CDMI 1.0.1h 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 18.

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_encription 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 #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.
2. 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

1 Section 13.1 Extended Protocol Structure, second bullet - Change “identify” to “identity”. (Done)

2 Table 22; description for *cdmi_acl* - Remove the “(Based on NFSv4)” comment and adjust the description as necessary. (Done)

3 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.)

4 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.)

5 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.)

6 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|Manditory/Optional| (Done)

2 All JSON bodies should be

|Name|Type|Description|Manditory/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 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.

  - 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?>/<Queue-
Name?>?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?>/<Queue-
Name?>?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?>/<DataObject-
Name?>?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 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 *completion-Status* 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/cdmi.domain+json".
5. 11.3 - One change in 3rd Example Request. The “Accept” row in all other cases already has a value of "application/cdmi.queue+json"
6. 12.2 - No change. The "Accept" row already has a value of "application/cdmi.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 unncessary 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 41.

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 41.

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 41.

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 41.

Trac Ticket #244

Update Notification JSON to be consistent with Query

The Notification JSON as described in section 19, Table 21, needs to be updated to be consistent with the updated query specification 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 41.

Trac Ticket #245

ACL JSON Example has wrong name

Description: In section 16.1.11, the JSON and the example should specify "cdmi_acl" as opposed to ACL.

Specifically, change:

"ACL" : [

to

"cdmi_acl" : [

in both places.

We should also run the JSON example in this section through JSONLint to fix the indenting.

CHANGES Fixed code and ran example through JSONLint. Replaced old code with new code.

Trac Ticket #246

Table 16 Updates (Storage System Metadata)

Description: Table 16 (Storage System Metadata) should have the following updates applied:

- The "Permissions" column should be changed to "Mandatory/Optional"
- The value of every row for the "Mandatory/Optional" column should be "Optional".
- The value of every row of the "Type" column except for the cdmi_acl row should be "JSON String".
- The value of the "Type" column for the cdmi_acl row should be "JSON Array (see section 16.1.11)"

CHANGES Made the specified changes to Table 16 Storage System Metadata in Section 16.3 Support for Storage System Metadata.

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 arbitary 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 41.

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 meta-data 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 41.

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 41.
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 41.

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

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 sanititization

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-contigious 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 pre-

ferred 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 intrepret 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,ssssss). 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 time-

stamps 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-Indicatesthetimeoftheevent that triggered the notification. The time will be formatted in ISO-8601 time."

Replace with:

"cdmi_event_time-Indicatesthetimeoftheevent 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",
            "acetime" : 12345678
        },
        {
            "acetype" : "0xnn",
            "identifier" : "<user-or-group-name>",
            "aceflags" : "0xnn",
            "acemask" : "0xnn",
            "acetime" : 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/cdmi-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/cdmi-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/cdmi-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/cdmi-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 metdata 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.

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,ssssss", should be "YYYY- MM-DDThh:mm:ss.ssssss" (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 dop 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 #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" : {
        "OCCI/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.

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>/cdmi_domains/<DomainName>/<TheDomainName>
PUT <root URI>/cdmi_domains/<DomainName>/
   <TheDomainName>?metadata
```

to

```
PUT <root URI>/cdmi_domains/<DomainName>/<TheDomainName>/
PUT <root URI>/cdmi_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, capabiliesURI 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, capabiliesURI 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*, *capabiliesURI* and *domainURI* fields, all object IDs shall be processed such that they are case insensitive.

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/",
```

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.

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".

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 #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 “databoject” 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 105).