

1



2

# Capabilities Selection CDMI Extension

## *Version 2.0*

3

4

5 ABSTRACT: This CDMI Extension is intended for developers who are considering a standardized way to add  
6 functionality to CDMI. When multiple compatible implementations are demonstrated and approved by the Technical  
7 Working Group, this extension will be incorporated into the CDMI standard.

8 This document has been released and approved by the SNIA. The SNIA believes that the ideas, methodologies, and  
9 technologies described in this document accurately represent the SNIA goals and are appropriate for widespread  
10 distribution. Suggestion for revision should be directed to <http://www.snia.org/feedback/>.

11

SNIA Working Draft

12

June 17, 2020

### 13 USAGE

14 Copyright © 2020 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their  
15 respective owners.

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

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

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

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

25 Permission to use this document for purposes other than those enumerated above may be requested by emailing  
26 [tcmd@snia.org](mailto:tcmd@snia.org). Please include the identity of the requesting individual or company and a brief description of the pur-  
27 pose, nature, and scope of the requested use.

28 All code fragments, scripts, data tables, and sample code in this SNIA document are made available under the following  
29 license:

30 BSD 3-Clause Software License

31 Copyright (c) 2020, The Storage Networking Industry Association.

32 Redistribution and use in source and binary forms, with or without modification, are permitted provided that the following  
33 conditions are met:

34 \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.

35 \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following  
36 disclaimer in the documentation and/or other materials provided with the distribution.

37 \* Neither the name of The Storage Networking Industry Association (SNIA) nor the names of its contributors may be  
38 used to endorse or promote products derived from this software without specific prior written permission.

39 THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EX-  
40 PRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MER-  
41 CHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE  
42 COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL,  
43 EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUB-  
44 STITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER  
45 CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUD-  
46 ING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF  
47 ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

48 **DISCLAIMER**

49 The information contained in this publication is subject to change without notice. The SNIA makes no warranty of any  
50 kind with regard to this specification, including, but not limited to, the implied warranties of merchantability and fitness for  
51 a particular purpose. The SNIA shall not be liable for errors contained herein or for incidental or consequential damages  
52 in connection with the furnishing, performance, or use of this specification.

53 Suggestions for revisions should be directed to <https://www.snia.org/feedback/>.

54 Copyright © 2020 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their  
55 respective owners.

# Contents

56

57	<b>Clause 1: Capabilities Selection CDMI Extension</b>	<b>1</b>
58	1.1 Overview . . . . .	1
59	1.2 Instructions to the Editor . . . . .	1

## 60 Clause 1

# 61 Capabilities Selection CDMI Extension

## 62 1.1 Overview

63 The capabilities objects in `/<cdmi_root>/cdmi_capabilities/` describe sets of capabilities supported for stored  
 64 objects. This extension extends CDMI to permit capabilities objects to also contain data system metadata values that  
 65 act as default values and permit the capabilitiesURI to be specified on create or update, which allows an object to inherit  
 66 these data system metadata values.

## 67 1.2 Instructions to the Editor

68 To merge this extension into the CDMI 2.0.0 specification, make the following changes:

- 69 1. Add an entry immediately below the `domainURI` entry in the table starting on line 138 of  
 70 `cdmi_core/cdmi_data_object/create.txt`, as follows:

Table 1: Request message body - Create a data object using CDMI

Field Name	Type	Description	Requirement
<code>capabilitiesURI</code>	JSON string	<p>URI of a capabilities object. The newly created data object shall inherit data system metadata values from the specified capabilities object.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the create operation, the contents of the data system metadata items in the create operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the created data object may not match the specified <code>capabilitiesURI</code>, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 <code>Bad Request</code> shall be returned.</p>	Optional

- 71 2. Add the following example to the end of the examples section starting on line 394 of  
 72 `cdmi_core/cdmi_data_object/create.txt`, as follows:

73 EXAMPLE 8: PUT to the container URI to create a “gold” object:

```

--> PUT /cdmi/2.0.0/MyContainer/MyGoldObject.txt HTTP/1.1
--> Host: cloud.example.com
--> Accept: application/cdmi-object
--> Content-Type: application/cdmi-object
-->
--> {
-->   "mimetype" : "text/plain",
    
```

(continues on next page)

(continued from previous page)

```

--> "capabilitiesURI": "/cdmi_capabilities/dataobject/gold/",
--> "value" : "This is the Value of this Data Object"
--> }

<-- HTTP/1.1 201 Created
<-- Content-Type: application/cdmi-object
<--
<-- {
<-- "objectType" : "application/cdmi-object",
<-- "objectID" : "00007ED90010D891022876A8DE0BC0FD",
<-- "objectName" : "MyGoldObject.txt",
<-- "parentURI" : "/MyContainer/",
<-- "parentID" : "00007E7F00102E230ED82694DAA975D2",
<-- "domainURI" : "/cdmi_domains/MyDomain/",
<-- "capabilitiesURI" : "/cdmi_capabilities/dataobject/gold/",
<-- "completionStatus" : "Complete",
<-- "mimetype" : "text/plain",
<-- "metadata" : {
<--   "cdmi_size" : "37",
<--   <Data System Metadata Inherited from Gold Capabilities>
<-- }
<-- }

```

- 74 3. Add an entry immediately below the domainURI entry in the table starting on line 110 of  
 75 cdmi\_core/cdmi\_data\_object/update.txt, as follows:

Table 2: Request message body - Update a CDMI data object using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The updated data object shall inherit data system metadata values from the specified capabilities object, overwriting data system metadata values if already present.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the update operation, the contents of the data system metadata items in the update operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the updated data object may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 Bad Request shall be returned.</p>	Optional

- 76 4. Add the following examples to the end of the examples section starting on line 291 of  
 77 cdmi\_core/cdmi\_data\_object/update.txt, as follows:

78 EXAMPLE 13: PATCH to the data object URI to change a “gold” object to a “silver” object:

```

--> PATCH /cdmi/2.0.0/MyContainer/MyGoldObject.txt HTTP/1.1
--> Host: cloud.example.com
--> Content-Type: application/cdmi-object
-->
--> {
-->   "capabilitiesURI": "/cdmi_capabilities/dataobject/silver/"
--> }

<-- HTTP/1.1 204 No Content

```

79 EXAMPLE 14: PATCH to the data object URI to change a “gold” object to a “silver” object, including a manually specified  
 80 data system metadata item value (also required to preserve any data system metadata item values that were previously  
 81 specified manually):

```

--> PATCH /cdmi/2.0.0/MyContainer/MyGoldObject.txt HTTP/1.1
--> Host: cloud.example.com

```

(continues on next page)

(continued from previous page)

```

--> Content-Type: application/cdmi-object
-->
--> {
-->   "capabilitiesURI": "/cdmi_capabilities/dataobject/silver/",
-->   "metadata": {
-->     "cdmi_data_redundancy": "3"
-->   }
--> }
<-- HTTP/1.1 204 No Content
    
```

5. Add an entry immediately below the domainURI entry in the table starting on line 112 of cdmi\_core/cdmi\_container\_object/create.txt, as follows:

Table 3: Request message body - Create a container object using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The newly created container object shall inherit data system metadata values from the specified capabilities object.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the create operation, the contents of the data system metadata items in the create operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the created container object may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code 400 Bad Request shall be returned.</p>	Optional

6. Add the following example to the end of the examples section starting on line 425 of cdmi\_core/cdmi\_container\_object/create.txt, as follows:

EXAMPLE 5: PUT to the container URI to create a "gold" container:

```

--> PUT /cdmi/2.0.0/MyContainer/ HTTP/1.1
--> Host: cloud.example.com
--> Accept: application/cdmi-container
--> Content-Type: application/cdmi-container
-->
--> {
-->   "capabilitiesURI" : "/cdmi_capabilities/container/gold/"
--> }

<-- HTTP/1.1 201 Created
<-- Content-Type: application/cdmi-container
<--
<-- {
<--   "objectType" : "application/cdmi-container",
<--   "objectID" : "00007ED900104E1D14771DC67C27BF8B",
<--   "objectName" : "MyContainer/",
<--   "parentURI" : "/",
<--   "parentID" : "00007E7F0010128E42D87EE34F5A6560",
<--   "domainURI" : "/cdmi_domains/MyDomain/",
<--   "capabilitiesURI" : "/cdmi_capabilities/container/gold/",
<--   "completionStatus" : "Complete",
<--   "metadata" : {
<--     <Data System Metadata Inherited from Gold Capabilities>
<--     ...
<--   },
<--   "childrenrange": "",
<--   "children": []
<-- }
    
```

7. Add an entry immediately below the domainURI entry in the table starting on line 111 of cdmi\_core/cdmi\_container\_object/update.txt, as follows:

Table 4: Request message body - Update a container object using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The updated container object shall inherit data system metadata values from the specified capabilities object, overwriting data system metadata values if already present.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the update operation, the contents of the data system metadata items in the update operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the updated container object may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 <i>Bad Request</i> shall be returned.</p>	Optional

89 8. Add the following example to the end of the examples section starting on line 275 of  
 90 cdm\_core/cdm\_container\_object/update.txt, as follows:

91 EXAMPLE 3: PATCH to the container object URI to change a “gold” container to a “silver” container:

```

--> PATCH /cdmi/2.0.0/MyContainer/ HTTP/1.1
--> Host: cloud.example.com
--> Content-Type: application/cdm-container
-->
--> {
-->   "capabilitiesURI" : "/cdmi_capabilities/container/silver/"
--> }
<-- HTTP/1.1 204 No Content
    
```

92 9. Add an entry immediately below the domainURI entry in the table starting on line 158 of  
 93 cdm\_core/cdm\_container\_object/create\_post\_object.txt, as follows:

Table 5: Request message body - Create a new data object Using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The newly created data object shall inherit data system metadata values from the specified capabilities object.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the create operation, the contents of the data system metadata items in the create operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the created data object may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 <i>Bad Request</i> shall be returned.</p>	Optional

94 10. Add an entry immediately below the domainURI entry in the table starting on line 146 of  
 95 cdm\_core/cdm\_container\_object/create\_post\_queue.txt, as follows:



Table 6: Request message body - Create a new queue object using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The newly created queue object shall inherit data system metadata values from the specified capabilities object.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the create operation, the contents of the data system metadata items in the create operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the created queue object may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 <i>Bad Request</i> shall be returned.</p>	Optional

96 11. Add an entry immediately below the domainURI entry in the table starting on line 265 of  
97 cdm\_advanced/cdm\_queue\_object.txt, as follows:

Table 7: Request message body - Create a queue object using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The newly created queue object shall inherit data system metadata values from the specified capabilities object.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the create operation, the contents of the data system metadata items in the create operation metadata field shall take precedence.</p> <p>The capabilitiesURI of the created queue object may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 <i>Bad Request</i> shall be returned.</p>	Optional

98 12. Add an entry immediately below the domainURI entry in the table starting on line 1087 of  
99 cdm\_advanced/cdm\_queue\_object.txt, as follows:

Table 8: Request message body - Update a queue object Using CDMI

Field Name	Type	Description	Requirement
capabilitiesURI	JSON string	<p>URI of a capabilities object. The updated queue object shall inherit data system metadata values from the specified capabilities object, overwriting data system metadata values if already present.</p> <p>If data system metadata items are specified in the capabilities object and in the metadata field included with the update operation, the contents of the data system metadata items in the update operation metadata field shall take precedence</p> <p>The capabilitiesURI of the updated queue may not match the specified capabilitiesURI, depending on the object type, specified data system metadata items, and other system configurations.</p> <p>If an invalid URI is specified, an HTTP status code of 400 <i>Bad Request</i> shall be returned.</p>	Optional

100 13. Replace the below text starting on line 1087 of cdm\_advanced/cdm\_capability\_object.txt, as follows:

101 Capabilities cannot be altered by clients, but may be changed by the CDMI server to reflect configuration changes or  
102 operational changes. For example, if a CDMI server is upgraded or reconfigured, additional capabilities may become  
103 present, or existing capabilities may no longer be present. In practice, capabilities rarely change, and a client can assume

104 that they shall remain constant for the duration of a client-server HTTP/HTTPS session.

105 Replace with:

106 Capabilities cannot be altered by clients, but may be created or changed by the CDMI server to reflect configuration  
 107 changes or operational changes. For example, if a CDMI server is upgraded or reconfigured, additional capabilities may  
 108 become present, or existing capabilities may no longer be present. In practice, capabilities rarely change, and a client  
 109 can assume that they shall remain constant for the duration of a client-server HTTP/HTTPS session.

110 A CDMI implementation may map objects to specific capabilities URIs (such as the “gold\_container” capabilities URI  
 111 shown above) when data system metadata fields are present in the created or updated object and match to the data  
 112 system metadata fields and values in a given capabilities object.

113 This mapping permits CDMI servers to create templates that can be used when creating or updating an object to specify a  
 114 default set of data system metadata values. For example, the “gold\_container” could include a `cdmi_data_redundancy`  
 115 metadata item with the value set to “4”. Any container created with this capabilities object specified shall inherit a  
 116 “`cdmi_data_redundancy`” metadata item with the value “4”, and, in the absence of additional and/or overriding data  
 117 system metadata values, shall have a `capabilitiesURI` set to the “gold\_container”.

118 14. Add an entry to the end of the table starting on line 135 of `cdmi_advanced/cdmi_capability_object.txt`, as follows:

Table 9: System-wide capabilities

Capability name	Type	Definition
<code>cdmi_capabilities_templates</code>	JSON string	If present and “true”, the CDMI server supports capabilities objects being used as templates when creating or modifying a CDMI object.
<code>cdmi_capabilities_exact_</code> ↔ <code>inherit</code>	JSON string	If present and “true”, the CDMI server supports inheriting data system metadata when a capabilities object is specified on create or update, which completely replace all existing data system metadata not explicitly specified in the object creation or update.

119 15. Add an entry immediately below the `capabilities` entry in the table starting on line 1015 of  
 120 `cdmi_advanced/cdmi_capability_object.txt`, as follows:

Table 10: Response message body - Read a capabilities object using CDMI

Field name	Type	Description	Requirement
<code>metadata</code>	JSON object	Metadata for the capabilities object. This field includes data system metadata associated with the capabilities object. See Clause 16 for a further description of data system metadata.	Mandatory

121 16. Update example 1 in the examples section starting on line 1113 of `cdmi_advanced/cdmi_capability_object.txt`, as  
 122 follows:

123 EXAMPLE 1: GET to the root container capabilities URI to read all fields of the container:

```

--> GET /cdmi/2.0.0/cdmi_capabilities/ HTTP/1.1
--> Host: cloud.example.com
--> Accept: application/cdmi-capability

<-- HTTP/1.1 200 OK
<-- Content-Type: application/cdmi-capability
<--
<-- {
<--   "objectType": "application/cdmi-capability",
<--   "objectID": "00007E7F00104BE66AB53A9572F9F51E",
<--   "objectName": "cdmi_capabilities/",
<--   "parentURI": "/",
<--   "parentID": "00007E7F0010128E42D87EE34F5A6560",
<--   "capabilities": {
<--     "cdmi_domains": "true",
<--     "cdmi_export_nfs": "true",
<--     "cdmi_export_iscsi": "true",
<--     "cdmi_queues": "true",
<--     "cdmi_notification": "true",
<--     "cdmi_query": "true",
<--     "cdmi_metadata_maxsize": "4096",
    
```

(continues on next page)

(continued from previous page)

```
<--      "cdmi_metadata_maxitems": "1024"
<--    },
<--    "metadata" : {
<--      <Data System Metadata for the capability object>
<--      ...
<--    },
<--    "childrenrange": "0-3",
<--    "children": [
<--      "domain/",
<--      "container/",
<--      "dataobject/",
<--      "queue/"
<--    ]
<--  }
```