the destination domain object, or else the deserialize will be handled as a create (see Section 10.2).

If the serialized domain does not contain children, the update is applied only to the domain object and any existing children are left as-is. If the serialized domain object does contain children, then creates, updates and deletes are recursively applied for each child, depending on the differences between the provided serialized state and the current state of the children.

Optional*

deserializevalue

JSON String

A domain object serialized as specified in Chapter 15, "Serialization/Deserialization". The object ID of the serialized domain object must match the object ID of the destination domain object, or else the deserialize will be handled as a create (see Section 9.2).

If the serialized domain does not contain children, the update is applied only to the domain object and any existing children are left as-is. If the serialized domain object does contain children, then creates, updates and deletes are recursively applied for each child, depending on the differences between the provided serialized state and the current state of the children.

Optional*

Note: The * needs to be added to the "Optional" in the "value" row.

Note: The ""Only one of these parameters shall be specified in any given operation, and except for value, these fields are not persisted." field needs to be added to the end of the table.

- 4 In section 11.4 Update a Queue Object, add the following rows to the request body JSON table, under the "domainURI" row:

deserialize

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 data object must match the object ID of the destination data object, or else the deserialize will be handled as a create (see Section 11.2).

All enqueued items in the serialized queue shall be added to the destination queue.

Optional*

copy

JSON String

URI of a CDMI queue queue that shall be copied into the existing queue object. Queue copy does not copy enqueued items. See section 11.6 to copy enqueued items)

Optional*

deserializevalue

JSON String

A data object serialized as specified in Chapter 15, "Serialization/Deserialization". The object ID of the serialized data object must match the object ID of the destination data object, or else the deserialize will be handled as a create (see Section 11.2).

All enqueued items in the serialized queue shall be added to the destination queue.

Optional*

Note: The * needs to be added to the "Optional" in the "value" row.

Note: The "*Only one of these parameters shall be specified in any given operation, and except for value, these fields are not persisted." field needs to be added to the end of the table.

description modified (diff)

Note: Replace "must" with "shall"

TWG discussion

What about when:

- A user deserializes an object to a path where there is not an object at that path, but there is an object in the system with that Object ID?
- A user deserializes an object to a path where there is an object at that path, but the object at that path has a different Object ID?

- 5 To avoid the second scenario here, we should amend the above spec text changes from:

"The object ID of the serialized data object must match the object ID of the destination data object, or else the deserialize will be handled as a create (see Section 8.2)."

to

"The object ID of the serialized data object must match the object ID of the destination data object."

(with data object replaced with container, etc...)

The TWG discussed what it would mean to do a "move" when an object exists at the destination URI. This could have value as an atomic delete-create. However, this was viewed to add significant complexity for little additional value, so this was not agreed to be added to the spec and left unspecified (no move field for update) .

Reviewed on 2011-05-20 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #494 Cleanup of Query, Notification & Logging queue status metadata

Description: When a query, notification or logging queue is created, there may be a lag in time between when the queue is created and when the query, notification or logging data is being enqueued into the queue.

In order to provide for this, we should add a "Pending" value to `cdmi_query_status`, `cdmi_notification_status` and `cdmi_logging_status`, respectively.

In addition, the spec text should not refer to logging and notification as a "query".

Proposed spec text changes incorporating in the changes from ticket #591:

- 1 In section 17.5, above Table 118, replace:
"When creating a logging queue, the metadata described in Table 118 shall be provided."
with:
"When creating a logging queue, the metadata described in Table 118 shall be provided. Attempts to alter metadata in this table will result in an HTTP 403 Forbidden HTTP status code. Once a logging queue has been created, with the exception of `cdmi_queue_type`, the metadata items in this table cannot be altered. `cdmi_queue_type` can only be removed, indicating to the system that the logging queue shall no longer receive log messages and shall be treated as a regular CDMI queue object."
- 2 In section 17.6, replace:
"Table 119 describes the system-created metadata that provides details on the completion status of the logging queue."
with
"Table 119 describes the system-created metadata that provides details on the status of the logging queue."
- 3 In Table 119 - Logging Status Metadata, replace the description;
"Indicates if the query is in progress or complete. The three values defined are "Error", "Processing", and "Complete"."
with
"A string indicating the state of the logging queue. Defined values are:
 - Processing - Indicates that the logging queue is scanning for results.
 - Halted - Indicates that new log messages will no longer be enqueued.
 - Current - Indicates that the logging queue contained all log messages that can be found at this time.
 - Error - Indicates that the logging queue metadata is not valid, or other errors were encountered that prevented logging messages from being enqueued. Arbitrary vendor defined text may follow the string "Error".
- 4 In section 19, replace:

"Table 121 describes the system-created metadata that provides details on the completion status of the notification queue."

with

"Table 121 describes the system-created metadata that provides details on the status of the notification queue."

5 In Table 121 - Notification Status Metadata, replace the description:

"Indicates if the query is in progress or complete. The two values defined are "Error", "Processing", and "Complete"."

with

"A string indicating the state of the notification queue. Defined values are:

- Processing - Indicates that the notification queue is scanning for results.
- Halted - Indicates that new notifications will no longer be enqueued.
- Current - Indicates that the notification queue contained all notifications that can be found at this time.
- Error - Indicates that the notification queue metadata is not valid, or other errors were encountered that prevented notification messages from being enqueued. Arbitrary vendor defined text may follow the string "Error".

If this metadata item does not exist, then notifications have not yet started being enqueued."

6 In section 20, replace:

"Table 123 describes the system-created metadata that provides details on the completion status of the query queue."

with

"Table 123 describes the system-created metadata that provides details on the status of the query queue."

7 In Table 27 - Query Status Metadata, replace the description;

"A string indicating if the query is in progress or has completed. The value shall be the string "Processing", the string "Complete", or an error string starting with the value "Error"."

with

"When present, this metadata item indicates the state of the query queue. Defined values are:

- Processing - Indicates that the query queue is scanning for results.
- Halted - Indicates that new query results will no longer be enqueued.
- Current - Indicates that the query queue contained all query results that can be found at this time.
- Error - Indicates that the query queue metadata was not valid, or other errors were encountered that prevented all query results from being enqueued. Arbitrary vendor defined text may follow the string "Error".

TWG: Approved with one abstain.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #495 Spec doesn't define what an empty scope means

Description: In section 20.1, we don't define what an empty scope means.

It could either match against nothing (no matching statements return true, thus false), or it could be defined to mean match everything (no matching statements return false, thus true).

I propose that we define that an empty scope evaluates to true, but we should discuss this at the next face-to-face.

Proposed spec test:

Under Section 20.1, replace the first paragraph:

"Each JSON object within the scope specification represents a set of conditions that shall all be true in order for the query to return an object as part of the query results (a logical AND relationship). Multiple JSON objects are used to express logical OR relationships."

with

"Each JSON object within the scope specification represents a set of conditions that shall all be true in order for an object to be considered to match against the scope (a logical AND relationship). For queries, a matching object would be returned in the query results. An empty scope specification is considered to evaluate to true. Multiple JSON objects are used to express logical OR relationships, where if any JSON object in the scope evaluates to true, then the object will be considered to have matched against the scope."

Reviewed on 2011-05-20 at the face-to-face meeting, and approved for incorporation into the spec.

CHANGES Made the specified changes.

Trac Ticket #498 Cornucopia of Confusion

Description: See Ralph's comments on page 40.

We need to review our terms, and decide on consistent names for each concept we are describing and referring to.

- 1 Change "metadata elements" to "metadata items".
- 2 In 15.3.1, change "data element" to "data value".
- 3 In A.4.1.2.3 Certificate Management, in 1st sentence, delete "identifying CDMI entity".
- 4 Do a global search on client and server and make sure terms are used appropriately.
- 5 Delete all references to Annex B.
- 6 Make sure all lists are formatted per ISO standards.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #499 Delete in Spec is unclear about non-CDMI use

Description: urrently we don't formally define a non-CDMI variant for DELETE. The only difference between a CDMi variant and a non-CDMI variant is the requirement for the X-CDMI-Specification-Version.

There's a couple of options for resolving this:

- Drop the X-CDMI-Specification-Version from the DELETE sections, and indicate that it is for both CDMI and non-CDMI.
- Duplicate the section, with the Non-CDMI copy having the X-CDMI-Specification-Version header removed.

TWG: Preference is to duplicate the DELETE section to create a Non-CDMI variant.

The following changes will need to be made:

- 1 Duplicate section 8.8 to create a new section 8.9
- 2 The section heading for 8.8 should include (CDMI Content Type)
- 3 The section heading for 8.9 should include (Non-CDMI Content Type)
- 4 In 8.9, remove the response headers table, and replace it with "None Specified."
- 5 In 8.9, remove the X-CDMI-Specification-Version header from the example.
- 6 Duplicate section 9.7 to create a new section 9.8
- 7 The section heading for 9.7 should include (CDMI Content Type)
- 8 The section heading for the new section 9.8 should include (Non-CDMI Content Type)
- 9 In new section 9.8, remove the response headers table, and replace it with "None Specified."
- 10 In new section 9.8, remove the X-CDMI-Specification-Version header from the example.

CHANGES Made the changes to Items #1-10, except for items 4 and 9, as the response header tables already had "None Specified."

Trac Ticket #502 Add objectType field to JSON

Description: There is no vendor-independent way to query for all objects of a certain type, eg, all queue objects, or all data objects. In notifications and query results, there is no vendor-independent way to distinguish a queue object from a data object, etc.

This would be resolved if we added a field, "objectType", that contains the MIME type of the object. For example, to a data object, we would add the following JSON field:

```
"objectType" : "application/cdmi-object",
```

The below spec text has been approved by the TWG:

- 1 For each example, replace "objectURI" with "objectType", and replace the value with the the Content-Type header value:

E.g.,

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-capability
X-CDMI-Specification-Version: 1.0.1
{
  "objectURI" : "/cdmi_capabilities/",
  "objectID" : "00007E7F0010CEC234AD9E3EBFE9531D",
  "objectName" : "cdmi_capabilities/",
  "parentURI" : "/",
  ...
}
```

would change the line:

```
"objectURI" : "/cdmi_capabilities/",
into
"objectType" : "application/cdmi-capability",
```

- 2 In Response Message Body Tables, replace "objectURI" with "objectType", and replace the description with the string "application/cdmi-object", "application/cdmi-container", "application/cdmi-queue", "application/cdmi-domain", and "application/cdmi-capability" respectively. (in quotes as shown below)

For example, in 8.2.7, the text would be changed from:

```
objectURI
JSON String
URI of the object as specified in the request
Mandatory
to
objectType
JSON String
"application/cdmi-object"
Mandatory
```

- 3 In 15.3.2, replace the three "objectURI" lines:

```
"objectURI" : "/MyContainer/",
"objectURI" : "/MyContainer/MyDataObject.txt",
"objectURI" : "/MyContainer/MyQueue",
with
"objectType" : "application/cdmi-container",
"objectType" : "application/cdmi-object",
"objectType" : "application/cdmi-queue",
respectively.
```

- 4 In Example 1 under Section 19 and 20, replace:

```
"objectURI" : "starts /sandbox",
with
"parentURI" : "starts /sandbox",
```

- 5 In the JSON under section 20.1, replace:
 "objectURI" : "/MyContainer/MyDataObject.txt",
 with
 "objectType" : "dataobject",
- 6 In section 20.1, replace:
 "When a URI is used as the constant for the equals and not equals operators against objectURI, parentURI, domainURI, and capabilitiesURI, either a URI by path or URI by Object ID can be specified and are considered interchangeable."
 with
 "When a URI is used as the constant for the equals and not equals operators against parentURI, domainURI, and capabilitiesURI, either a URI by path or URI by Object ID can be specified and are considered interchangeable."
- 7 In section 20.1, replace:
 "If an Object ID is used in a query scope, such as in the objectID field, in the objectName field, when the objectPath is set to "/cdmi_objectid/", and as part of a URI compared against the objectURI, parentURI, capabilitiesURI and domainURI fields, all object IDs shall be processed such that they are case insensitive."
 with
 "If an Object ID is used in a query scope in the objectID field or the parentID field, all object IDs shall be processed such that they are case insensitive."
- 8 In section 20.1, replace:
 "For most common use cases, either the Object ID or Object URI will be requested in the cdmi_results_specification. If the Object URI is included, it is up to the implementation to choose when a Object URI by path or an Object URI by ID should be returned, and both are equally valid. If the Object Name and/or parent URI are included, and paths are supported, the implementation shall return the object name and object path, respectively. If the Object URI is included in the results and the object has a path, then the path shall be returned in this field."
 with
 "For most common use cases, clients request Object ID, objectName and parentURI, or both in the cdmi_results_specification. If the parentURI or objectName is requested, the field shall only be returned for objects existing in a container."

Once all of these changes are made, there shall be no more instances of "objectURI" in the spec.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #503 Clarify definition of numeric in numeric query matching expressions

Description: In Section 18, Table 120:

Proposed spec text changes:

- 1 In section 20.1, Table 28 - Query Matching Expression, for the "==" and "!=" expressions:
 - Add the following paragraph to the end of the description:

"If the matching expression starts with a "#" character (for example, "#=="), the value of the field is considered to be numeric for the purposes of comparison. Numeric constant strings shall be processed according to the JSON number representation described in [RFC4627]. A numeric matching expression shall be considered to be non-matching against a non-numeric field value."
 - Be sure to update the example with the correct operator.
- 2 In section 20.1, Table 28 - Query Matching Expression, for the ">", ">=", "<", and "<=" expressions:

Replace:

"If the constant starts with a "+" or "-" sign, the value of the field is considered to be numeric for the purposes of comparison."

With:

"If the matching expression starts with a "#" character (for example, "#>"), the value of the field is considered to be numeric for the purposes of comparison. Numeric constant strings shall be processed according to the JSON number representation described in [RFC4627]. A numeric matching expression shall be considered to be non-matching against a non-numeric field value."

Be sure to update the example with the correct operator.
- 3 In Section 22.1, in the first example, replace:

"cdmi_size" : "> +100000"

With

"cdmi_size" : "#> 100000"

TWG: Comment is that this is "less ugly", but there is still general dissatisfaction with the query language and a desire to address this in 1.1.

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #506 Editorial changes throughout document (part 1)

Description: Per Ralph's markups in ENDLTxReviewofCDMI[1-8].pdf:

- 1 Change paper size to A-4.
- 2 Change copyright to ISO/IEC in headers.
- 3 Change page numbers to Western Arabic, continuous
- 4 Remove all cross-references to glossary terms.
- 5 Change all references to chapters to clauses (change all x-ref formats).
- 6 Change all multi-page tables to show continuation (use TableContinuations.fm).

- 7 Use periods to indicate the end of sentences, not the end of fragments (i.e., Table 1).

CHANGES Made the specified changes throughout Version 1.0.1j.

Trac Ticket #507 Editorial changes: Title Page, TOC, Introduction, Scope, and References

Description: Per Ralph's markups in ENDLTxReviewofCDMI[1-8].pdf:

Title Page:

- 1 Change title page (ISOCover.fm).
- 2 Remove Revision History (before submission).
- 3 (Permission paragraph) SNIA hereby grants... - The tpage1.fm file in ISO/IEC provided FrameMaker 6 templates (see <http://www.iso.org/templates>) provides some interesting ideas that might substitute for this. I cannot vouch for them because I have never seen them used.
- 4 Change copyright to ISO/IEC on title page (see example).

Table of Contents

- 1 In TOC, indent Foreword and Introduction lines to align with title text.
- 2 If TOC entries line wrap, format to ensure at least three dots before page number
- 3 Replace Foreword with ISOForeword.fm

Introduction

- 1 Remove subclauses from Introduction and leave as two paragraphs.
- 2 Replace first paragraph of Introduction as follows:
 Replace:
 This document tells you how to access cloud storage and to manage the data stored there. The intended audience is application developers who are implementing or using cloud storage.
 With
 This International Standard is intended for application developers who are implementing or using cloud storage. It documents how to access cloud storage and to manage the data stored there.

1 Scope

Add References to same page as Scope (don't start a new chapter.)

2 Normative References

- 1 Change 2.1 Normative References to 2 Normative References.
- 2 Move all Informative References to last Annex (Informative) – Bibliography (create new chapter).

- 3 After moving informative references to bibliography annex, duplicate format for all references as shown on page 29 of ENDLTxReviewofCDMI[1-8].pdf and incorporate notes from page 28 and page 30 of same document.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #509 10.2 Create Domain missing "move" field

Description: The "move" field to allow a domain to be renamed is missing.

Proposed spec text:

Add below existing "copy" field in the Request Message Body table:

move

JSON String

URI of an existing local CDMI domain (source URI) that shall be relocated, along with all child domains, to the URI specified in the PUT. The contents of the domain and all sub-domains shall be preserved by a move.

If there are insufficient permissions to read the objects at the source URI, write the objects at the destination URI, or delete the objects at the source URI, or if any of these operations fail, the move shall return a 400 Bad Request result code, and the source and destination are left unchanged.

Optional*

Note: We subsequently updated the text for move in a previous trac ticket. Here is an updated proposed spec text:

move

JSON String

URI of an existing local CDMI domain object (source URI) that shall be relocated, along with all child domains, to the URI specified in the PUT. The contents of the domain and all sub-domains, including the object ID, shall be preserved by a move, and the domain and sub-domains of the source URI shall be removed after the objects at the destination have been successfully created.

If there are insufficient permissions to read the objects at the source URI, write the objects at the destination URI, or delete the objects at the source URI, or if any of these operations fail, the move shall return a 400 Bad Request result code, and the source and destination are left unchanged.

Optional*

Discussed if this should support both remote or local. Decision was to just define local.

TWG approved the above spec text.

CHANGES Made the specified change.

Trac Ticket #510 "cdmi_encryption" table row in Table 20 split into two parts

Description: Minor formatting--

- The "cdmi_encryption" table row in Table 20 is split into two parts in 1.0.1i.
- The last item is also missing a bullet.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #511 Reversed cdmi_hash descriptions

Description: The descriptions for "cdmi_value_hash" in Table 12 and the description of "cdmi_value_hash" in Table 20 are reversed.

- 1 In Table 104, "cdmi_value_hash", replace:

"If present, this capability lists the hash algorithms and lengths supported. If absent, objects shall not be hashed. When a CDMI implementation supports hashing, the system-wide capability of "cdmi_security_data_integrity" specified in Table 102 of 12.1.1 shall be set to "true". Otherwise, it shall not be present, indicating that there is no support for hashing."

with

"If present, this capability shall list the hash algorithms and lengths supported. If absent, object values shall not be hashed. Values shall be in the form of "Algorithm Length", for example, "SHA256". When a CDMI implementation supports hashing, the system-wide capability of "cdmi_security_data_integrity" specified in Table 102 of 12.1.1 shall be set to "true". Otherwise, it shall not be present, indicating that there is no support for hashing."
- 2 In Table 116, "cdmi_value_hash", replace:

"If present, this capability metadata lists the hash algorithm/lengths supported. If absent, objects shall not present a hash value as system metadata. Values are in the form of "Algorithm Length", for examplee.g., "SHA256". When a CDMI implementation supports hashing, the system-wide capability of "cdmi_security_data_integrity" specified in Table 102 of 12.1.1 shall be set to "true". Otherwise, it shall not be present, indicating that there is no hashing support."

with

"This metadata shall be set by the client to indicate that the server shall hash the object value using the hashing algorithm and length specified. The result of the hash shall be provided in the cdmi_hash storage system metadata item. If absent or when an algorithm or length that is not supported by the server is specified, the cdmi_hash storage system metadata item shall not be provided.

Supported algorithm/length values are provided by the cdmi_value_hash capability."

TWG: Updated spec text approved.

CHANGES All specified changes made in Version 1.0.1m.

Trac Ticket #512 Table 20 `cdmi_encryption` examples should be in Table 12

The list and format of encryption modes supported should be documented in Table 12, not Table 20.

Proposed spec changes:

- 1 Table 104, "`cdmi_encryption`", replace the definition with:

If present, this capability shall list the encryption algorithms and key lengths supported. If absent, objects shall not be encrypted. When a CDMI implementation supports at-rest encryption, the system-wide capability of "`cdmi_security_encryption`" specified in Table 102 of Section 12.1.1 shall be set to "true". Otherwise, it shall not be present, indicating that there is no support for at-rest encryption.

Supported encryption algorithms are in the form of `ALGORITHM_MODE_KEYLENGTH`, where:

- "ALGORITHM" is the encryption algorithm (e.g., "AES" or "3DES").
- "MODE" is the mode of operation (e.g., "XTS", "CBC", or "CTR").
- "KEYLENGTH" is the key size (e.g., "128", "192", "256").

To improve interoperability between CDMI implementations, the following designators should be used for the more common encryption combinations:

- "3DES_ECB_168" for the three-key Triple DES algorithm, the Electronic Code Book (ECB) mode of operation, and a key size of 168 bits.
- "3DES_CBC_168" for the three-key Triple DES algorithm, the Cipher Block Chaining (CBC) mode of operation, and a key size of 168 bits.
- "AES_CBC_128" for the AES algorithm, the CBC mode of operation, and a key size of 128 bits.
- "AES_CBC_256" for the AES algorithm, the CBC mode of operation, and a key size of 256 bits.
- "AES_XTS_128" for the AES algorithm, the XTS mode of operation, and a key size of 128 bits.
- "AES_XTS_256" for the AES algorithm, the XTS mode of operation, and a key size of 256 bits.

- 2 Table 20:

This metadata shall be set by the client to indicate that the server shall encrypt the object using the encryption algorithm and key length specified. All data related to the data item/container shall be encrypted when this value is set, including metadata.

Supported algorithm/length values are provided by the `cdmi_encryption` capability.

CHANGES All specified changes made in Version 1.0.1m.

Trac Ticket #516 Clarify if mimetype field is case-insensitive

Description: Section 3.7 Media Types of the HTTP/1.1 RFC (<http://www.ietf.org/rfc/rfc2616.txt>) states that mimetypes as used in HTTP Content-Type/Accept headers are case-insensitive.

It is not clear if the mimetype field of a CDMI data object is also intended to be case-insensitive and should be treated as such in queries.

Given that for a non-CDMI data object create the mimetype field is populated from the HTTP Content-Type header (which is case-insensitive), one would expect that the mimetype field of the data object should also be treated as case-insensitive.

Another possibility would be to canonicalize the Content-Type (e.g., to lower case) when populating the mimetype field for a non-CDMI data object create.

Proposed spec text changes:

- 1 In section 8.2.5 Request Message Body, Table 6, metadata row, add the following line to the end of the description:
 - This mimetype value shall be converted to lower case before being stored.
- 2 In section 8.6.4 Request Message Body, Table 20, metadata row, add the following line to the end of the description:
 - This mimetype value shall be converted to lower case before being stored.
- 3 In section 9.9.5 Request Message Body, Table 50, metadata row, add the following line to the end of the description:
 - This mimetype value shall be converted to lower case before being stored.
- 4 In section 9.10.3 Request Header, Table 54, Content-Type row, in the description, replace:

The value specified here shall be used as the "mimetype" field of the CDMI data object.

with

The value specified here shall be converted to lower case and stored in the "mimetype" field of the CDMI data object.
- 5 In section 11.6.4 Request Message Body, Table 98, metadata row, add the following line to the end of the description:
 - This mimetype value shall be converted to lower case before being stored.

CHANGES All specified changes made in Version 1.0.1n.

Trac Ticket #519 Add parentID field

Description: In each response, we have an ObjectURI and ObjectID field returned to a client. However, while we return a parentURI, we aren't returning a parentID. Thus, there isn't a place where a CDMI client can consistently find the parentID (which are provided by systems that only support IDs, but not currently provided by systems that support paths).

Proposed spec change, including some cleanup of the objectName and parentURI fields:

Change existing instances of objectName and parentURI in response body tables, and add new "parentID" row below the parentURI row.

objectName

Name of the object.

- If containers are supported, the objectName shall be set to the last part of the path of the object, as defined by the right-most "/" to the end of the path.
- If containers are not supported, the objectName shall be set to the Object ID of the object.

parentURI

URI for the parent object.

- If containers are supported, the parentURI shall either be set to the URI path of the parent container or set to the URI path by ID of the parent container.
- If containers are not supported, the parentURI shall be set to "/cdmi_objectid/".

parentID

Object ID of the parent container object.

- If containers are supported, the parentID shall be set to the object ID of the parent container.
- If containers are not supported, parentID shall be set to the object ID of the container "/cdmi_objectid/".

TWG: Need to define "URI path by ID". This is "/cdmi_objectID/<nnnnn>/" where <nnnnn> is the object ID of the container.

No object to making this change.

TWG: Change approved.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #520 Remove extra " from parentURI description

There is an extra " at end of parentURI description.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #521 Unclear description of "capabilities" in 12.2 response body

Description: The "capabilities" row in the response body in section 12.2 is misspelled, and does not clearly represent the format of the response.

Proposed spec changes:

- 1 Fix misspelling of "capabilities"
- 2 Replace

"A tag list of capabilities supported by the corresponding object. Capabilities in the "cdmi_capabilities" object are system-wide capabilities. Capabilities found in children objects correspond to the capabilities of a specific subset of objects."

with

"The capabilities supported by the corresponding object. Capabilities in the "/cdmi_capabilities/" object are system-wide capabilities. Capabilities found in children objects under "/cdmi_capabilities/" correspond to the capabilities of a specific subset of objects. Each capability is expressed as a JSON string."

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #523 Sections missing 404 response status

Description: The 404 response status should be added to sections 8.2, 8.3, 8.7, 9.2, 9.11, 10.2, and 11.2.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #524 ctime, atime & mtime refer to "data item"

Description: The cdmi_ctime, cdmi_atime and cdmi_mtime storage system metadata descriptions refer to a "data item", when the rest of the storage system metadata descriptions refer to "object". Is this a relic, and this should be "object"? If not, what is meant by "data item"?

TWG: There is a desire to be consistent with what is done with NFS/CIFS. In these protocols, the atime/mtime should only be updated when the value is updated. But what about the other fields? What about user metadata?

The use of "data item" should also be fixed in Table 20 and Table 21 in Data System Metadata.

Replace "Data Item" with "Object".

TWG: Change approved.

In Table 115, rows for cdmi_ctime, cdmi_atime and cdmi_mtime, replace "data item" with "object".

CHANGES All specified changes made in Version 1.0.1m.

Trac Ticket #530 Incorrect CDMI header in Non-CDMI DELETE methods

Description: The X-CDMI-Specification-Version HTTP header should not be present in the Non-CDMI DELETE methods in section 8.9 and 9.8.

Proposed spec change:

In the request headers section of section 8.9 and 9.8, replace the table with "None specified."

Change approved by TWG.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #531 Typo in DELETE 409 response status description

Description: Replace "...or he data..." with "...or the data..." in sections 8.8 and 8.9.

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #532 Typo in "deserialize value" field name in 8.6 Update a Data Object

Description: In Section 8.6 Update a Data Object (CDMI Content Type), in the Request Message Body table:

* the field name "deserialize value" should be "deserializevalue".

CHANGES All specified changes made in Version 1.0.1j.

Trac Ticket #533 Editorial Changes #3

Description: Per Ralph's markups in ENDLTxReviewofCDMI[1-8].pdf:

All: Remove square brackets from all references, i.e., (see RFC 4627).

3-Terms

- 1 Remove note about SNIA dictionary.
- 2 Change the definition of ACL to "...principles (users and groups) to access resources.
- 3 Change the definition of cloud storage to "delivery of virtualized storage and data services on demand over a network (i.e., a non-technical term for DaaS)"
- 4 Change format of all explanatory text to notes (cloud storage, domain, IaaS, object, PaaS, private cloud, Representational State Transfer).
- 5 Add RFC reference for iSCSI, i.e., "Internet Small Computer Systems Interface (see RFC 3720)".

- 6 Add RFC reference for LUN, MIME, and NFS. For LUN, I recommend SAM-4: "ISO/IEC 14776-414, SCSI Architecture Model - 4 (SAM-4)" as an addition to clause 2 as well as here. MIME and NFS already have clause 2 entries (RFC 2045 and RFC 3530, respectively). Recommend the same reference format as for iSCSI.
- 7 Remove recursive definition of OID. Both OID and URI have definitions. Therefore, a better rendering of the definition would be: "an entity that has an OID, a unique URI, and contains state".
- 8 Change the note after private cloud to "Private clouds are created to address security issues."
- 9 Change definition of public cloud to ""delivery of SaaS, PaaS, IaaS, and/or DaaS to a unrestricted set of customers as compared to a private cloud".
- 10 In note that follows Representational State Transfer, replace "Architectures that follow these principles...." with "Architectures that follow Representational State Transfer principles...".
- 11 Add REST as an acronym to 3.18 Representational State Transfer.
- 12 Add normative reference to XAM and reference it here. If not a normative reference, add an informative reference to the bibliography annex.
- 13 Add terms mentioned throughout document. (These will be added in subsequent trac tickets.)

CHANGES All specified changes made in Version 1.0.1j. Ready for TWG to review.

Trac Ticket #534 Editorial Changes - 3-Terms, 4-Conventions, & Misc. Editorial changes

Description: Make the following editorial changes:

- 1 Clause 4.1 Interface Format - Replace "Section" with "Title".
- 2 Clause 4.2 Typographical Conventions - Remove this section and clear all extraneous formatting throughout entire document.
- 3 Clause 5 - Terms used inconsistently - servers/clients/applications and objects/entities/elements/data elements - will split out as separate trac ticket to be addressed by TWG (all instances of these words are to be reviewed).
- 4 Throughout document - change "this standard" to "this international standard".

CHANGES All specified changes made in Version 1.0.1j, except replaced "title" with "component".

Trac Ticket #535 Replace incorrect use of URL with URI

Description: There are a number of places in the spec where we use "URL" instead of "URI", mostly with respect to the location header. I believe it is safe to replace all instances of URL with URI.

TWG: Change approved.

Marie can do a global search and replace to change all instances of "URL" to "URI".

CHANGES All specified changes made in Version 1.0.1m.

Trac Ticket #538 Specify RFC reference for hex encoding

Description: We should update the sections that refer to hex encoding to include a reference to the RFC where this encoding is defined:

Proposed spec changes:

- 1 In section 5.11, replace:

"The native format for an Object ID is binary. When necessary, such as when included in JSON strings, the Object ID textual representation shall be hex-encoded and case insensitive."

with

"The native format for an Object ID is binary. When necessary, such as when included in URIs and JSON strings, the Object ID textual representation shall be encoded using Base16 encoding rules described in [RFC4648], and shall be case insensitive."
- 2 In section 16.3, Table 19, row "cdmi_hash", replace:

"The hash of the value of the object, hex-encoded as a string. This metadata field shall be present when the "cdmi_value_hash" data system metadata for the object or a parent object indicates that the value of the object should be hashed."

with

"The hash of the value of the object, ncoded using Base16 encoding rules described in [RFC4648]. This metadata field shall be present when the "cdmi_value_hash" data system metadata for the object or a parent object indicates that the value of the object should be hashed."

CHANGES All specified changes made in Version 1.0.1j; also added RFC 4648 to Clause 2 Normative References.

Trac Ticket #539 Clause 5 changes - Pages 42-47 of ENDLTxReviewOfCDMI[1to8].pdf

Description: Per Ralph's comments in ENDLTxReviewOfCDMI[1to8].pdf.

Note: The following numbers correspond to the numbers in the PDF.

Page 42-43:

2. [translation concern] In the three green highlighted phrases, it is difficult to determine the relationship between text and the parenthetical expression. For example, do "data paths" further define "functional interfaces" (to whit "i.e.") or do "data paths" provide an example of "functional interfaces" (to whit "e.g."). It would be a good idea to determine which is which and modify the parenthetical expression accordingly. (Will separate into another ticket.)

3. Change "this standard" to "this international standard. (Completed in Trac Ticket #534)
4. Delete "rich" from last paragraph of clause 5.1.

Page 44-45:

1. In 5.2, change "dominant networking icon" to "icon for a network".
2. In 5.2, change

"The cloud conceptually represents any-to-any connectivity in a network. The cloud also conceptually represents an abstraction of concerns, such that the actual connectivity and the services running in the network that accomplish that connectivity do so with little manual intervention."

to

"The cloud represents any-to-any network connectivity in an abstract way. In this abstraction, the network connectivity in the cloud is represented without concern for how it is made to happen."
3. In 5.2, change

"This abstraction of complexity and promotion of simplicity is what primarily constitutes a cloud of resources, regardless of type."

to

"The cloud abstraction of complexity produces a simple base upon which other features can be built. The general cloud model extends this base by adding a pool of resources, regardless of type."
- 4, 5, 6. Delete "in general" from paragraph 2 of 5.2 and rewrite last two sentences of 2nd paragraph 5.2, eliminating "you" and unqualified parenthetical as follows:

"An important part of the cloud model is the concept of a pool of resources that is drawn from small increments. A relatively recent innovation that has made this possible is virtualization."
7. Eliminate definition of DaaS.
 1. Change

"...delivered, not based on demand, but on fixed capacity increments."

to

"...delivered in fixed-capacity increments instead of based on demand."

Page 46-47:

2. Change reference to figures to look exactly like references to Tables (number only).
3. Remove "merely" and "in other cases" from first paragraph following Figure 1.
4. Change "of" to "for" in 2nd paragraph following Figure 1.

5. Change "this interface" to "this out-of-band interface" in last sentence of para 2. following figure 1.
6. Change "...services, such as snapshot and cloning." to ..."services (e.g., snapshot and cloning).
7. Replace "In this model, the underlying storage space is abstracted and exposed by these interfaces using..." with "In the model defined by this international standard, the underlying storage space exposed by the out-of-band interfaces is abstracted and exposed using..."
8. Replace
 "...data storage interface of limited functionality, with the emphasis on scalability rather than features."
 with
 "...data storage interface that emphasizes scalability while placing known limits on functionality."
9. Change "nodes" to "database nodes" or "database-like nodes".
10. Change "...values, affording..." to values. This model provides...

CHANGES All specified changes made in Version 1.0.1j, except where new trac tickets were created. Ready for TWG to review.

Trac Ticket #542 **Clause 5.1 - 1st para - needs clarification per Ralph's comments**

Description: Per Ralph's comments in ENDLTxReviewOfCDMI[1to8].pdf. Note: The following numbers correspond to the numbers in the PDF.

Page 42-43: (referring to 1st paragraph of 5.1):

2. [translation concern] In the three green highlighted phrases, it is difficult to determine the relationship between text and the parenthetical expression. For example, do "data paths" further define "functional interfaces" (to what "i.e.") or do "data paths" provide an example of "functional interfaces" (to what "e.g."). It would be a good idea to determine which is which and modify the parenthetical expression accordingly.

Change:

These resources are exposed to clients as functional interfaces (data paths) and are managed by management interfaces (control paths).

To:

These resources are exposed to clients as functional interfaces (i.e. data paths) and are managed by management interfaces (i.e. control paths).

TWG: Approved.

CHANGES All specified changes made in Version 1.0.1m.

Trac Ticket #543 **Clause 5 changes: 5.3, 5.4, 5.5, and 5.6 - Pages 48-64 of ENDLTxReviewOfCDMI[1to8].pdf**

Description: Per Ralph's comments in ENDLTxReviewOfCDMI[1to8].pdf.

Note: The following numbers correspond to the numbers in the PDF.

5.3 pp. 48-51:

11. Change "these interfaces" to "table space storage interfaces".
12. Change "proprietary" to "vendor-specific".
13. Make sure references to figures are the same as references to tables (see 6.6.7.4 starting on page 35 in the ISO/IEC directives). (Fixed in trac ticket # 539.)
14. Delete two paragraphs after Figure 2 (use FM conditional text to include text in CDMIv2).
15. Change "...is created, retrieved, updated, and deleted (CRUD semantics)" to "...is processed using CRUD semantics as ...".
16. Delete "in this case".
17. Keep paragraph preceding Figure 3 on same page (set orphan rows to 3).
18. Delete "of such".
19. Change "This type of container" to "The type of container defined in this international standard".
20. Delete paragraph after Figure 3 (just before 5.4).

5.4 pp. 52-57:

1. Change 5.4 "Data Management in the Cloud" to "Data Management for Cloud Storage".
2. Change "...with very little offering of additional data services for that data" to "... and ignored most other types of data services".
3. Change "pressure" to "need".
4. Change "better quality of service" to "better quality of service features".
5. Change 2nd paragraph of 5.4 as follows:

Cloud storage may lose its complexity-abstraction and simplicity benefits if new data services that require enhanced management are added. Cloud storage customers are likely to resist new demands on their time (e.g., setting up backup

schedules through dedicated interfaces, deploying data services individually for data elements).

7. Change "gives us" to "provides".
8. Delete "minimize this complexity and".
9. Change "that model" to "the SIRDM model".
10. Change "created" to "specified".
11. Change "undue" to "unnecessary".
12. Update Figure 4, per Ralph's comments: "If "Cloud Data Storage Interface" is a layer in figure 4, then the font size should match that of the other layered elements, and maybe a box could be drawn around it or arrows drawn from it. If "Cloud Data Storage Interface" is a figure title, then it should be removed, since figure titles must appear below the figure (see 6.6.5.4 on page 29 in the ISO/IEC Directives)." (See [Trac Ticket #545](#).)
13. Change Figure 4 title to "Cloud Storage Usage of SIRDM".
14. In 1st paragraph following Figure 4, remove "standard".
15. Change "the cloud storage paradigm" to "the cloud storage model" (or just "cloud storage").
16. Change "maintained and may still address" to "maintained while still addressing".
17. Change "doing" to "performing".
18. Change "each application, domain, or the user" to "each application, domain, or user".
19. Change all instances of "chapter" to "clause" and remove all references to "section".
20. Change the 1st sentence in paragraph 3 following Figure 4 as follows: Change: "Storage system metadata is produced and interpreted by the cloud offering or basic storage functions, such as modification and access statistics, and for governing access control." to "Storage system metadata is produced/interpreted by the cloud offering and basic storage functions (e.g., modification and access statistics, access control)."
- 21, 22. In paragraph 4 following figure 4, change "drive" to "control" and delete "even".

5.5 pp 58-61:

1. Change "paradigms" to "interfaces."
2. Change "this standard proposes" to "this international standard specifies"
3. Delete "essentially".
4. Change "would similarly inherit" to "inherits".

5. Change "Of course" to "After a data element is created," (Note: make sure "data element" is correct term after reviewing naming rules suggested in comment 1 on clause 5).
6. In 5.5, replace "Even if the functional interface that the offering provides does not support setting this type of metadata on individual data elements, it may still be applied to the containers, even though it may not be able to be overridden on the basis of individual data elements through that interface." with "Even if the provided interface does not support setting metadata on individual data elements, metadata may still be applied to the containers. In such a case, the interface does not provide a mechanism to override metadata that an individual data element inherits from its parent container."
7. Are CIFS and NFSv4 the only file-based interfaces that support extended attributes? If yes, then the "i.e." is correct. If no, then the "i.e." should be changed to "e.g.". (See Trac Ticket #546.)
8. Delete "through these existing standards".
9. Delete the last paragraph of 5.5 that begins with "The mapping ..."

5.6 pp. 61-64:

1. Change "By putting all of these elements together, a reference model is created, as shown in Figure 5, "Cloud Storage Reference Model":." to "The Cloud Storage Reference Model is shown in figure 5."
2. In Figure 5, change "Clients can be in the cloud or enterprise and provide additional services (computing, data, etc.) " to "Clients can be inside the storage cloud (i.e., providing storage resources to the cloud as well as consuming them) or outside the storage cloud (i.e., only consuming resources)."
3. In Figure 5, change "Proprietary" to "Vendor-Specific".
4. In Figure 5, change "overall management of your cloud computing" to "overall cloud computing management".
5. In Figure 5, move the text box "SNIA Cloud Data Management Interface" so that it is not obscured by "Cloud Data Management".
6. Replace "capacity" with "storage capacity (i.e., the storage being provided on demand)".

CHANGES All specified changes made in Version 1.0.1j, except where new trac tickets were created. Ready for TWG to review.

Trac Ticket #544 Clause 5 changes: 5.7 thru 5.13 - Pages 65-83 of ENDLTxReviewOfCDMI[1to8].pdf

Description: Per Ralph's comments in ENDLTxReviewOfCDMI[1to8].pdf.

Note: The following numbers correspond to the numbers in the PDF.

5.7 pp 64-65

1. Delete two instances of "SNIA".
2. Change first sentence of paragraph 1 to "The Cloud Data Management Interface (CDMI) shown in figure 5 may be used to create, retrieve, update, and delete data elements in a cloud."
3. Change 2nd sentence of paragraph 1 to: "The features of the CDMI include functions that:
 - allow clients to discover the capabilities available in the cloud storage offering,
 - manage containers and the data that is placed in them, and
 - allow metadata to be associated with containers and the data element they contain."
4. Delete 2nd paragraph of 5.7.
5. Change "This interface" to "CDMI".
6. Change "offer" to "support".
7. Change "CDMI interface" to "CDMI".
8. Change "part of the interface" to "report via the interface".
9. Change [REST] to "(see REST)".
10. Delete 2nd sentence in last paragraph beginning with "Typically".

5.8 pp 65-71

11. Change "behind" to "for".
 1. The most recent time data paths were mentioned was in 5.1, where they are inconclusively related to "functional interfaces". The only other time data paths are mentioned in clause 5 is two paragraphs below, where the statement amounts to freeing CDMI from implementing data paths. These are mighty slim pickings upon which to lay requirements. Is there any chance that the cited text really means to say: "All CDMI data path implementations are required? (create new ticket for TWG).
 2. Change "containers, a sort of grouping of data objects. " to "...containers (see 5.5)."
 3. Change "As shown in Figure 6, the client may do a PUT" to "Using the CDMI object model (see figure 6), the client may send a PUT via the CDMI (see 5.6)..."
 4. Change "KEY/VALUE" to "Key Value" throughout spec.
 5. Change "Once a container is created," to "After a container is created,"

6. If MIME is an example of content type, then change "content type (MIME)" to "content type (e.g., MIME)". If there is some other relationship between MIME and content type, then take steps to explain this fully or eliminate the parenthetical expression.
7. Change "the CDMI standard" to "this international standard".
8. Change 1st sentence of paragraph 3 beginning with "CDMI also defines..." to "This international standard also defines a queue object (see figure 6), that has special properties for in-order, first in, first-out creation and fetching of queue objects. The relationship between queue objects and a container is similar to the relationship between data objects and a container."
9. After correcting last three paragraphs of 5.8, move to 5.7.
10. When read in conjunction with the opening part of this sentence - "data path, and it [i.e., CDMI] applies to cloud storage that is exposed as either standard or proprietary interfaces" - seems to say that can manage over proprietary interfaces. This is not the kind of thing a standard does. Is the real intent that CDMI can manage cloud storage properties without regard for the data path interface (e.g., standard or vendor specific) being used? If the answer is yes, then consider changing the cited text to: "data path. The CDMI is able to manage cloud storage properties for any data path interface (e.g., standardized or vendor specific)."
11. If the guess about intent in comment 10 (immediately above) is correct, then change the next sentence (sentence 2 of paragraph 3) as follows: "When the CDMI is used only for cloud storage management, the abilities of the CDMI client may be limited to container creation."
12. New paragraph beginning with "The metadata is also..."
13. Rewrite this sentence as follows: "The metadata is also used to configure the data requirements of the storage under the exported protocol (e.g., block protocol or file protocol) that the container exposes."
14. Replace 2 sentences beginning with "While many implementations..." with "When an implementation is based on an underlying file to store data for a block protocol (e.g., iSCSI), the CDMI container provides a useful abstraction for representing the data system metadata for the underlying data and the structures that govern the exported protocols. One example of this implementation is a block storage device (Target, LUN) that is represented as a CDMI container filled with block-sized data objects that are identified by their Logical Block Address."
15. Change ", which" to "that".
16. Paragraph 4 - Change second sentence to a list: "Domains provide a mechanism to:
 - determine how user credentials are mapped to principles used in an Access-Control List (ACL),
 - allow the granting of special cloud-related privileges, and

- allow delegation to external user authorization systems (e.g., LDAP or Active Directory)."

17. Change "The interface" to "CDMI" or "This international standard".

5.9 pp 72-75

1. Change "the HTTP protocol, such as content-size, content-type, etc." to "the HTTP protocol (e.g., content-size, content-type)".
2. Change "This type of metadata" to "HTTP metadata".
3. Change "the CDMI standard" to "this international standard".
4. Change "attached to" to "associated with".
5. Change "data object, abstractly specifying" to "data object. Data system metadata abstractly specifies".
6. Change "requirements that are then supplied by data services that" to "requirements associated with data services that".
7. Change "objects" to "data objects" or "data or container objects".
8. Change "self-administered (such as using the reverse domain name)" to "self-administered (e.g., using the reverse domain name)".
9. Change "Storage system metadata is read-only metadata that is generated by the storage services in the system to provide useful information to a CDMI client. Examples include ACLs, creation time, etc." to "Storage system metadata is read-only metadata that is generated by the storage services in the system (e.g., ACLs, creation time) to provide useful information to a CDMI client."
10. Add sentence just before Table 4 to introduce it.

5.10 pp 74-75

1. Change "will have" to "shall have".
2. Change "string but has rules" to "string with requirements".
3. Change "Every CDMI system shall allow access to stored objects, by object ID, by allowing the object ID to be appended to the root container URI." to "Every CDMI system shall allow OID-based access to stored objects by allowing the object's OID to be appended to the root container URI."
- 4,5. Use ISO requirements for all examples throughout document. Make examples standalone paragraphs, changing this sentence as follows:

EXAMPLE - For the root container URI shown on the first line, the URI on the second line provides access to objects by object ID.
6. If Table 4 (see 4.2 in the working draft) is to be believed, then this should be in italics. (TWG agreed to remove all of this formatting.)

5.11 pp 76-79

1. Change "As shown in Figure 7," to "The fields shown in figure 7 are defined as follows."
2. Format all lists according to ISO specifications: (see 5.2.5 on page 16 in the ISO/IEC Directives), and use less white space between bullets in the 2nd bulleted list.
3. Make last bullet a body text paragraph (no bullet).

5.12 pp 78-81

1. Format list per ISO.
2. Change para 2 (after bulleted list) to a bulleted list per ISO requirements.
3. Changing "security capability queries to determine" to "security capability queries (see 12.1.1) to determine" and removing the sentence that cross references 12.1.1.3.
4. Change "responses from the CDMI implementation" to "responses to these queries".
5. The current phrasing applies an ISO/IEC keyword "shall" (see Annex H starting on page 69 in the ISO/IEC Directives) to a behavior that is not mandated by this international standard. Suggest changing "regulated, such that it shall be protected" to "regulated in a way that requires the data to be protected".
6. Change "for example" to "e.g." (and check for other instances throughout document).
7. Change "this version of CDMI" should be changed to "this international standard".
9. Change "CDMI entities" to "CDMI clients and servers" two times in this paragraph.
10. Change "are required to implement" to "shall implement".
11. Delete 2nd reference to Annex A.
12. Remove last paragraph.

5.13 pp 82-84

1. Remove "hanging paragraph" by adding subclause before first paragraph - "5.13.1 RFC 2616 Support Requirements".
2. Change "an Accept header" to "an HTTP Accept header".
3. Remove parentheses from 2nd sentence of first paragraph of 5.13.1 Content-Type Negotiation.
4. Change "Range headers" to "HTTP Range headers".
5. Change "All date/time values, unless otherwise specified, are" to "Unless otherwise specified, all date/time values shall be".

6. Change "must" to "shall" (four times) in sentence beginning "The full precision must". Change each item in the series to a sentence (four sentences).
7. Change "must" to "shall" on two instances on this page that are not otherwise modified.
8. Change "All time durations, unless otherwise specified, are" to "Unless otherwise specified, all time durations shall be".
9. Change "must" to "shall" (four times) in sentence beginning "The end-date must". Change each item in the series to a sentence (four sentences).

CHANGES All specified changes made in Version 1.0.1j, except where new trac tickets were created. Ready for TWG to review.

Trac Ticket #546 Clarification of CIFS and NFSv4 - Subclause 5.5

Description: (Per Ralph's comments) In 2nd sentence of 2nd paragraph of 5.5: Are CIFS and NFSv4 the only file-based interfaces that support extended attributes? If yes, then the "i.e." is correct. If no, then the "i.e." should be changed to "e.g."

There are other file-based interface that can support extended attributes. Thus, this should be changed to "e.g."

CHANGES: Made the specified change.

Trac Ticket #547 Clarification of data paths in 5.8

Description: Per Ralph's Comments, which refer to 1st paragraph, 2nd sentence, beginning with "All data path implementations..."

"The most recent time data paths were mentioned was in 5.1, where they are inconclusively related to "functional interfaces". The only other time data paths are mentioned in clause 5 is two paragraphs below, where the statement amounts to freeing CDMI from implementing data paths. These are mighty slim pickings upon which to lay requirements. Is there any chance that the cited text really means to say: "All CDMI data path implementations are required?"

Reference is to subclause 5.8, first paragraph following Figure 6, second sentence.

In Section 5.7, change:

This international standard uses RESTful principles in the interface design where possible (see REST).

CDMI does not need to be used as the data path and is able to manage cloud storage properties for any data path interface (e.g., standardized or vendor specific).

To:

This international standard uses RESTful principles in the interface design where possible (see REST).

CDMI defines both a means to manage the data as well as a means to store and retrieve the data. The means by which the storage and retrieval of data is achieved is termed a "data path". The means by which the data is managed is termed the "control path". CDMI specifies both a data path and control path interface.

CDMI does not need to be used as the only data path and is able to manage cloud storage properties for any data path interface (e.g., standardized or vendor specific).

TWG: Approved.

CHANGES: Made the specified change to Version 1.0.1m.

Trac Ticket #549 Clause 6 and Clause 7 changes: Pages 86-91 of ENDLTxReviewOfCDMI[1to8].pdf

Description: Per Ralph's comments in ENDLTxReviewOfCDMI[1to8].pdf.

Note: The following numbers correspond to the numbers in the PDF.

Clause 6:

1. Remove gray background from code examples and introduce each code example with a sentence, i.e., "The following shows the response." Change all examples to conform to ISO/IEC-required example format (see the examples in 6.5.1 on page 25 in the ISO/IEC Directives.)
2. Fix hanging paragraph by adding "Overview" as follows:

6.1 Overview

This clause illustrates the CDMI operations

- discovering the capabilities of a cloud storage provider (see 6.2),
- creating a new container (see 6.3),
- creating a new data object (see 6.4),
- listing the contents of a container (see 6.5),
- reading the contents of a data object (see 6.6),
- reading only the value of a data object (see 6.7), and
- deleting a data object (see 6.8).

Clause 7:

1. Change first sentence in 7.1 to "HTTP status codes (see Table 5) are used to convey the results of RESTful operations and to follow the basic semantics of HTTP with minimal overloading."
2. Change "Other status codes" to "Other HTTP status codes".

3. Change Not Implemented definition to "An unimplemented CDMI operation or metadata access was attempted."
4. Change "resource objects" to "HTTP resource objects."
5. In 7.2, change introductory sentence to "The types of HTTP resource objects defined by this international standard are shown in table 6."
6. Add third column to Table 6 with a title of Reference. Put reference in third column.
7. Change "varies by" to "depends on".
8. Change "...system, and the..." be changed to "...system. The..."
9. Change "may not" to "shall not".
10. Format all examples per ISO standards.

CHANGES All specified changes made in Version 1.0.1j, except where new trac tickets were created. Ready for TWG to review.

Trac Ticket #551 Item 2 of 8.2 of ENDLTxReviewOfCDMI[1to8].pdf

Description:

2. Fix subclause headers to conform to ISO/IEC. 3 options:
- Change all "section" headers to subclause headers; add titles to all tables; and add introductory sentence to tables.
 - Change all "section" headers to subclause headers; add "TOC" to the Synopsis section and introduce each subclause; add titles to all tables; and add introductory sentence to tables.
 - Change all "sections" to tables; add "TOC" to the Synopsis section that introduces tables; add introductory sentence to all tables.

(Marie recommends option 1).

CHANGES All specified changes made in Version 1.0.1j. Ready for TWG to review.

Trac Ticket #554 Clause 8.1 and 8.2 changes: pp. 93-123 of ENDLTxReviewOfCDMI[1to8].pdf

Description:

Note: Note: All referenced numbers correspond to numbers in PDF.

8.1

1. Change "filesystem" to file system".

2. Rewrite sentence 2, para 1: "Each data object has a set of well-defined fields that include:
 - single value; and
 - optional metadata that is generated by the cloud storage system and or specified by the cloud user."
3. This appears to be a data object that is not resident within a container. Figure 6 does not show data objects that are not in containers. No other data object URI that I can remember reading skips the container, except those on this page. If a data object is not required to be in a container, this fact needs to be made clear in clause 5. The first mention of this should not be hidden in an "example". (See [ticket #550](#)).
4. Remove 1st sentence of paragraph 3 (after bullets), and recast bulleted list (para 2) as follows:

Data objects are addressed by CDMI in one of the following ways:

 - by URI (e.g., `http://cloud.example.com/container/dataobject`); and
 - by OID (e.g., `http://cloud.example.com/cdmi_objectid/0000706D0010B84FAD185C425D8B537E`).
5. Change "For example, the "budget.xls" data object stored at "http://cloud.example.com/finance/budget.xls" would inherit data system metadata from its parent container, "finance". " to "Thus, the "budget.xls" data object at the following URI inherits data system metadata from the parent container "finance"."
6. Change "For example, the following URI would return just the value field in the response body: " to "Thus, the following URI returns just the value field in the response body."
7. Change "For example, the following URI would return the first thousand bytes of the value field: " to "Thus, the following URI returns the first 1000 bytes of the value field."
8. Change "For example, the following URI would return the value and metadata fields in the response body: " to "Thus, the following URI returns the value and metadata fields in the response body."

8.1.2

1. Change "Writing to a data object is an atomic operation. If a client were to read an object simultaneously with a write to that same object, it shall get either the old version or the new version, but not a mix of both. Writes are also atomic in the face of errors." To ""Writing to a data object is an atomic operation.
 - If a client reads a data object simultaneously with a write to that same data object, the reading client shall get either the old version or the new version, but not a mixture of both.
 - If a write is terminated due to errors, the contents of the data object shall be as if the write never occurred (i.e., writes are atomic in the face of errors)."

2. Change “Multiple simultaneous writes that complete without errors shall be ordered by the timestamps on the returning responses, which is to say, by the timestamps placed on the responses by the server-side implementation (i.e., according to the principle of eventual consistency).” to “The timestamp returned in the response to a write indicates whether the write is the newest (i.e., the write whose data is returned to subsequent reads, until another write is processed). If the timestamp returned for one write shows a more recent time than the timestamp for another write, then the write with the new timestamp shall be the one that whose data is currently stored in the data object wherever the two writes overlap.”

(I did not copy the phrase “according to the principle of eventual consistency” into the rewrite because I have never heard of this principle. If this phrase is important, it needs to be defined via a Bibliographic reference or in the Terms clause.)

3. Change “that has not previously been written to” to “to which data has not been previously written.”
4. Change “the byte value” to “the value”.
5. Change the note text to “Implementations of this international standard shall provide the atomicity features described in this subclause for data objects that are accessed via CDMI. The atomicity properties of data objects that are not accessed via CDMI are outside the scope of this international standard.”

8.1.3

6. Change “this section” to “clause 8”.
7. Move “Both clients and servers shall support JSON representation.” to the second sentence in paragraph.

8.2 - p. 101

1. Change all Synopsis sections in clause 8, 9, 10, 11, and 12 to conform to ISO/IEC. As an example, replace:

Synopsis:

Creates a new data object at the specified URI.

- <root URI> is the path to the CDMI cloud.
- <ContainerName?> is zero or more intermediate containers that already exist.
- <DataObjectName?> is the name specified for the data object to be created.

With:

The following HTTP PUT creates a new data object at the specified URI.

PUT <root URI>/<ContainerName?>/<DataObjectName?>

Where:

- <root URI> is the path to the CDMI cloud.
- <ContainerName?> is zero or more intermediate containers that already exist.
- <DataObjectName?> is the name specified for the data object to be created.

2. Fix subclause headers to conform to ISO/IEC. 3 options:
 - Change all “section” headers to subclause headers; add titles to all tables; and add introductory sentence to tables.
 - Change all “section” headers to subclause headers; add “TOC” to the Synopsis section and introduce each subclause; add titles to all tables; and add introductory sentence to tables.
 - Change all “sections” to tables; add “TOC” to the Synopsis section that introduces tables; add introductory sentence to all tables.(Marie recommends option 1). (See ticket #551)
3. Add title and an introductory sentence to all tables. Make sure tables that cross a page have “(part n of m) in the title of the table.
4. Make all description entries consistent (some are sentences, some are fragments, some are fragments followed by sentences). (See ticket #552).
5. Make all table footnotes lowercase letters.

8.2 - p. 111

1. Change “...Object (CDMI Content Type)” to “...Object with CDMI Content” (Change throughout document.)
2. Change “zero or more intermediate containers that already exist.” to “zero or more intermediate containers that already exist with one slash (i.e., “/”) between each pair the container names” (change throughout document).
3. Change “Once created, the object may also be accessed at” to “After it is created, the data object may shall be accessible at”. (Change throughout document.)
4. Change “On a create operation for a data object, the server may return a response of 202 Accepted. In this case, the object is in the process of being created. This response is particularly useful for long-running operations, for instance, copying a large data object from a source URI. Such a response has the following implications:” to “In response to a create operation for a data object, the server may return 202 Accepted to indicate that the object is in the process of being created. This response is useful for long-running operations (e.g., copying a large data object from a source URI). Such a response has the following implications.”
5. Change “The server returns” to “The server shall return”
6. Make sub-bulleted list conform to ISO standards (lower case separated by semicolons and last entry ending with a period).
7. Change “modulo” to “unless”; define eventual consistency in Terms.
8. Change “The client performs GET operations to the URI to track the progress of the operation. In particular, the server returns two fields in its response body to indicate progress:” to “The client performs GET operations to the URI to track the

progress of the create operation. In response, the server returns two fields in its response body to indicate progress."

8.2 - Page 115

1. Change "last PUT" to "Accepted PUT".
2. Change ""GET does not return any value for the object when completionStatus is not Complete. When the final result of the create operation is an error," " to "GET shall not return any value for the data object when completionStatus is not Complete. If the final result of the create operation is an error"
3. (See ticket #553.)
4. Change "When set, the completionStatus field shall be set to "Processing". to "If X-CDMI-Partial is present, the completionStatus field in the Response Body shall be set to "Processing"."

8.2 - p. 117

1. Fix font consistency errors.
2. Change "({})" to "(i.e., {})".
3. Change "then removing" to "and removed".
4. Change "operation, and except" to "operation. Except".

8.2 - p. 119

1. Change ("")" with (i.e., "").
2. Change "for example, "1.0." with "e.g., "1.0)."
3. N/A (already corrected.)
4. Fix list per ISO.
5. Remove extra quote.

8.2 - p. 121

1. In Response Message Body table, change completionStatus description to: "A string that shall indicate the status of the data object creation operation using one of the following values."
 - "Processing" indicates that the data object is still in the process of being created.
 - "Completed" indicates that the data object has been successfully created.
 - A string that begins with "Error" indicates that an error prevented creation of the data object."
2. Change "Investigate" to "The CDMI client should monitor".

3. Use fonts consistently for HTTP status values. (Already corrected.)

CHANGES All specified changes made in Version 1.0.1j, except where new trac tickets were created. Ready for TWG to review.

Trac Ticket #559 Remove extra space ("deserialize value" -> "deserializevalue")

Description: In section 8.6, Request Message Body, "deserialize value" field name should be "deserializevalue" (no space)

CHANGES Made the specified change.

Trac Ticket #561 Clause 8.3 thru 8.8 changes: pp. 123-141 of ENDLTxReviewOfCDMI[1to8].pdf

Description:

Note: All numbers in this ticket reference a number in the PDF.

8.3 - p. 123

1. Same as Item 1, p. 111. (see "8.2 - p. 111" on page 250.)
2. Same as Item 2, p. 111. (see "8.2 - p. 111" on page 250.)
3. Same as Item 3, p. 111. (see "8.2 - p. 111" on page 250.)
4. Same as Item 3, p. 115. Another possible solution would be to define "presence" as a keyword (assuming that a Keywords subclause is added to Clause 4). The catch is that, if "presence" is defined as a keyword, then the working draft must be scrubbed for uses of "presence" that do not fit the keyword definition. (see ticket #555).
5. Change "When set, the completionStatus field shall be set to "Processing"." to "If X-CDMI-Partial is present, the completionStatus field in the Response Body shall be set to "Processing"."

8.4 - p. 125

1. With regard to bullet 4 "<fieldname> is the name of a field", per Ralph's comments: "Based on the words in front of me, this looked like a way to address metadata. Apparently, this is a wrong guess, but it is hard to be sure. This is another first-rate opportunity to add a Common Fields subclause somewhere." (see ticket #556)
2. No change, just a comment.

8.4 - p. 127

1. Move these two sentences to Synopsis, after bulleted list (also in 8.5):

"Any read from a specific byte location not previously written to by a create or update operation shall return zero for the byte value."

and

"Any read from a specific byte location within the value range specified not previously written to by a create or update operation shall return zero for the byte value."

2. Create a cross-ref to the five CDMI MIME types.
3. To prevent a translator from seeing meaning in the presence or absence of the phrase "The server shall respond", arrange to use the same structure for all Response tables in the working draft. (See Ticket #558.)

8.4 - p. 129

1. Search and replace "will indicate" to "shall indicate" or "indicates". Also do a global search for "will" and replace with "shall", if appropriate.
2. A field that is "only provided" under certain circumstances cannot also be "Mandatory". This is not to say the field should be marked "Optional". The work-around I have seen frequently is to put a table footnote on the "Mandatory" and specify the conditions under which the field is mandatory in the table footnote. (see ticket #557)
3. In this row, the text says Complete. In the completionStatus row, "Complete" is used. Pick one and use it globally.

8.4 - p. 131

1. In the following sentences:

"If individual fields are specified in the GET request, only these fields are returned in the result body. Optional fields that are requested but do not exist are omitted from the result body."

These two look like a cases where a "shall" might be the TWG's intention. Of course, if a Keywords subclause is not added to clause 4, then this concern is irrelevant (but uniformity of implementation is likely to suffer too).

(See ticket #560).

p. 133

1. Change "results" to "shall result".
2. Change "codeof" to "code of".

8.6 - p. 135

1. Change "object" to "data object".
2. Change "The object may also be accessed at <root URI>/cdmi_objectid/<objectID>, and an update shall not result in a change to the Object ID." to "The object may also be accessed at <root URI>/cdmi_objectid/<objectID>. An update shall not result in a change to the Object ID. (This change will also make 8.6 more consistent with 8.4 and 8.5. These changes are also needed in 8.7.)"

8.6 - p. 137

1. Create cross-ref to the description of privileges.
2. Change "is inserted" to "shall be inserted."

8.8 - p. 139

1. Ensure all Response Status tables stay on the same page.

CHANGES Made the specified changes to Version 1.0.1j.

Trac Ticket #562 Missing Updates to Figure 15

Description: The diagram for Figure 15 seems to have lost the original formatting, and still refers to com.netapp.versioning, instead of the cdmi_versions as per the changes to the text. Original visio diagram attached.

CHANGES Corrected diagram and reimported it into Version 1.0.1j.

Trac Ticket #563 Incorrect capitalization in B.6.1

Description: "X-CDMI-PARTIAL" should be "X-CDMI-Partial".

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #567 Extra "?" in section 5.10 example data object name

Description: Extra "?" in the example data object name "MyDataObject?.txt" in section 5.10. The data object name should be "MyDataObject.txt".

In the same section, "MyContainer?" should not have a "?".

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #568 Normative Reference link for RFC 4648 is incorrect

Description: It should be "http://www.ietf.org/rfc/rfc4648.txt".

CHANGES Made the specified change to Version 1.0.1k.

Trac Ticket #569 Error in 9.4.5 text

Description: "The HTML response headers for reading a CDMI container object with non-CDMI content are shown in Table 37."

should be:

"The HTML response headers for reading a CDMI container object with CDMI content are shown in Table 37."

Same problem in the table 37 heading.

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #571 Copyright Notice

Description: Under copyright notice, determine contact information to go in title page of CDMI standard.

Use:

The Storage Networking Industry Association
425 Market Street, Suite 1020
San Francisco, CA 94105

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #572 "5.13.2 Range Support Header" Dropped

Description: The header "5.13.2 Range Support Header" has been dropped and should be added back in.

In 1.0.1j this would be "5.12.2 Range Support Header".

No change. 5.13.3 Range Support is in version 1.0.1j.

CHANGES None.

Trac Ticket #573 cdmi_owner storage system metadata listed as optional, but no capability

Description:

Section 16.3 Support for Storage System Metadata, Table 19 lists the storage system metadata item "cdmi_owner" as "Optional". However, unlike the rest of the storage system metadata items, there is no corresponding row for "cdmi_owner" in Section 12.1.2 Storage System Metadata Capabilities, Table 11.

- If "cdmi_owner" is optional, a corresponding capability should be added to Table 11.
- If "cdmi_owner" is not optional, Table 19 should be updated to indicate that it is mandatory.

TWG: Proceed with the change below.

The cdmi_owner row in Table 115 should be "Mandatory".

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #574 Remove references to HTML headers

Description: The spec erroneously refers to "HTML" request and response headers. These are HTTP headers.

- 1 Replace "HTML request header" with "HTTP request header"
- 2 Replace "HTML response header" with "HTTP response header"

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #576 Update examples to show 1.0.1 as CDMI version

Description: The examples in the 1.0.1 draft spec should be updated so that the X-CDMI-Specification-Version in request and response headers is "1.0.1" instead of "1.0".

CHANGES Made the specified changes throughout the document.

Trac Ticket #577 This mimetype value shall be converted to lower case before being stored.

Description: Need to add spec text that indicates what happens if a client does not provide a CDMI Specification Version in the request header that is supported by the server.

Recommendation is to return a 206 result code.

206 does not make sense; 501 Not Implemented makes more sense.

TWG: Decision to return a 501 in this situation.

501 isn't permitted, because this is only valid when an HTTP method is not supported.

Proposed spec text changes:

Throughout the document, replace:

"The server shall respond with the highest version supported by both the client and the server, e.g., "1.0.1".

with

"The server shall respond with the highest version supported by both the client and the server, e.g., "1.0.1".

If the server does not support any of the versions supported by the client, the server shall return a 400 Bad Request status code."

CHANGES Made the specified changes throughout the document to Version 1.0.1n.

Trac Ticket #579 Change Figure 6 to "name-value"

Description: Figure 6 should read "name-value" instead of "key value" to be consistent with JSON.

(the text below also needs to be updated to refer to "name-value" instead of "key-value").

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #580 Remove "OID", instead use "object ID"

Description: Remove use of the acronym "OID", and replace it with "object ID".

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #582 In HTTP Status Codes, minor update to 405 description text

Description: As discussed and approved in the TWG review, replace:

"Requested HTTP verb not allowed on this resource."

with

"Requested HTTP verb was not allowed on this resource."

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #583 Add "eventual consistency" to terms section

Description: We should add a definition explaining what we mean by "eventual consistency" to the terms section.

Add a new term to Clause 3 as follows:

3.7 eventual consistency

a behavior of transactional systems that relaxes the immediate consistency guarantee in lieu of system availability and tolerance to network partition

NOTE: This is part of the CAP theorem, which states that of three properties of shared-data systems: data consistency, system availability and tolerance to network partition one can only achieve two at any given time.

Changes

Made the specified changes to Version 1.0.1m.

Trac Ticket #586 Missing text describing how ACLs control field access

Description: We need to include normative text that describes how ACLs control access to the value and to attributes of objects.

- Access to the valuerange and value fields are controlled by the "READ_OBJECT"/"WRITE_METADATA" ACL mask bits.
- Access to the metadata field is controlled by the "READ_OBJECT"/"WRITE_METADATA" ACL mask bits.
- Access to remaining fields is controlled by the "READ_ATTRIBUTES"/"WRITE_ATTRIBUTES" ACL mask bits.

This needs to be added to section 16.1.5 that describes the ACE Mask Bits. Need to propose spec text.

Given that you're aligning with the NFSv4 spec, I'd recommend using similar text to section 5.11.2 of rfc3530: <http://www.ietf.org/rfc/rfc3530.txt>

Replacing file/directory with the object/container you'd have:

Access	Description
READ_OBJECT	Permission to read the data of an object
LIST_CONTAINER	Permission to list the contents of a container
WRITE_OBJECT	Permission to modify the data of an object
ADD_OBJECT	Permission to add a new object to a container
APPEND_DATA	Permission to append data to an object
ADD_SUBCONTAINER	Permission to create a subcontainer in a container
READ_METADATA	Permission to read the metadata of an object
WRITE_METADATA	Permission to write the metadata of an object

EXECUTE	Permission to execute an object
DELETE_OBJECT container	Permission to delete an object or container within a container
READ_ATTRIBUTES object or container	The ability to read basic attributes (non-acls) of an object or container
WRITE_ATTRIBUTES object or container	Permission to change basic attributes (non-acls) of an object or container
DELETE	Permission to Delete the object
READ_ACL	Permission to Read the ACL
WRITE_ACL	Permission to Write the ACL
WRITE_OWNER	Permission to change the owner
SYNCHRONIZE	Permission to access object locally at the server with synchronous reads and writes

BTW, it looks like the CDMI spec has changed DELETE_CHILD (as used in Windows and NFSv4) to DELETE_OBJECT, which might be confused with DELETE. In Windows/NFSv4, DELETE_CHILD is only relevant on a directory and means that a user can delete things in that directory even if that user doesn't have DELETE permission on thing being deleted. A CDMI implementation relying on an underlying file system that implements DELETE_CHILD would find that DELETE_OBJECT has no relevance on objects and, on a container, allows both objects and subcontainers to be deleted. If DELETE_OBJECT is really DELETE_CHILD, I'd recommend calling it DELETE_CHILD.

TWG: Table approved with minor changes, see e-mail to TWG mailing list.

The same table format should be used for the below:

- ACE Types in section 16.1.2 ACE Type
- ACE Flags in section 16.1.4 ACE Flags

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #587 Global changes per review of Ralph's changes

Description: After reviewing Ralph's changes through Clause 8, TWG agreed to the following changes:

- 1 With respect to headers, intro sentences, and table titles, change:
"with CDMI Content" to "using CDMI Content Type"
"with non-CDMI Content" to "using a non-CDMI Content Type"
- 2 In each Synopsis section, change:

"shall be accessible" to "shall also be accessible"

"may be accessed" to "shall also be accessible"

"<ContainerName> is zero or more intermediate containers that already exist with one slash (i.e., "/") between each pair of container names."

to

"<ContainerName> is zero or more intermediate containers that already exist, with one slash (i.e., "/") between each pair of container names.

3 Under "Delayed Completion of Create", change

"Future accesses to the URI created (or the object ID) will succeed modulo any delays due to use of eventual consistency."

to

"Future accesses to the URI created (or the object ID) shall succeed except when delays occur due to an implementation's use of eventual consistency.

4 Change the description of objectName to

"Name of the object

If objectName and parentURI are both provided, appending the objectName to the parentURI shall produce a valid URI for the object."

5 Change the description of parentURI to

"URI for the parent object

If provided, the parentURI value shall be set to the URI path of the parent container."

6 Change the description of parentID to

"Object ID of the parent container object

The parentID field shall always be provided if the object has a parent container."

CHANGES Made the specified changes to Version 1.0.1k.

Trac Ticket #588 **Should there be text directly under the Clause 6 heading?**

Description: If I recall correctly, you said that under ISO rules, there can't be any text under a clause if it has sub-clauses.

If that is correct, Section 6 should have the text, "The example transactions in this chapter illustrate some of the more common CDMI operations." removed.

CHANGES Made the specified changes to Version 1.0.1l.

Trac Ticket #592 **Definition of backup_operator wrt ACL checks**

Description: In table 64, the "backup_operator" privilege is described as meaning that "All read ACL access checks are always successful."

This definition may be incorrect, since the intention is for this privilege to be consistent with the CIFS definition of backup operator.

TWG: Backup operator includes restore workflows. Thus, this text needs to be updated. Will need to wait until we have the right folks available to propose spec changes.

In Windows, a member of the Backup Operators and Administrators groups can backup and restore all files/directories (and associated attributes, ACLs and metadata) regardless of whether or not the user has permission to perform those operations via regular access control. This is achieved by enabling two privileges on the Backup Operators group:

Backup privilege bypasses file and directory permissions for backup and is equivalent to having list directory, read data, read metadata and read ACL.

Restore privilege bypasses file and directory permissions for restore operations and is equivalent to create/list directory, read data, metadata and ACLs, and set ownership.

For the purpose of the CDMI spec, I'd propose:

"backup_operator". Bypass regular ACL checks to allow backup and restore of objects and containers, including all associated attributes, metadata, ACLs and ownership.

BTW, the "administrator" privilege as described in the CDMI spec isn't consistent with the privileges granted to the Windows Administrators group. On Windows there is a TCB (trusted computing base) privilege that bypasses all access control (it's used by operating system processes) but that's different from Administrators. Members of the Administrators group are granted the ability to take ownership of any object/container and can perform all administrative operations (configure shares/exports, shutdown, backup/restore, set quotas etc.) but they are still explicitly subject to access control (ACLs). If an ACL does not grant access to Administrators, the only thing an administrator can do is take ownership. Of course, once an administrator has taken ownership, he/she can then change the ACL and give themselves access. The reason being that these operations would be subject to auditing (if that was enabled) so that they can be tracked.

Assuming object/container access control will typically be enforced by the underlying file system, I would assume that the CDMI "administrator" privilege is intended to align with the privileges granted to the Administrators group rather than the TCB privilege. Which behavior is intended?

Proposed Changes

- 1 In Table 64 - , in the "cdmi_member_privileges" row, description column, replace:
 - "backup_operator". All read ACL access checks are always successful.with
 - "backup_operator". Bypass regular ACL checks to allow backup and restore of objects and containers, including all associated attributes, metadata, ACLs and ownership.

- 2 In Table 64 - Required Settings for Domain Member User Objects, in the "cdmi_member_privileges" row, description column, replace:
 - "administrator". All ACL access checks are always successful.
 - with
 - "administrator". Allows the principal to take ownership of any object/container.

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #595 Figure 6 has bounding-box and shadows

Description: The updated diagram for figure 6 has a bounding box and problems where the shadows are.

- 1 The bounding box and shadows should be removed.
- 2 "Root" should also be changed to "Root Container", and should have the "Name/Value" box added.
- 3 The URIs for containers, capabilities and domains are also incorrect, in that they are missing their trailing slashes:
 - "https://<offering>" -> "https://<offering>/"
 - "https://<offering>/cdmi_capabilities" -> "https://<offering>/cdmi_capabilities/"
 - "https://<offering>/containerA" -> "https://<offering>/containerA/"
 - "https://<offering>/containerB" -> "https://<offering>/containerB/"
 - "https://<offering>/cdmi_domains" -> "https://<offering>/cdmi_domains/"

Discussed at TWG call, and approved.

CHANGES: Made the specified changes to 1.0.1l.

Trac Ticket #596 "root capabilities container" -> "root container capabilities"

Description: The phrase "root capabilities container" should be changed to "root container capabilities" to be clear as to what it is referring to.

CHANGES: Made the specified changes to 1.0.1l.

Trac Ticket #598 Clause 9 Changes

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h.

Page 48

- 1 Change "container" to "container object" consistently throughout text; likewise, change parent object to parent container object.
- 2 Change "This model allows direct mapping between CDMI-managed cloud storage and filesystems (for example), as CDMI containers may be exported as NFSv4 or WebDAV filesystems, with all metadata visible. As files are created, the files and directories are then visible through the CDMI interface acting as a data path. This mapping, though possible, is not further described in this standard.
to
This model allows direct mapping between CDMI-managed cloud storage and filesystems (e.g., NFSv4 or WebDAV). If a CDMI container object is exported as a filesystem, then the filesystem may make the CDMI metadata accessible via filesystem-specific mechanisms. As files and directories are created by the filesystem, they become visible through the CDMI interface acting as a data path. The mapping between filesystem constructs and CDMI data object, container objects, and metadata is outside the scope of this international standard.
- 3 Make sure all examples are formatted per ISO.
- 4 Change "fashion" to "a way that is similar to".

Page 49

- 1 Change "(may be thin provisioned)" to "(e.g., the device may be thin provisioned)."
- 2 Change "Conflicting writes via different paths are managed " to "Conflicting writes via different paths shall be managed".

Pages 50-61

- 3 Add cross-reference to "serialize" field name in Request Message Body - Create a Container Object using CDMI Content Type.
- 4 In Response Message Body table - Create a Container Object using CDMI Content Type, change "(returned only if present)" in the Requirements column to a footnote. Do this globally.
- 5 Under Synopsis, last paragraph, change "object" to "container object".
- 6 In paragraph after Response Message Body table - Create a Container Object using CDI Content Type, determine if this text should be part of the footnote.
- 7 Change "object" to "container object." (1st paragraph after bullets)
- 8 In Response Message Body, change "results in a status code" to "shall cause a 4xx ... status code to be returned"

Pages 63-69

- 1 Change "object" to "container object".
- 2 In last sentence in Synopsis (after bullets), change to two sentences.
- 3 In Delayed completion of a snapshot, change "On a snapshot operation for a container" to "If the Request Message Body contains a snapshot field,"
- 4 Change "Once created," to "After they are created," and consider changing this structure globally.
- 5 Consider adding a cross-reference to "Exported Protocols" (Clause 13) to the "exports" row of the Request message Body table. Consider a global change.

- 6 Change "object" to "data object" (Once created, the object may also be accessed at <root URI>/cdmi_objectid/<objectID>.).
- 7 In all Request Message Body tables, ensure footnote wording is consistent and that it includes the sentence "If more than one of these fields are supplied, the server shall respond with a 400 Bad Request error response."

Page 79

- 1 Change "goes up" to "increases". Check for this globally.
- 2 In queueValues row of Response Message Body table, change "i.e.," to "e.g.,"

CHANGES Made all specified changes in Version 1.0.11.

Changes discussed by TWG:**Page 49**

- 1 Consider adding a glossary entry for "thin provisioned".
TWG: Get definition from SNIA dictionary.
- 2 Consider adding a subclause that describes "eventual consistency" to clause 5.
TWG: Discussed; no change.
- 3 Consider adding a "Delayed Completion" subclause to clause 5 and then cross-reference where appropriate. It also would assure that the delayed completion feature operates in the same way in all the cases where it is used. It would also be easier to enhance the delayed completion feature in future CDMI versions.
TWG: Discussed; no change.

Pages 50-61

- 1 Rewrite "modulo any delays".
TWG: Addressed in Trac Ticket #587.
- 2 Define "privilege" in Terms clause; cross reference.
TWG: Considered and rejected.
- 3 Define "structure" (see exports field in Request Message Body - Create a Container Object using CDI Content Type).
TWG: Considered and rejected.
- 4 Change note. Per Ralph's comments:
 - a) This is not a note.
 - b) Consider revising the text to read: "The format of a GET for a container when fields are not being requested is outside the scope of this international standard." If that wording is repugnant, maybe the following is the intended meaning: "A GET for a container that does not request fields shall be processed as an error."
 - c) Is this statement true only for non-CDMI type container objects, or does it apply to CDMI type container objects (which means this text belongs in 9.4 too).
TWG: No change; already implemented.

Pages 63-69

Per Ralph's comments with respect to "also": "Access via objectID appears to be the only choice for this function. If the CDMI server assigns another name to this data object child, then a new sentence after this one would be the place where that requirement is specified.

TWG: Change "Once created, the data object may also be accessed at <root URI>/cdmi_objectid/<objectID>."

to two lines:

"When a new data object is created by posting to a container, the data object shall also be accessible at <root URI>/cdmi_objectid/<objectID>."

"When a new data object is created by posting to <root URI>/cdmi_objectid/, the data object shall only be accessible at <root URI>/cdmi_objectid/<objectID>."

CHANGES Made all specified changes in Version 1.0.1m.

Trac Ticket #599 Clause 10 changes

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h.

Page 81

- 1 Make sure all URL examples are formatted consistently.
- 2 Change "may have standard storage system metadata, such as an Access Control List (ACL) that permits access to this information to be restricted." to "may have an Access Control List (see 16.1) that restricts access to this information."

Page 84

- 1 Change "metadata items standardized here." to "metadata items specified by this international standard, as long as the field names for those metadata items do not begin with "cdmi_"."
- 2 Change "just as domains" to "in the same way that domains" (to lower the risk of mistranslation).
- 3 Change "CDMI standard" to "this international standard".
- 4 Change ", which form the fundamental basis for access control and object/container ownership," to "(see 16.1),"
- 5 Change "may have standard metadata, such as ACLs that permit access to this information to be restricted." to "may have standard metadata, such as an Access Control List (see 16.1) that restricts access to this information."
- 6 See notes with respect to the following paragraph:
 "The domain membership container may also contain sub-containers with user data objects. User data objects in these sub-containers are treated the same as user data objects in the domain membership container, and no meaning is inferred

from the sub-container name. This is allowed to create different access security relationships for groups of user objects (via container ACLs) and to allow delegation to common user lists.

- Change "user objects" to "data objects". (If not, then change all instances of "data objects" to "user objects" globally (including in subclause titles), or define a fifth type of object, specifically "user object".
- Change "objects" to "data objects".
- Change "user data objects" to "data objects".

Page 86

- 1 After second example, add subclause heading, "Domain Usage in Access Control".
- 2 Change "When a transaction is initiated against a CDMI URI, the domain associated with the object at the URI (as indicated by the domainURI) is used as the authentication context." to "When a CDMI URI is used as the address for a transaction, the domain associated with the object (i.e., the domain indicated by the domainURI) specifies the authentication context."
- 3 Change "against" to "to" or "with".
- 4 Change "principal is then" to "user's principal is".
- 5 Change "record, a delegation, where the delegation" to "record that is a delegation in which the delegation".

CHANGES Made all specified changes in Version 1.0.11.

Changes discussed by TWG:

Page 81

- 1 In the description of `cdmi_domain_enabled` in 10.1.1, with regard to "Values may be "true" or "false"." - Is there a good reason for confusing implementors by defining a field where "false" cannot be indicated by omitting the field? Consistency in the standard is the best way to ensure consistency in implementations.
TWG: Change "Values may be "true" or "false" " to "Values shall be "true" or "false"". Rest of items addressed in trac ticket #610.
- 2 Under 10.1.2 Domain Summaries, regarding "reporting functionality" - What is the purpose of "gathering information" if that information is never reported (i.e., subject to a form of reporting functionality)? Imagine what this will read like when translated into French.

TWG: Delete offending sentence.

Page 84

See notes with respect to the following paragraph:

"The domain membership container may also contain sub-containers with user data objects. User data objects in these sub-containers are treated the same as user data objects in the domain membership container, and no meaning is inferred from the sub-container name. This is allowed to create different access security relationships for groups of user objects (via container ACLs) and to allow delegation to common user lists.

- ACLs on what containers? This parenthetical expression probably needs to be moved to a separate, clearly stated sentence.
- "user list" needs to be added to the glossary, or a cross reference to a sub-clause that describes these beasts needs to be added here.

TWG: Delete "(via container ACLs)". Change "user lists" to "set of members".

CHANGES Made all specified changes in Version 1.0.1m.

Trac Ticket #600 Clause 11 and 12 changes

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h.

Clause 11 - Pages 97-110

- 1 Change "same mechanism used to create data objects." to "mechanisms described in 9.10 and this clause."
- 2 Change "metadata. For example," to "metadata (e.g.,"
- 3 Change "finance"." to ""finance")." (to complete the e.g.)
- 4 In the queueValues table row, change "i.e.," to "e.g.,"
- 5 Change "is considered" to "is processed as if it is".
- 6 Change "Queue" to "queue object".

Clause 12

Page 116

- 1 Change "against" to "for".
- 2 Change "It is important to note that capabilities" to "Capabilities"
- 3 Change "(ACL) associated" to "(see 16.1) associated".
- 4 Change "to be performed against" to "on".
- 5 Change "Every CDMI data object, container domain, and queue" to "Every CDMI data object, container object, domain object, and queue object"
- 6 Change "as part of this CDMI standard" to "in this international standard".
- 7 Change "later in this section" to "in this clause".
- 8 Change "this standard" to "this international standard".
- 9 Change "permitted to allow cloud storage system implementors to add additional capabilities." to "permitted so that cloud system implementors have the ability to extend the use of capabilities in ways that are outside the scope of this international standard."
- 10 Delete the following sentence, as it is just a restatement: "The base set of CDMI capabilities are based on the operations defined in the previous sections, with additional cloud-specific capabilities added based on use cases for standard cloud storage".

11 Change "The hierarchy of capabilities (see Figure 8) shows" to "Figure 8 shows".

12 Change "in and offering and shows" to "in an offering as well as"

Page 117-128

1 Delete "cdmi_xmlrepresentation" capability. (Done in subsequent version.)

2 Make sure there are no blank pages in the ISO/IEC international standard.

CHANGES Made all of the specified changes in Version 1.0.11.

Changes discussed by TWG:

Clause 11

Comment: Consider creating a queueValues subclause and create a cross-reference to it in the tables and describe this field.

TWG: No change.

Clause 12

1 Clause 12 - Comment: With respect to "list capabilities" in paragraph 4 of 12.1 Overview:

I believe this intends to require the ability to return all the capabilities metadata via a single operation. This might be clarified by adding an (e.g., example) of how to retrieve a list. The only other alternative is to add a glossary entry for "list capabilities".

TWG: Change to: "All CDMI-compliant cloud storage systems shall implement the ability to read capabilities, but support for the functionality indicated by each capability is optional."

2 In the last sentence of Section 12.1, it states, "Capabilities do not have CDMI metadata."

At least the following capabilities appear to violate this: `cdmi_metadata_maxitems` and `cdmi_metadata_maxsize`.

TWG: Change sentence to "Capabilities do not have a CDMI metadata field."

3 Review comment: With respect to `cdmi_export_iscsi` capability: `iscsi` is a curious capability name to indicate FC export support.

TWG: Change definition to "... cloud storage system supports iSCSI exports."

4 Comment: With respect to "capability of `cdmi_security_encryption`" in Table 12:

If present, this capability lists the encryption algorithms and key lengths supported. If absent, objects shall not be encrypted. When a CDMI implementation supports at-rest encryption, the system-wide capability of `cdmi_security_encryption` specified in Table 10 of Section 12.1.1, "Cloud Storage System-Wide Capabilities" shall be set to "true". Otherwise, it shall not be present, indicating that there is no support for at-rest encryption.

*This is not the notation that table 20 or any capability entries in clause 8, clause 9, clause 10, or clause 11 use to name a capability. Pick one notation and used it consistently throughout this international standard.

TWG: No change; already fixed.

- 5 Define RPO and RTO in the Terms clause.

TWG: No change; already fixed.

CHANGES Made all of the specified changes in Version 1.0.1m.

Trac Ticket #601 Changes to clause 13, 14, and 15 - discussion topics for TWG

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h. Text in italics are Ralph's comments.

Clause 13 - Page 129

- 1 With respect to "CDMI containers":

A strict-constructionist reading of this would conclude that data objects, queue objects, and perhaps others are not exportable. The TWG might wish to eliminate any opportunity for reader extrapolations on this thought.

TWG:

- 2 With respect to cloud computing infrastructure management:

Figure 9 labels this "Computing and Storage Infrastructure" not "cloud computing infrastructure management". Which one is correct?

TWG:

- 3 With respect to "CDMI Exported Protocols" in Figure 9:

If I am extrapolating the descriptive text below the figure correctly, the "CDMI Exported Protocols" are iSCSI, NFS, and WebDAV. This notion is not conveyed by this location of the non-cloud cloud.

TWG:

Page 130

With respect to "a type of exported protocol":

Is "a type of exported protocol" a shorthand way of saying "... JSON object called exports ..."? If yes, please devise a way to tell it like it is. If no, please find a way to express this thought without suggesting that CDMI is exporting the protocols (because I have yet to find any evidence that such a thing is being specified in this international standard).

TWG:

Page 131

- 1 In 13.3.2 "The following code creates a container" -

Either 9.2 is sadly mistaken, or the "code" shown on this page does not "create a container". If I read rest of the document correctly, "Adding the following JSON object to a container creation or update operation (see clause 9) adds a new iSCSI

export to the container being processed or updates the information in the current iSCSI export."

This leads to the interesting question, "Can one container have more than one iSCSI export?" If yes, then how is that supposed to be represented?"

TWG:

- 2 Move 2nd paragraph under "13.3.2 Create Container" according to the following:
Depending on how code examples are ultimately shown, this paragraph may need to be located after the code fragment. A related question is: "How are errors in an exports JSON structure to be handled?" Suppose somebody puts a LUN in their iSCSI export, what happens?
TWG: Move paragraph as specified, leaving an intro sentence for the code segment.

Page 132

In 13.3.3 Modify an Export:

The information in this subclause appears to be redundant with 13.3.1.

13.2, 13.4, and 13.5 should have "Read" and "Create/Update" subclauses added, or the subclauses should be removed from 13.3. Showing favoritism to one protocol is not a good idea.

TWG: From code example in 13.4.3, delete line ending with "host1".

Clause 14 - Page 133

1st paragraph regarding snapshots:

Troublemaker that I am ... what happens when two snapshot requests specify the same name for the snapshot container?

TWG: Add text to Table 43, "snapshots row" and add a new paragraph that states "Creating a snapshot with the same name as an existing snapshot, the new snapshot will replace the existing snapshot."

Clause 15 - page 134

- 1 In 15.1, with respect to "If a container has an exported block protocol, the serialized data may contain the block-by-block contents of that container along with its metadata as if it were a data object.":
Since containers contain data objects, not data per-se, the sentence in which this phrase occurs seems to be obfuscating information that might be important for consistent implementations.
TWG: Delete last part of sentence ... "as if it were a data object."
- 2 In 15.1, with respect to "Only objects that the user who is performing the serialization operation has permissions to read shall be included in the resulting serialized object.":
The "user" is not sitting at his/her terminal serializing the data. The cloud implementation is "performing" the serialization operation. The "user" merely requested that the operation be performed. On a related subject ... Given the time required to serialize all the data objects in a cloud container, the user might have requested the serialization operation many moons ago. This calls into question when may of

the tests described in clause 15 (e.g., if the user has permission to read a CDMI object) are performed.

TWG: Replace last sentence as follows: "When performing a serialization operation, objects shall only be included if the principal initiating the serialization has sufficient permissions to read those objects."

- 3 With respect to the first paragraph under 15.2 and the first three sentences of the second paragraph:

It is very difficult to determine what serialized data import operations are allowed. The other point that is not clear is whether importing to a data or queue object results in the object being replaced or updated.

TWG: No change.

- 4 In 15.2, with respect to "but the exported block protocol shall be set up with a separate update.

I cannot determine what a "separate update" is, which means the likelihood of interoperable implementations seems low.

TWG: Delete last sentence of 2nd paragraph beginning with "If a serialized data object contains block-by-block ..."

- 5 In 15.2, with respect to "If the user who is deserializing"

The "user" is not deserializing anything. The "user" merely instructed the cloud to do the deserializing. There are two more instances of this problem on the next page, but this is the last time it will be noted.

TWG: No change.

Page 135

- 1 With respect to "If the user does not have the "cross_domain" privilege, only the domainURI of the parent object may be specified."

This "may be specified" looks like a back-handed way of describing an error condition. To avoid translation problems, a more straight-forward statement is recommended.

TWG: Change this sentence to "To specify a domainURI other than the domainURI of the parent, the user shall have the cross-domain privilege. If the user does not have the cross-domain privilege, and specifies a domainURI other than the domainURI of the parent, a 400 Bad Request response shall be returned."

Will open a new ticket to address result codes for error conditions for insufficient privileges. See ticket #614.

- 2 With respect to "Because metadata is inherited from enclosing containers, all parent metadata shall be represented in the canonical format (essentially flattening the hierarchy)."

Is the hierarchy flattened, or simply reduced to recursive JSON entries? If the hierarchy is flattened, then how can it be restored during a deserialize operation?

TWG: Create a new ticket (see ticket #615).

Ticket assigned to Marie to incorporate changes.

CHANGES Made all specified changes except those in red. Reopened ticket for TWG discussion.

Trac Ticket #602 Changes to clause 13, 14, and 15 - editorial changes

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h. Text in italics are Ralph's comments.

Clause 13 - Page 129

- 1 Remove figure title from within graphic.
- 2 Add VDMI to bibliography with reference to standard, if applicable.
TWG: Delete Delete [OCCI_VMID] and [CDMI_ObjectID] from Figure 9.
- 3 Fix all graphics so text is not sideways. (Figure 9 and do a global check.)
- 4 Change "hook" to "associate" or other appropriate word.
- 5 Add OCCI to the glossary (if not already there).

Page 130

- 1 Change "To do this interoperably" to "To support exported protocols and improve their interoperability with CDMI".
- 2 Change "would perform the following operations as an example" to "perform operations that align the architectures, including the following."
- 3 Change the numbers to bullets.
- 4 Change "done" to "accomplished".
- 5 Change last paragraph of 13.1 to "This international standard defines JSON export structures for several protocols. Export protocol structures may also be defined for proprietary and vendor extensions of protocols (see Annex B)".
- 6 Make sure objectID is consistent throughout doc.
- 7 Change 1st paragraph of 13.3 to ""CDMI defines the export of a container using the iSCSI protocol (see RFC 3720). Each container is exported as a single SCSI Logical Unit at a Logical Unit Number (LUN). One or more iSCSI initiators import the LUN through an iSCSI target node and port using one or more iSCSI network portals (IP addresses)."
- 8 Eliminate all hanging paragraphs throughout document (add subheads where appropriate).

Page 131 and 132

- 1 Change "13.3.2 Create Container" to "13.3.2 Create and Update Containers".
- 2 Page 132 - Remove note from 13.4.

Clause 14 - Page 133

- 1 Change "snapshot is taken" to "snapshot is requested".
- 2 Change "The operation results in a child destination container of the cdmi_snapshots container under the source container." to "A snapshot operation creates a new container to contain the point-in-time image. The first processing of a snapshot operation also adds a cdmi_snapshots child container to the Source Container. Each new snapshot container is added as a child of the cdmi_snapshots container."

- 3 Change Figure 10 title "Snapshot Operation" to "Snapshot Container Structure".
- 4 Change "To take a snapshot, a Container Update operation is performed, as described in Section 9.6, "Update a Container (CDMI Content Type)", supplying the snapshot parameter (the name of the snapshot)." to "A snapshot operation is requested using the Container Update operation (see 9.6) in which the snapshot field specifies the requested name of the snapshot."
- 5 Remove Table 17 and the sentence that precedes it. This sentence and table 17 are redundant information. The cross reference to 9.6 is sufficient. Furthermore, duplicating this information presents the possibility that it will become out of synch with 9.6 in a future revision of this standard.
- 6 Change last sentence "Snapshots may be accessed in place or used as the source for copy operations, restoring a container to a previous point in time." to "A snapshot may be accessed in the same way that any other CDMI object is accessed. An important use of a snapshot is to allow the contents of the Source Container to be restored to their values at a previous point in time using a CDMI copy operation."

Clause 15 - page 134

- 1 Fix hanging paragraph from Clause 15 by adding "Overview" as first subclause.
- 2 Remove first sentence beginning with "Cloud storage provides benefits.."
- 3 Change "data is moved between, into, or out of clouds in bulk operations" to "bulk data movement is needed between, into, or out of clouds"
- 4 Change "Cloud serialization operations are enabled by normalizing data to a canonical, self-describing format." to "Cloud serialization operations provide a means to normalize data to a canonical, self-describing format."
- 5 Change "This stream may be transported in any desirable conveyance, and as long as no alteration occurs," to "As long as this byte stream is not altered during the transfer from source to destination,"
- 6 Fix hanging paragraph under 15.2 by adding "Overview" as first subclause.

Page 135 and 137

- 1 Move "see Annex B" from the end of the paragraph to follow "custom metadata and key values,".
- 2 Change "container JSON object" to "canonical serialized JSON format".

CHANGES Made all specified changes in Version 1.0.1l, except reference to VMID in Figure 9, which was updated in Version 1.0.1m.

Trac Ticket #603 Changes to Clauses 16, 17, and 18 (editorial)

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h. Text in italics are Ralph's comments.

Clause 16 - Pages 138-150

- 1 Fix hanging paragraph (add intro clause).

- 2 Make sure code examples stay on same page.
- 3 Remote "Note that" from beginning for first two paragraphs after code example.
- 4 Change the examples in 16.1.7 to ISO/IEC example format.
- 5 Fix alignment in first code example.
- 6 Since ISO/IEC directives do not describe ordered lists, add description of the special-case list format in Clause 4.
- 7 Remote "Note that" from paragraph preceding 16.1.9.
- 8 Change "Once" to "After".
- 9 Change "As described in Table 20, data system metadata is inherited" to "Data system metadata (see table 20) is inherited"
- 10 Make sure table rows do not break across pages.
- 11 In Table 20, in the cdmi_encryption table row, format last paragraph as bullet.
- 12 Change "CDMI envisions" to "This international standard supports"
- 13 Change "CDMI also envisions" to "This international standard also supports"

Clause 17 - Pages 152 to 155

- 1 Fix hanging paragraph by adding an overview clause.
- 2 Change "CDMI standard" to "this international standard".
- 3 Change "object notifications" to "object notifications (see clause 19)".
- 4 Change "group" to "number", "array", or "range".
- 5 Change "such as," to "e.g.,"
- 6 Change "Otherwise, it" to "Otherwise, "cdmi_security_audit"."
- 7 Change the "For example," paragraph to an ISO/IEC example.
- 8 Change ", and clients can operate in a disconnected and casual manner. For example, a logging analysis application might use logging queues to obtain information about actions performed to a cloud, and it may receive only logging messages that are relevant to a domain or set of objects." to ". If different logging queues are used for different clients, then each client operates independently from the others (e.g., an analysis application may retrieve information about actions performed in a specific domain or set of objects using a logging queue that is uniquely configured to its specific needs)."
- 9 Change "notification queues" to "notification queues (see clause 19)".
- 10 Change "message. (User has any ACE from Section 16.1.5, "ACE Mask Bits".)" to "message, i.e., user has any ACE from Section 16.1.5, "ACE Mask Bits")."
- 11 Change "correlation (knowing" " to "correlation (i.e., knowing"
- 12 Change "the CDMI standard" to "this international standard"

Clause 18 - Page 156

- 1 Fix hanging paragraph by adding an introduction clause.
- 2 Change "to be held for specific purposes (typically litigation)" to "to be held for specific purposes (e.g, litigation)"
- 3 Change all references of "chapter" or "section" to "clause".
- 4 For consistency, change "items" to "objects".

- 5 Change "CDMI application" to "CDMI client".
- 6 Change "apply to" to "affect". e.g., "CDMI retention, deletion, and hold management affect any CDMI application".
- 7 In first bullet under 18.1, change "however" to "but".
- 8 Change "tries" to "attempts". Change other versions of "try" to match tense.
- 9 Change "non-fatal errors" to "errors".
- 10 Change "CDMI application tries" to "CDMI client tries".

Page 157

- 1 Moved bulleted list in 18.1 to 18.2, since it only discusses retention.
- 2 Delete last sentence of third paragraph starting "This failure should be "
- 3 Change two instances of "new" to "destination" and replace "existing" with "source".
- 4 Change "an application-" to "a client-".

Pages 159 to 160

- 1 Change bulleted list to sentences following first sentence of 2nd paragraph.
- 2 Delete last sentence/paragraph on the page.
- 3 Change "For example" sentence to an ISO example.
- 4 Change "are strongly encouraged " to "should".

CHANGES Made all specified changes in Version 1.0.1l.

Trac Ticket #604 Changes to Clause 16, 17, & 18 (Discussion)

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h.

Clause 16 - Page 137

RE: well-known mechanism (1st paragraph)

If the mechanism is well-known, might there be a published document (maybe even a standard) that can be referenced? After over 100 pages of HTTP and JSON, it is unsettling to find recourse to C code examples for ACL definitions.

TWG: No change.

Page 139

- 1 RE: This ordering conforms to both Windows and NFSv4.
Why CDMI selected this ordering is not relevant ... unless the hidden objective of this statement is to say that the use of ACLs is incompatible with iSCSI and Web-DAV exports.
TWG: Delete sentence "This ordering ... NFSv4."
- 2 RE: DNS domain

This is the only mention of DNS in the entire document, which leads to the question, "Does CDMI need to be dependent on DNS or is there more CDMI-centric way to say this?"

TWG: Change to "...rather than in the context of a particular external security domain (see Table 113)."

3 RE: NFS client accesses the server

Isn't NFS a supported export for CDMI? So, is "the server" the CDMI server or the NFS server? Yes, I am confused. The more I read the more it looks like 16.1 was beamed in from Mars (or some other place where CDMI is a foreign language).

TWG: Change "NFS" to "CDMI" in 16.1.3 - 2nd & 3rd sentence.

Page 140-145

1 Page 140 - RE: CDMI_ACE_FLAGS_OBJECT_INHERIT_ACE (and others)

Everywhere else in this standard, `cdmi_` names are lower case, why be inconsistent here?

TWG: Format tables to avoid wrapping. Add "INHERITED_ACE" after "/" in last line of code. Add 5th bullet to describe last line of code.

"If a client constructs an ACL which represents all of the ACEs that apply to an object, including inherited ACEs, ACEs which are inherited shall be flagged with CDMI_ACE_FLAGS_INHERITED_ACE."

2 Page 142 - RE: First code example - Is this example JSON or C? It looks like JSON.

TWG: No change - (it's JSON)

3 Page 144 - RE: "ACE flags and masks are members of a 32-bit quantity that is widely understood in its hexadecimal representations."

To flog the horse first beaten in a 16.1.3 comment one more time, why not build lists of these hex values, reference ACL processing rules published elsewhere, and ashcan all of the C code?

TWG: Considered but rejected - No change.

4 Page 145 - RE: "All objects" -

Of late, I have become enamoured of "All CDMI objects" for cases such as this. It avoids somebody thinking that only data objects are required to permit arbitrary user-defined metadata items.

TWG: Change "All objects" to "All CDMI objects"

Page 146

1 RE: Table 19. "cdmi_size - Optional"

The "shall be generated" in the introductory sentence text on the previous page is in direct conflict with all the "Optional" assignments in this column. Are these metadata items mandatory or optional?

TWG: No longer relevant. No change.

2 RE: Table 19. "cdmi_acl - Optional."

If I read 16.1 correctly, the absence of a `cdmi_acl` on the CDMI root makes the entire cloud inaccessible for any operations, including changing the ACL or creat-

ing containers. Viewed in this light, adding some specific details about how this metadata item "shall be filled in by the system" seems like a good idea.

TWG: Create new ticket for key words to be added (optional, mandatory, conditional, etc.) - see trac ticket #616.

Page 150

RE: 16.5 Support for Data Copies

This subclause does not look like a primarily metadata topic. Consider moving it to clause 5.

TWG: Delete 16.5.

Clause 17 - Page 152

1 RE: First paragraph

My stomach churns at the repeated mention of regulatory issues in this paragraph. Somewhere in the world, a bureaucrat is waiting to pounce on the ISO/IEC review of this document and claim that his country's regulations are not properly considered. Would it be possible to justify logging using criteria other than regulatory needs?

TWG: Delete.

2 RE: "...three functional areas ... include:"

If there are only three functional categories as the previous sentence claims, then these "are" them. Of course, if the "three" is removed from the previous sentence, "include" might be right here.

TWG: Remove three.

3 RE: "Information contained in these logs shall be for CDMI operations. Logging of non-CDMI operations is optional, but may be provided through a CDMI logging interface. Is it only "access" to logged information that is provided, or can the logging operations themselves use the CDMI interface?"

TWG: Delete.

4 RE: "system", "storage system"

Okay! I have finally cracked. Are we defining the behavior of a "system", a "storage system", or a "CDMI system"? Pick a term and use that one term globally.

TWG: Changed to "cloud storage system."

5 RE: "principal" - The only "principal" defined is for ACLs. Should this read "ACL principal (see 16.1)"?

TWG: No change.

6 RE: "It is anticipated that a standardized format and set of log messages for CDMI operations will be added to a future release of the standard."

Go ahead! Wave a red, "this standard is not ready for publication" flag in front of the ISO/IEC reviewers, but do not be surprised at the results. Perhaps the better part of valor is to axe clause 17 for the time being.

TWG: Delete this paragraph.

Page 155

- 1 RE: `cdmi_logging_status` description "String Indicates if the query is in progress or complete. The three values defined are "Error", "Processing", and "Complete"."
Is this a shorthand way of saying that queue object creation may be delayed? If yes, why not replace all of it with a sentence that references 11.2 for details of queue object creation?
TWG: Addressed by ticket #494.
- 2 RE: "The accuracy and integrity of the log entries depend on the accuracy and integrity of the clock"
This appears to say that a logging queue will not return valid JSON objects if the clock is bogus. Perhaps it could be made more specific to avoid such broad-brush interpretations.
TWG: Change to "The timestamp accuracy and ..."
- 3 RE: "NIST standard time"
This looks like another sure-fire way to attract a No vote from the Russians or Japanese.
TWG: Change "NIST" to "a"
- 4 RE: "log messages should be signed"
If this standard provides a way to accomplish this signing function, then do not be vague. Require the use of the CDMI signing feature. Otherwise, this might be another reason to set clause 17 aside for the ISO/IEC submission.
TWG: Changed online.
- 5 RE: "requiring the presence of sequence counts"
Second verse ... same as the first. Point to a CDMI feature that does this or forget it for now.
TWG: Deleted paragraph.

Clause 18 - Page 156

- 1 RE: non-fatal errors:
This clause is the only portion of the standard that discusses non-fatal errors (or fatal errors, for that matter). On the next page the specified error is HTTP code 403, which other parts of this document describe as "Forbidden: This user is not allowed to perform this request." and "Client lacks the proper authorization to perform this request." I am having a tough time squaring these definitions with "non-fatal". Suggest deleting "non-fatal" where ever it appears.
TWG: Deleted "non-fatal". Check for this globally.
- 2 RE: "one concurrent"
I cannot recall ever having seen the phrase "one concurrent". Since concurrency cannot be an issue in a system with only one thing to discuss, one is tempted to call "one concurrent" a non sequitur. It is also difficult to make this phrase jibe with the "... evaluate all such retention ..." statement in the last paragraph on this page. Attempts should be made to determine the intended meaning of this statement, and rewrite it accordingly.

TWG: Delete "concurrent".

3 Page 157, last sentence "... or unexpected state changes on operations"

This leaves me with the impression (forboding, really) that the MIME type of the object might change unexpectedly. Better specificity here would go a long way to making the feature easier to implement.

TWG: Delete sentence beginning with "When an object is on hold ...failures or unexpected state changes ..."

Page 160

1 RE: "stratum 1 time is"

Consider clarifying the importance of this "requirement" by changing "is" to either "shall be" (not recommended unless "stratum 1 time" is a standard of some kind) or "should be". Also consider coordinating the time requirements in 17.6 with the time requirements in this subclause.

TWG: Delete 2nd sentence beginning with "A stratum 1 ...". In the third sentence, change "the reference clock" to "standard time".

2 RE: stratum 1 time

"stratum 1 time" needs to be defined in the glossary.

TWG: Not applicable -

CHANGES Made all of the specified changes to Version 1.0.1m.

Trac Ticket #605 Changes to Clause 19 & 20, Annex A & B (TWG review)

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h.

Clause 19 - Page 161

RE: "When creating a notification queue, the metadata described in Table 24 shall be provided. Attempts to alter metadata in this table will result in an HTTP 403 Forbidden HTTP status code. Once a notification queue has been created, with the exception of `cdmi_queue_type`, the metadata items in this table cannot be altered.

This exception conflicts with the first sentence in this paragraph. Pick a single rule, and stick to it.

TWG: No change.

Page 164

RE: Description of `cdmi_notification_status` "Indicates if the query is in progress or complete. The two values defined are "Error", "Processing", and "Complete"."

Is this a shorthand way of saying that queue object creation may be delayed? If yes, why not replace all of it with a sentence that references 11.2 for details of queue object creation?

TWG: Addressed in trac ticket #494.

Clause 20 - Page 165

RE: "When creating a query queue, the metadata described in Table 26 shall be provided. Attempts to alter metadata in this table will result in an HTTP 403 Forbidden HTTP status code. Once a query queue has been created, with the exception of `cdmi_queue_type`, the metadata items in this table cannot be altered.

This exception conflicts with the previous sentence.

TWG: No change.

Page 166

RE: Description of `cdmi_query_status` "A string indicating if the query is in progress or has completed. The value shall be the string "Error", "Processing", and "Complete"."

Is this a shorthand way of saying that queue object creation may be delayed? If yes, why not replace all of it with a sentence that references 11.2 for details of queue object creation?

TWG: Addressed in trac ticket #494.

Annex B - Page 180

TWG: All Annex B changes moved to new ticket (see #618) to be incorporated into Version 1.1. For this version, remove Annex B.

Reassigned to Marie to remove Annex B from 1.0.1m.

CHANGES Removed Annex B from 1.0.1m.

Trac Ticket #606 Changes to Clause 19 and 20; Annex A and B (editorial)

Description: Recommended changes per Ralph's review of CDMI 1.0.1h dated March 30, 2011 in ENDLTxReview of CDMI[9 to End].pdf.

Note: Page numbers refer to the spec page number in CDMI 1.0.1h. Text in italics are Ralph's comments.

Clause 19 - Page 161

1 Change:

"and clients may operate in a disconnected and casual manner. For example, an application might use notification queues to keep its database current without having to do full scans of a container to discover what objects have been added, modified, or removed. "

to

"If different logging queues are used for different clients, then each client operates independently from the others (e.g., a storage management application may use a notification queue to keep its database current without having to do full scans of a container to discover what data objects have been added, modified, or removed)."

- 2 Change "Attempts to alter metadata in this table *will* result in an HTTP 403 Forbidden HTTP status code." to "Attempts to alter metadata in this table *shall* result in an HTTP 403 Forbidden HTTP status code."

Annex A

- 1 Fix hanging paragraph by adding an introduction clause.
- 2 Change "GET Request-URI" to "<GET Request-URI>". Change this format globally, if applicable.

Annex B - Page 180

- 1 Change "The CDMI standard" to "This international standard"
- 2 Change ""cdmi"" to ""cdmi_"".
- 3 Change "section" to "annex".
- 4 Change " non-version-aware CDMI client" to "CDMI client that is not aware of versioning".
- 5 Under B.2 Terms, add an introductory sentence before the list. Change the terms/ definitions per ISO/IEC specs.

Page 181

- 1 Change "Metadaw" to "Metadata".
- 2 Change "object (data or metadata)" to "data object, including metadata".
- 3 Change "container. For example," to "container, e.g.,"

Page 182

- 1 Change "versioned object" to "versioned data object".
- 2 Change "object versions" to "data object versions".
- 3 Change "the 8FE object" to "the object whose ID ends with 8FE". Apply the same change to all three instances of this shorthand in the paragraph.

Page 183 - 184

- 1 Change "To illustrate this" and text after to an ISO/IEC example.
- 2 Change "system" to "destination" (twice).
- 3 Change "as" to "since" (global change to facilitate translation).
- 4 Move last paragraph beginning with "If a version of a data object..." to immediately after the copy paragraph.

CHANGES Made all specified changes in Version 1.0.11.

Trac Ticket #607 Clarify meaning of "Mandatory/Optional" in "Requirement" column

Description: In request and response tables, we have a column labelled "Requirement" that contains "Mandatory" or "Optional". We need to be clear and consistent regarding what these two values mean.

TWG Notes: The word "discretionary" or "conditional" may be useful. This would indicate to look at the description for more details.

The Requirement column contains one of three values:

- Mandatory. The value specified in this row shall be provided.
- Optional. The value specified in this row may be provided.
- Conditional. If the condition specified in the "Description" cell of this row (to the left of the Requirement) evaluates to true, the value specified in this row shall be provided. Otherwise it may be provided unless the Description specifically prohibits it, in which case it shall not be provided.

TWG: Approved.

Create new sub-section after section 4.2, with the following text:

4.3 Request and Response Body Requirements

In request and response body tables, the requirement column contains one of the following three values:

- Mandatory. The value specified in this row shall be provided.
- Conditional. If the condition(s) specified in the "Description" cell of this row (to the left of the Requirement) is met, the value specified in this row shall be provided. Otherwise it may be provided unless the Description specifically prohibits it, in which case it shall not be provided.
- Optional. The value specified in this row may be provided.

CHANGES Made all specified changes in Version 1.0.1m.

Trac Ticket #609 Typos in CDMI 1.0.1I draft

Description: Skimming through the latest I.0.1I draft, there are a number of typos and other trivial errors that should be fixed:

The below proposed spec text changes address the remaining issues:

Clause 2

Issue 1 - Clause 2, second paragraph, replace:

"...referenced specifications the staters of an international standard."

with

"...referenced specifications the status of an international standard."

Clause 5

- 1 Issue 3 - Clause 5.7, fourth paragraph, replace:

"Conformant cloud offerings may support a subset of the CDMI, as long as they expose the limitations in the capabilities report via the interface."

with

"Conformant cloud offerings may support a subset of the CDMI, as long as they expose the limitations in the capabilities reported via the interface."

- 2 Issue 5 - Clause 5.14, second paragraph, replace:

"Unless otherwise specified, all date/time values are in the ISO 8601:2004 start date/end date representation (YYYY-MM-DDThh:mm:ss.ssssssZ/YYYY-MM-DDThh:mm:ss.ssssssZ)."

with

"Unless otherwise specified, all date/time intervals are in the ISO 8601:2004 start date/end date representation (YYYY-MM-DDThh:mm:ss.ssssssZ/YYYY-MM-DDThh:mm:ss.ssssssZ)."

- 3 Issue 6 - Clause 5.15, replace:

"CDMI version 1.0.1 introduces the concept of value transfer encoding to enable the storage and retrieval of arbitrary binary data via CDMI content-type operations. Data objects created by CDMI 1.0.1 clients through CDMI content-type operations shall have a value transfer encoding of "utf-8", and data objects created through non-CDMI content-type operations shall have a value transfer encoding of "base64".

Data objects with a value transfer encoding of base 64 shall not have their value field accessible to CDMI 1.0.1 clients through CDMI content-type operations. Attempts to read the value of these objects shall return an empty value field ("") to these clients. CDMI 1.0.1 clients can detect this condition when the `cdmi_size` metadata is not 0 and the value field is empty."

with

"CDMI version 1.0.1 introduces the concept of value transfer encoding to enable the storage and retrieval of arbitrary binary data via CDMI content-type operations. Data objects created by CDMI 1.0 clients through CDMI content-type operations shall have a value transfer encoding of "utf-8", and data objects created through non-CDMI content-type operations shall have a value transfer encoding of "base64".

Data objects with a value transfer encoding of base 64 shall not have their value field accessible to CDMI 1.0 clients through CDMI content-type operations. Attempts to read the value of these objects shall return an empty value field ("") to these clients. CDMI 1.0 clients can detect this condition when the `cdmi_size` metadata is not 0 and the value field is empty."

Clause 7

- Issue 7 - Clause 7.3, third paragraph, replace:

"To continue, when CDMI clients receive a 302 Found redirect, they should retry the operation on the URI contained with the "Location" header."

with

"To continue, when CDMI clients receive a 302 Found redirect, they should retry the operation using the URI contained within the "Location" header."

Clause 8

- 1 Issue 10 - Clause 8.4.6, "valuerange" row, description, replace:

"• If the object value has gaps (due to PUTs with non-contiguous value ranges), the value range will indicate the range to the first gap in the object value."

with

"• If the object value has gaps (due to PUTs with non-contiguous value ranges), the value range will indicate the range to the first gap in the object value."

- 2 Issue 10 - Clause 8.4.6, "value" row, description, replace:

"• The value field shall only be provided when the "completionStatus" field contains "Complete."

with

"• The value field shall only be provided when the "completionStatus" field contains "Complete"

(Removing extra ".", and adding in missing ")

- 3 Issue 12 - Clause 8.5.8, Example 3, replace:

"Content-Range: bytes 0-10"

with

"Content-Range: bytes 0-10/37"

- 4 Issue 13 - Clause 8.7.8, Example 2, replace:

"Content-Range: bytes=21-24"

with

"Content-Range: bytes 21-24/37"

(Note that the "=" is dropped)

Clause 9

- 1 Issue 14 - Clause 9.2.5, "move" row, description, replace:

"URI of an existing local or remote CDMI data container object (source URI) that shall be relocated, along with all child objects, to the URI specified in the PUT."

with

"URI of an existing local or remote CDMI container object (source URI) that shall be relocated, along with all child objects, to the URI specified in the PUT."

- 2 Issue 15 - Clause 9.2.7, "snapshots" row, description, replace:

"URI(s) of the SnapShot? container objects. See Clause 14."

with

"URI(s) of the snapshot container objects. See Clause 14."

- 3 Issue 17 - Clause 9.4.6, "snapshots" row, description, replace:

"URI(s) of the SnapShot? container objects"

with

"URI(s) of the snapshot container objects. See Clause 14."

4 Issue 19 - Clause 9.9.2, replace:

"• An optional percentComplete field contains the percentage that the Accepted PUT has completed (0 to 100)."

with

"• An optional percentComplete field contains the percentage that the Accepted POST has completed (0 to 100)."

5 Issue 20 - Clause 9.9.5, "move" row, description, replace:

"URI of a CDMI data object or queue that shall be copied into the new data object, and removed the data object or queue value at the source URI upon the successful completion of the copy"

with

"URI of a CDMI data object or queue object value that shall be copied into the new data object, and the data object or queue object value at the source URI shall be removed upon the successful completion of the copy"

6 Issue 20 - Clause 9.9.5, "valuetransferencoding" row, description, replace:

"• "utf-8" indicates that the container 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 container 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 container object value field to any value other than a valid base 64 string shall result in 400 Bad Request error being returned to the client."

with

"• "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 400 Bad Request error being returned to the client."

7 Issue 22 - Clause 9.11.2, replace:

"• An optional percentComplete field contains the percentage that the Accepted PUT has completed (0 to 100)."

with

"• An optional percentComplete field contains the percentage that the Accepted POST has completed (0 to 100)."

8 Issue 23 - Clause 9.11.9, Example, replace:

"POST to the container object URI the queue object name and metadata:"

with

"POST to the container object URI the queue object contents"

Clause 10

- 1 Issue 24 - Clause 10.1.2, Table 63, "cdmi_summary_bytesaverage" row, description, replace:
"The average number of bytes belonging to the domain during the summary time perio"
with
"The average number of bytes belonging to the domain during the summary time period"
- 2 Issue 26 - Clause 10.2.2, first bullet, replace:
"Support for the ability to create a new domain object is indicated by the presence of the "cdmi_create_domain " capability in the parent container."
with
"Support for the ability to create a new domain object is indicated by the presence of the "cdmi_create_domain" capability in the parent container."
(Removed extra space at end of "cdmi_create_domain".)
- 3 Issue 27 - Clause 10.2.4, "metadata" row, description, replace:
"Metadata for the domain. If this field is included when copying a domain object, the value provided shall be replace the metadata from the source URI. If this field is not specified, an empty JSON object (i.e., "{}") shall be assigned as the field value."
with
"Metadata for the domain object."
 - If this field is included when deserializing, serializing, copying, or moving a domain object, the value provided in this field shall replace the metadata from the source URI.
 - If this field is not included when deserializing, serializing, copying, or moving a domain object, the metadata from the source URI shall be used.
 - If this field is included when creating a new domain object by specifying a value, the value provided in this field shall be used as the metadata.
 - If this field is not included when creating a new domain object by specifying a value, an empty JSON object ("{}")(i.e., "{}") shall be assigned as the field value."
- 4 Issue 29 - Clause 10.2.8, Example response, replace:
"objectID" : "00007E7F00104BE66AB53A9572F9F51E/",
with
"objectID" : "00007E7F00104BE66AB53A9572F9F51E",
(Remove the trailing slash)
- 5 Issue 30 - Clause 10.3.3, Table 71, "Accept" row, Type, replace:
"Header StrinTg?"
with
"Header String"
- 6 Issue 31 - Clause 10.3.8, Example 1, replace:
"objectID" : "00007E7F00104BE66AB53A9572F9F51E/",

with
 "objectID" : "00007E7F00104BE66AB53A9572F9F51E",
 (Remove the trailing slash)

Clause 11

- 1 Issue 32 - Clause 11.2, Example response, replace:
 "queuevalues" : ""
 with
 "queueValues" : ""
- 2 Issue 33 - Clause 11.7.7, Table 101, "204 No Content" row, description, replace:
 "Queue object was successfully deleted."
 with
 "Queue object value was successfully deleted."

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #610 Clarify "Requirement" for object metadata

Description: When describing object metadata, such as in Table 62, we need to be clear about what is meant by "Requirement".

Proposed spec changes:

- 1 In tables 118, 119, 120, 121, 122, and 123, change the heading of the last column from "Requirement" to "Implementation".
- 2 In Table 62, change the table heading from "Table 62 - Domain Object Metadata" to "Table 62 - Required Metadata for a Domain Object"
- 3 In Table 120, change the table heading from "Table 120 - Required Data for a Notification Queue" to "Table 120 - Required Metadata for a Notification Queue"

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #611 Clarify capabilities and "shalls"

Description: We need to add spec text to the capabilities section that clarifies that "shall" only means "shall" if the corresponding capability is implemented by the system.

TWG: Approved the below text:

In Clause 12.1, following the existing text:

"Cloud clients may use capabilities to discover what operations are supported. If an operation is performed on a CDMI object that does not have a corresponding capability, an HTTP 400 status code shall be returned to the client. All CDMI-compliant

cloud storage systems shall implement the ability to list capabilities, but support for all other capabilities is optional."

Add the following paragraph:

"Requirements throughout this international standard shall be understood in the context of CDMI capabilities. Mandatory requirements on functionality that is conditioned on a CDMI capability shall not be interpreted to require implementation of that capability, but rather shall be interpreted to apply only to implementations that support the functionality required by that capability."

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #612 Fix "non-CDMI operation"/"cdmi operation" to be consistent

Description: In the spec, instances of "CDMI operation" should be changed to "CDMI content-type operation", and instances of "non-CDMI operations" should be changed to "non-CDMI content-type operations".

We may want to define and explain these before we use it in section 5.7.

TWG: Search and replace approved.

- Search and replace "CDMI operations" with "CDMI content-type operations".
- Search and replace "non-CDMI operations" with "non-CDMI content-type operations".

Dug to write proposal for clarifying this in section 5.7.

Proposal:

In section 5.7, after the first paragraph's bullets and before the paragraph that starts with "CDMI may also be used by administrative...."

add this paragraph:

This specification divides operations into two types: those that use a CDMI content type in the HTTP Body and those that do not. While much of the same data is available via both types, providing both allows for CDMI-aware clients and non-CDMI-aware clients to interact with a CDMI provider.

TWG: Approved.

Make both the global search and replaces, and apply Doug's change.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #613 Clarifications of objectName, parentURI and parentID description

Description: As per the TWG review of Ralph's changes at the face-to-face on Monday, the description text for objectName, parentURI and parentID needed further refinement.

Updated proposed text based on the resolution of ticket #349 is:**1** objectName

- JSON String
- Name of the object

For objects in a container, the objectName field shall be returned.

For objects not in a container (objects that are only accessible by ID), the objectName field shall not be returned.

- Mandatory

2 parentURI

- JSON String
- URI for the parent object.

For objects in a container, the parentURI field shall be returned.

For objects not in a container (objects that are only accessible by ID), the parentURI field shall not be returned.

Appending the objectName to the parentURI shall always produce a valid URI for the object.

- Mandatory

3 parentID

- JSON String
- Object ID of the parent container object

For objects in a container, the parentID field shall be returned.

For objects not in a container (objects that are only accessible by ID), the parentID field shall not be returned.

Appending the objectName to the parentURI shall always produce a valid URI for the object.

- Mandatory

TWG: Change approved.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #617 Scope (Section 20.1) should come before sections that reference it

Description: Sections 17, 19 and 20 reference section 20.1 Scope Specification. We should consider moving section 20.1 in front of these sections.

TWG: Approved the following changes:

- 1 Move section "18 Retention and Hold Management" before section "17 Logging".
- 2 Move section 20.1 Scope Specification to become a new section before section "17 Logging".
- 3 Add the following text to the beginning of the "Scope Specification" section:
"CDMI provides a standardized mechanism by which sets of objects matching certain characteristics can be defined. This mechanism is known as a CDMI Scope Specification. Scope specifications are typically used to provide a CDMI client with a way to indicate in what set of CDMI objects it is interested."
- 4 Move section 20.2 Results Specification to become a new section after section "Scope Specification".
- 5 Add the following text to the beginning of the "Results Specification" section:
"CDMI provides a standardized mechanism by which subsets of the contents of objects can be defined. This mechanism is known as a CDMI Results Specification. Result specifications are typically used to provide a CDMI client with a way to indicate on what subset of the contents of CDMI objects it intends to retrieve or operate."

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #619 Remove capabilities for creating containers and queues in domains

Description: The following capabilities entries should be removed from Table 107:

- cdm_create_container
- cdm_create_queue

TWG: Change approved.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #620 Editorial: URI equivalence != Object equivalence

Description: Section 5.10 says:

If the data object "MyDataObject?.txt" has an object ID of "00006FFD001001CCE3B2B4F602032653", the following pair of URIs are considered equivalent:

`http://cloud.example.com/root/MyDataObject.txt`

http://cloud.example.com/root/cdmi_objectid/
00006FFD001001CCE3B2B4F602032653

This isn't true. The URIs are not equivalent, they reference the same dataobject.

Proposal: Change it to say:

The following pair of URIs reference the same data object:

Same for the next paragraph about container objects.

1 In Section 5.10, replace:

"If the data object "MyDataObject??.txt" has an object ID of "00006FFD001001CCE3B2B4F602032653", the following pair of URIs are considered equivalent: "

with

"If the data object "MyDataObject??.txt" has an object ID of "00006FFD001001CCE3B2B4F602032653", the following pair of URIs access the same object:"

2 In Section 5.10, replace:

"If the container "MyContainer?" has an object ID of "00006FFD0010AA33D8CEF9711E0835CA", the following pairs of URIs are considered equivalent:"

with

"If the container "MyContainer?" has an object ID of "00006FFD0010AA33D8CEF9711E0835CA", the following pairs of URIs access the same object:"

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #622 **Missing Content-Type header isn't always a bad thing**

Description: In 5.13.2 it says:

If the content type of the requested resource is not present in the header, the server shall return a 406 Not Acceptable status code.

Not having a Content-Type header is ok as long as there is no HTTP body. For example, a 409 on a Delete w/o a CT header would be bad.

Proposal: Replace the above sentence with:

If a request message contains a Body but does not also include a Content-Type HTTP Header, the server shall return a 406 Not Acceptable status code.

In section 5.13.2, replace:

"If the content type of the requested resource is not present in the header, the server shall return a 406 Not Acceptable status code. See Section 12 of RFC 2616."

with

"If a request body is present, a Content-Type header shall be provided by the client. If a response body is present, a Content-Type header shall be provided by the server. If the client does not provide a Content-Type header when required, the server shall return a 406 Not Acceptable status code. See Section 12 of RFC 2616."

Note to Marie: The next sub-heading (Range Support) become normal text instead of a heading.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #623 Unnecessary restrictions on non-CDMI GET

Description: Right now, non-CDMI GETs of metadata are of the form:

```
GET <root URI>/<ContainerName>/<DataObjectName>?<fieldname>
```

But a CDMI get looks like:

```
GET <root URI>/<ContainerName>/
<DataObjectName>?<fieldname>;<fieldname>;...
```

Notice the CDMI GET can have multiple fieldnames. Its not clear why there is this extra restriction on the non-CDMI GET.

Proposal:

Allow for multiple fieldnames to appear on a non-CDMI GET. So, modify the Synopsis so it looks like the CDMI GET section.

TWG: Have agreed that we can add support to retrieve multiple fields.

Dug will review.

In section 8.5.1, replace:

```
GET <root URI>/<ContainerName>/<DataObjectName>?<fieldname>
```

with

```
GET <root URI>/<ContainerName>/<DataObjectName>?<fieldname>;<field-
name>;...
```

and replace:

```
GET <root URI>/<ContainerName>/<DataObjectName>?metadata:<prefix>
```

with

```
GET <root URI>/<ContainerName>/<DataObjectName>?metadata:<pre-
fix>?...

```

Note to Marie: Remember to remove the extra "?" that trac adds in.

CHANGES Made the specified changes to Version 1.0.1n.

Trac Ticket #624 Add new capability to indicate support for access by OID

Description: Add a new capability to indicate whether the provider supports the ability to access objects by ObjectID. This would be for all CRUD ops.

Proposed spec text change:

Add to the end of Table 102 - "System-Wide Capabilities"

cdmi_object_access_by_ID

JSON String

If present and "true", this capability indicates that objects can be accessed, updated and deleted through "/cdmi_objectid/"

TWG: Change approved.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #625 Example ACL JSON malformed

Description: The ACL JSON at the end of section "16.1.6 ACL Evaluation" is invalid.

TWG: Approved the below change to clause "16.1.6 ACL Evaluation"

Replace JSON with:

```
"cdmi_acl":
[
  {
    "acetype": "ALLOW",
    "identifier": "OWNER@",
    "aceflags": "OBJECT_INHERIT, CONTAINER_INHERIT",
    "acemask": "ALL_PERMS"
  },
  {
    "acetype": "ALLOW",
    "identifier": "AUTHENTICATED@",
    "aceflags": "OBJECT_INHERIT, CONTAINER_INHERIT",
    "acemask": "READ"
  }
]
```

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #627 Consistency - "Metadata Elements" -> "Metadata Items"

Description: We should use "Metadata Items" consistently throughout the spec to refer to JSON entities stored within the metadata field.

Proposed spec text change:

- 1 Global search and replace to change "Metadata Elements" to "Metadata Items"
- 2 Global search and replace to change "Metadata Element" to "Metadata Item"

TWG: Change approved.

CHANGES Made the specified changes to Version 1.0.1m.

Trac Ticket #628 Duplicate Terms Entry

Description: The entries for 3.12 and 3.13 are the same. I suspect this was a copy and paste error that overwrote the entry for 3.12. Can you investigate this?

CHANGES Removed duplicate entry in 1.0.1m; checked Terms against 1.0.1j; no terms missing.

Trac Ticket #632 Terms from SNIA Dictionary corrupted by ISO Review

Description: Some Terms borrowed from the SNIA Dictionary were somehow corrupted by the ISO review.

- 1 Change:
 - Data storage as a Service
 - DaaS
 - cloud storage
 - delivery of virtualized storage and data services on demand over a network (i.e., a non-technical term for DaaS)
 - NOTE Typically, DaaS hides limits to scalability, is either self-provisioned or provisionless, and is billed based on consumption.
 - To:
 - Data storage as a Service
 - DaaS
 - cloud storage
 - delivery of virtualized storage and data services on demand over a network, based on a request for a given service level
 - NOTE Typically, DaaS hides limits to scalability, is either self-provisioned or provisionless, and is billed based on consumption.

- 2** Change:
3.12
object identifier
an entity that has an object ID, a unique URI, and contains state
To:
object identifier (OID)
a globally unique value assigned at creation time to identify an object
- 3** Change:
3.16
private cloud
delivery of SaaS, PaaS, IaaS, and/or DaaS to a restricted set of customers, usually within a single organization
NOTE Private clouds are created to address security issues.
To:
3.16
private cloud
delivery of SaaS, PaaS, IaaS, and/or DaaS to a restricted set of customers, usually within a single organization
NOTE Private Clouds are created due to issues of trust.
- 4** Change:
3.17
public cloud
delivery of SaaS, PaaS, IaaS, and/or DaaS to an unrestricted set of customers as compared to a private cloud
To:
3.17
public cloud
delivery of SaaS, PaaS, IaaS and/or DaaS to a relatively unrestricted set of customers.
- 5** Change:
3.18
Representational State Transfer
REST
specific set of principles for defining, addressing, and interacting with resources addressable by URIs
NOTE Architectures that follow Representational State Transfer principles are said to be RESTful. The principles include abstraction of state into resources and a uniform set of representations and operations (e.g., HTTP verbs like GET and PUT as the only means to manipulate a resource). RESTful interfaces are contrasted with Web-Based Enterprise Management (WBEM), which tend to be RPC-like.
To:
3.18
Representational State Transfer

REST

specific set of principles for defining, addressing and interacting with resources addressable by URIs

NOTE Architectures that follow these principles are said to be RESTful. The principles include: abstraction of state into resources and a uniform set of representations and operations (e.g., HTTP verbs like GET and PUT as the only means to manipulate a resource). RESTful interfaces are contrasted with Web Services interfaces such as WBEM, which tend to be RPC-like.

CHANGES Made the specified changes in Version 1.0.1n.

Trac Ticket #635 **cdmi_owner should specify cdmi_member_principal rather than cdmi_member_name**

Description: In Table 117 - Storage System Metadata on Page 194, I think the definition of the cdmi_owner field should be changed from specifying a cdmi_member_name to specifying a cdmi_member_principal for the following reasons:

- ACLs are evaluated by principal rather than by user name.
- If the object is created anonymously, I don't think there is a well-defined cdmi_member_name, whereas there is an "ANONYMOUS@" principal
- Consistency with the definition of the cdmi_event_user field for Notifications, which is defined as specifying a principal

The following proposed spec text changes address the remaining issues:

Clause 16.3, Table 117, "cdmi_owner" row, replace:

"The cdmi_member_name of the principal that has owner privileges for the object."
with
"The name of the principal that has owner privileges for the object."

CHANGES Made the specified changes in Version 1.0.1n.

Trac Ticket #636 **Typos/errors in CDMI 1.0.1m draft**

Description: In addition to the corrections listed in ticket #609 (most of which are still applicable to 1.0.1m), a few more trivial errors found in the 1.0.1m draft:

The following proposed spec text changes address the remaining issues:

- 1 Issue 1 - Clause 6.5, Example response, replace:
"objectType" : "application/cdmi-object",
with
"objectType" : "application/cdmi-container",

- 2 Issue 2 - Clause 9.6.3, Table 42:
Drop the "Accept" row as there is no response body.
- 3 Issue 3 - Clause 12.1.1, Table 102, "cdmi_metadata_maxtotalsize" row:
The font size appears to be larger than other rows in the table.
- 4 Issue 4 - Clause 12.1.5, Table 106, "cdmi_move_dataobject" row:
The font size appears to be larger than other rows in the table.
- 5 Issue 4 - Clause 12.1.5, Table 106, "cdmi_copy_dataobject" row:
The font size appears to be larger than other rows in the table.
- 6 Issue 5 - Clause 15.3, second bullet, replace:
 - "• If the user who is deserializing a serialized object specifies a domainURI as part of the deserialize operation, the domainURI of every object being deserialized shall be set to the specified domainURI. To specify a domainURI other than the domainURI of the parent, the user shall have the cross-domain privilege. If the user does not have the cross-domain privilege and specifies a domainURI other than the domainURI of the parent, a 400 Bad Request response shall be returned."
 - with
 - "• If the user who is deserializing a serialized object specifies a domainURI as part of the deserialize operation, the domainURI of every object being deserialized shall be set to the specified domainURI. To specify a domainURI other than the domainURI of the parent, the user shall have the cross_domain privilege. If the user does not have the cross_domain privilege and specifies a domainURI other than the domainURI of the parent, a 400 Bad Request response shall be returned."
 - (Note extra space added at the end of the first sentence, and "cross-domain" changed to "cross_domain")
- 7 Issue 6 - Clause 18, example, replace:
 - "objectType" : "dataobject",
 - with
 - "objectType" : "application/cdmi-object",

CHANGES Made the specified changes in Version 1.0.1n.

Trac Ticket #637 **Table 115 - ACE Flags has extra row at end**

Description:

Table 115 - ACE Flags has a duplicated "No Flags" row. The extra one at the end of the table should be removed.

CHANGES Deleted duplicate row in Version 1.0.1n.

Trac Ticket #638 Table 116 - ACE Mask Bits is Truncated

Description:

Most of this table is missing in revision 1.0.1m.

CHANGES Added the missing rows "EXECUTE" through "SYNCHRONIZE" to Version 1.0.1n per original e-mail that included tables for Trac Ticket #586.

Trac Ticket #639 Add the trademark symbol to mentions of CDMI as appropriate

Description: Add the (TM) symbol to CDMI and include a "CDMI is a trademark of the Storage Networking Industry Association" statement to the appropriate place (use XAM(tm) spec as a guide).

CHANGES Made the specified changes in Version 1.0.1n.

Trac Ticket #640 Queue Enqueue returns incorrect HTTP status code

Description: The Queue Enqueue operation should return a 204 No Content instead of a 201 Created, since there is no resource created.

Proposed spec text:

- 1 In 11.6.7 Response Status, replace:
201 Created
New queue object value was enqueued.
with
204 No Content
New queue value(s) were enqueued.
- 2 In 11.6.8 Examples, Examples 1 through 5, in the response, replace:
HTTP/1.1 201 Created
with
HTTP/1.1 204 No Content

CHANGES Made the specified changes in Version 1.0.1n.

Trac Ticket #641 Add clarifying text around handling unknown fields

Description: The intent of the TWG is that unknown fields are preserved. This is required to handle vendor extensions and to preserve information specified in future versions of the spec.

Proposed spec text changes approved by the TWG:

In section 8.1, 9.1, 10.1, and 11.1, add the following sentence to the end of the section:

"When a client provides or includes as part of a deserialization fields that are not defined in this specification, these fields shall be stored as part of the object."

CHANGES Made the specified changes in Version 1.0.1n.