



CONFORMANCE TESTING PROGRAM

CLOUD DATA MANAGEMENT INTERFACE (CDMI) TEST
SPECIFICATION

VERSION 1.0a

WORKING DRAFT

APRIL 9, 2014

SNIA Notice

This document contains a test specification that is based on the Storage Networking Industry Association (SNIA) Cloud Storage international standard (CDMI 1.01 and 1.02) provided by the SNIA Cloud Storage Technical Working Group (TWG).

This draft represents a "best effort" attempt by the Conformance Testing working group to reach preliminary consensus, and it may be updated, replaced, or deemed obsolete at any time. This document should not be used as reference material.

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

USAGE

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

1. Any text, diagram, chart, table or definition reproduced must be reproduced in its entirety with no alteration, and,
2. Any document, printed or electronic, in which material from this document (or any portion hereof) is reproduced must acknowledge the SNIA copyright on that material, and must credit the SNIA for granting permission for its reuse.

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

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

DISCLAIMER

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

Copyright © 2014 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.

Revision History

Version	Date	Description of Changes	Author
1.0		First draft of 1.0 Test Specification based on the CDMI 1.0.2 Test Spec provided by SNIA.	Ankit Agarwal
4.0	4/8/14	Performed technical edit of version 3.5. Reformatted document for consistency and proofed for errors. Added front matter per SNIA drafting standards and added DRAFT watermark. Made changes as specified in Trac Tickets 938, 940-943, 945, and 946.	Marie McMinn
4.1	4/9/14	Added the following: <ul style="list-style-type: none"> • Create DataObject via Non-CDMI, read via CDMI: (6.4.2.14 CDMI_DataObject_Read_19) • Create DataObject via Non-CDMI, delete via CDMI: (6.4.4.5 CDMI_DataObject_Delete_6) • Create DataObject via CDMI, read via Non-CDMI: (7.4.2.9 DataObject_Read_NC_2R_10) • Create DataObject via CDMI, delete via Non-CDMI: (7.4.4.4 DataObject_Delete_NC_2D_4) 	Sachin Goswami
4.2	4/10/14	Performed consistency edits & fixed auto-numbering	Marie McMinn
1.0a	4/17/14	Changed internal version number to v1.0a for its first public Working Draft release	Arnold Jones

Table of Contents

1	Background and Overview	1
2	Scope	1
3	References	1
4	Rules to Follow	1
4.1	Conformance Test Process	1
4.2	Test Specification Definitions and Details	1
5	Architecture and Design of the CDMI Test Specification	2
6	CDMI Support	2
6.1	Server Authentication Test.....	2
6.1.1	<i>Server Authentication Success (1C_L_1)</i>	2
6.1.1.1	Basic Authentication (1C_L_1.1).....	2
6.1.1.2	Digest Authentication (1C_L_1.2).....	3
6.1.2	<i>Server Authorization failure (1C_L_2)</i>	5
6.1.2.1	Basic Authentication (1C_L_2.1).....	5
6.1.2.2	Digest Authentication (1C_L_2.2):.....	5
6.2	Capability.....	7
6.2.1	<i>CDMI_Capability_Read</i>	7
6.2.1.1	CDMI_Capability_Read_3R_1.....	7
6.2.1.2	CDMI_Capability_Read_3R_2.....	8
6.2.1.3	CDMI_Capability_Read_3R_3.....	8
6.2.1.4	CDMI_Capability_Read_3R_4.....	9
6.2.1.5	CDMI_Capability_Read_3R_5.....	10
6.3	Container.....	11
6.3.1	<i>CDMI_Container_Create</i>	11
6.3.1.1	CDMI_Container_Create_1C_1.....	11
6.3.1.2	CDMI_Container_Create_1C_3.....	12
6.3.1.3	CDMI_Container_Create_1C_4.....	13
6.3.1.4	CDMI_Container_Create_1C_5.....	13
6.3.1.4.1	<i>CDMI_Container_Create_1C_5.1</i>	13
6.3.1.4.2	<i>CDMI_Container_Create_1C_5.2</i>	14
6.3.1.5	CDMI_Container_Create_1C_9.....	15
6.3.1.6	CDMI_Container_Create_1C_10	16
6.3.1.7	CDMI_Container_Create_1C_11	16
6.3.1.8	CDMI_Container_Create_1C_12	18
6.3.2	<i>CDMI_Container_Read</i>	19
6.3.2.1	CDMI_Container_Read_1R_1.....	19
6.3.2.2	CDMI_Container_Read_1R_2.....	20
6.3.2.3	CDMI_Container_Read_1R_4.....	20
6.3.2.4	CDMI_Container_Read_1R_6.....	21

6.3.2.5	CDMI_Container_Read_1R_7	22
6.3.2.6	CDMI_Container_Read_1R_9	22
6.3.2.7	CDMI_Container_Read_1R_10	23
6.3.2.7.1	CDMI_Container_Read_1R_10.1	23
6.3.2.7.2	CDMI_Container_Read_1R_10.2	24
6.3.2.7.3	CDMI_Container_Read_1R_10.4	25
6.3.2.7.4	CDMI_Container_Read_1R_10.5	26
6.3.2.8	CDMI_Container_Read_1R_11	27
6.3.2.8.1	CDMI_Container_Read_1R_11.1	27
6.3.2.8.2	CDMI_Container_Read_1R_11.2	28
6.3.2.8.3	CDMI_Container_Read_1R_11.4	29
6.3.2.8.4	CDMI_Container_Read_1R_11.5	30
6.3.2.9	CDMI_Container_Read_1R_12	30
6.3.3	CDMI_Container_Update	31
6.3.3.1	CDMI_Container_Update_1U_1	31
6.3.3.2	CDMI_Container_Update_1U_2	32
6.3.3.3	CDMI_Container_Update_1U_5	33
6.3.3.4	CDMI_Container_Update_1U_6	34
6.3.3.5	CDMI_Container_Update_1U_7	34
6.3.3.6	CDMI_Container_Update_1U_8	35
6.3.3.7	CDMI_Container_Update_1U_9	36
6.3.3.7.1	CDMI_Container_Update_1U_9.1	36
6.3.3.8	CDMI_Container_Update_1U_12	37
6.3.3.9	CDMI_Container_Update_1U_13	38
6.3.4	CDMI_Container_Delete	39
6.3.4.1	CDMI_Container_Delete_1D_1	39
6.3.4.2	CDMI_Container_Delete_1D_2	40
6.3.4.3	CDMI_Container_Delete_1D_3	41
6.3.4.4	CDMI_Container_Delete_1D_4	41
6.4	Data Object	42
6.4.1	CDMI_DataObject_Create	42
6.4.1.1	CDMI_DataObject_Create_2C_1	42
6.4.1.2	CDMI_DataObject_Create_2C_3	43
6.4.1.3	CDMI_DataObject_Create_2C_6	44
6.4.1.4	CDMI_DataObject_Create_2C_7	45
6.4.1.5	CDMI_DataObject_Create_2C_8	46
6.4.1.6	CDMI_DataObject_Create_2C_9	47
6.4.1.7	CDMI_DataObject_Create_2C_10	48
6.4.1.8	CDMI_DataObject_Create_2C_11	49
6.4.1.9	CDMI_DataObject_Create_2C_12	50
6.4.1.10	CDMI_DataObject_Create_2C_14	50
6.4.1.11	CDMI_DataObject_Create_2C_15	52
6.4.1.12	CDMI_DataObject_Create_2C_16	53
6.4.1.13	CDMI_DataObject_Create_2C_17	54
6.4.1.14	CDMI_DataObject_Create_2C_18	54

6.4.2	<i>CDMI_DataObject_Read</i>	55
6.4.2.1	CDMI_DataObject_Read_1.....	55
6.4.2.2	CDMI_DataObject_Read_2.....	56
6.4.2.2.1	<i>CDMI_DataObject_Read_2.1</i>	56
6.4.2.2.2	<i>CDMI_DataObject_Read_2.2</i>	57
6.4.2.3	CDMI_DataObject_Read_4.....	58
6.4.2.4	CDMI_DataObject_Read_5.....	58
6.4.2.5	CDMI_DataObject_Read_6.....	59
6.4.2.5.1	<i>CDMI_DataObject_Read_6.1</i>	59
6.4.2.5.2	<i>CDMI_DataObject_Read_6.2</i>	60
6.4.2.6	CDMI_DataObject_Read_7.....	61
6.4.2.7	CDMI_DataObject_Read_9.....	62
6.4.2.8	CDMI_DataObject_Read_10.....	62
6.4.2.9	CDMI_DataObject_Read_11.....	63
6.4.2.10	CDMI_DataObject_Read_12.....	64
6.4.2.11	CDMI_DataObject_Read_13.....	65
6.4.2.12	CDMI_DataObject_Read_16.....	66
6.4.2.13	CDMI_DataObject_Read_18.....	67
6.4.2.14	CDMI_DataObject_Read_19.....	67
6.4.3	<i>CDMI_DataObject_Update</i>	68
6.4.3.1	CDMI_DataObject_Update_1.....	68
6.4.3.2	CDMI_DataObject_Update_2.....	69
6.4.3.2.1	<i>CDMI_DataObject_Update_2.1</i>	69
6.4.3.2.2	<i>CDMI_DataObject_Update_2.2</i>	70
6.4.3.2.3	<i>CDMI_DataObject_Update_2.3</i>	71
6.4.3.2.4	<i>CDMI_DataObject_Update_2.4</i>	71
6.4.3.2.5	<i>CDMI_DataObject_Update_2.5</i>	72
6.4.3.3	CDMI_DataObject_Update_4.....	73
6.4.3.4	CDMI_DataObject_Update_5.....	74
6.4.3.5	CDMI_DataObject_Update_6.....	74
6.4.3.6	CDMI_DataObject_Update_7.....	75
6.4.4	<i>CDMI_DataObject_Delete</i>	76
6.4.4.1	CDMI_DataObject_Delete_1.....	76
6.4.4.2	CDMI_DataObject_Delete_2.....	77
6.4.4.3	CDMI_DataObject_Delete_3.....	78
6.4.4.4	CDMI_DataObject_Delete_5.....	78
6.4.4.5	CDMI_DataObject_Delete_6.....	79
6.5	Server Cleanup Process.....	80
6.5.1	<i>Container Cleanup</i>	80
6.5.2	<i>Data Object Cleanup</i>	80
7	Non-CDMI Support	81
7.1	Server Authentication Test.....	81
7.2	Capability.....	81
7.3	Non-CDMI Container.....	81

7.3.1	<i>Non_CDMI_Container_Create</i>	81
7.3.1.1	Container_Create_NC_1C_1.....	81
7.3.1.2	Container_Create_NC_1C_2.....	82
7.3.1.3	Container_Create_NC_1C_3.....	82
7.3.1.3.1	Container_Create_NC_1C_3.1.....	82
7.3.2	<i>Non_CDMI_Container_Delete</i>	83
7.3.2.1	Container_Delete_NC_1D_1	83
7.3.2.2	Container_Delete_NC_1D_2	84
7.3.2.3	Container_Delete_NC_1D_4	84
7.3.2.4	Container_Delete_NC_1D_5	85
7.3.2.5	Container_Delete_NC_1D_6	86
7.3.2.5.1	Container_Delete_NC_1D_6.1.....	86
7.3.2.5.2	Container_Delete_NC_1D_6.2.....	86
7.3.2.5.3	Container_Delete_NC_1D_6.3.....	87
7.3.2.6	Container_Delete_NC_1D_7	87
7.3.2.7	Container_Delete_NC_1D_8	88
7.4	<i>Non_CDMI_Data_Object</i>	89
7.4.1	<i>Non_CDMI_DataObject_Create</i>	89
7.4.1.1	DataObject_Create_NC_2C_1	89
7.4.1.2	DataObject_Create_NC_2C_2	90
7.4.1.3	DataObject_Create_NC_2C_4	90
7.4.1.4	DataObject_Create_NC_2C_5	91
7.4.1.5	DataObject_Create_NC_2C_7	92
7.4.2	<i>Non_CDMI_DataObject_Read</i>	93
7.4.2.1	DataObject_Read_NC_2R_1.....	93
7.4.2.2	DataObject_Read_NC_2R_2.....	93
7.4.2.3	DataObject_Read_NC_2R_3.....	94
7.4.2.4	DataObject_Read_NC_2R_5.....	95
7.4.2.5	DataObject_Read_NC_2R_6.....	95
7.4.2.6	DataObject_Read_NC_2R_7.....	96
7.4.2.7	DataObject_Read_NC_2R_9.....	97
7.4.2.8	DataObject_Read_NC_2R_10.....	97
7.4.3	<i>Non_CDMI_DataObject_Update</i>	98
7.4.3.1	DataObject_Update_NC_2U_1.....	98
7.4.3.2	DataObject_Update_NC_2U_2.....	99
7.4.3.3	DataObject_Update_NC_2U_3.....	100
7.4.3.4	DataObject_Update_NC_2U_4.....	100
7.4.3.5	DataObject_Update_NC_2U_7.....	101
7.4.4	<i>Non_CDMI_DataObject_Delete</i>	102
7.4.4.1	DataObject_Delete_NC_2D_1	102
7.4.4.2	DataObject_Delete_NC_2D_2	103
7.4.4.3	DataObject_Delete_NC_2D_3	104
7.4.4.4	DataObject_Delete_NC_2D_4	104
7.5	<i>Server Cleanup Process</i>	105

7.5.1	<i>Non-CDMI Container Cleanup</i>	105
7.5.2	<i>Non-CDMI Data Object Cleanup</i>	105
8	Profile-Based Support	106
8.1	Basic Service-Based Profile.....	106
8.1.1	<i>Mandatory Tests</i>	106
8.1.2	<i>Optional Tests</i>	106
8.2	Self -Storage Management Profile	107
8.2.1	<i>Mandatory Tests</i>	107
8.2.2	<i>Optional Tests</i>	107
9	Test Standards Conventions	108

DRAFT

1 Background and Overview

SNIA has introduced the Cloud Data Management Interface (CDMI) international standard. By adopting this international standard, vendors gain greater market acceptance than they do from adopting proprietary interfaces, while end users of products complying with the CDMI international standard gain greater interoperability between cloud storage providers.

2 Scope

In this test specification document, we define the overall test scope and test specifications for the Storage Network Industry Association (SNIA) to test the conformance to the CDMI specifications.

This document plays a key role in creating conformance certification test suites. The fundamental purpose of this specification is as follows:

- To observe the overall scope of the test specification and the total number of test specifications
- To observe the test coverage achieved over the SNIA standard that a vendor product addresses
- To identify the various components and test the configurations that are used
- To explain and understand each scenario to be tested and identify the requirements it covers

3 References

References are based on the SNIA-provided international standard CDMI 1.0.2. This document is listed for reference.

CDMI Specification - <http://cdmi.sniacloud.com/>

4 Rules to Follow

4.1 Conformance Test Process

The goal of conformance testing is to ensure that the target product is implemented according to the CDMI international standard and complies with the mandatory features and specific profiles that are specified in this international standard. In addition, optional features may be checked when all capabilities are available on the CDMI server.

4.2 Test Specification Definitions and Details

The test descriptions are provided in a test specification format, which contains the following:

- Header with the test case description
- Test specification ID
- Test case name
- Test case description

- Pre-test dependencies, which define conditions to apply before starting the test
- Description, which identifies the test case execution steps, expected results, verification header fields, expected values, and messages provided in the CDMI international standard

5 Architecture and Design of the CDMI Test Specification

6 CDMI Support

The SNIA CDMI international standard is intended for application developers who are implementing or using cloud storage. It documents how to access cloud storage and to manage the data stored there.

6.1 Server Authentication Test

A CDMI server shall implement basic only, digest only, or both. Which authorization is supported will be indicated by the authentication challenge returned when an operation is attempted without authentication.

These test cases will perform digest or basic authentication according to a WWW-Authenticate response. Digest or basic authentication test cases should run depending on the implementation.

6.1.1 Server Authentication Success (IC_L_1)

6.1.1.1 Basic Authentication (1C_L_1.1)

CDMI Container Create Logging Credentials Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_L_1.1
Test Case Name	LoginAuthentication_Success
Test Case Description	To create a container with correct credentials
Pre-Test Dependencies	This test case will run if CDMI server implements Basic (WWW- Authenticate).

D E S C R I P T I O N

<Test Case : Start>

1. Set valid credentials in Request.
 UserName : <Correct user name>
 Password : <Correct password>
2. Set Basic Authorization header:
 Authorization: "Basic " + Base64(Username:Password)
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.2.1>
 Request Header: <Reference: CDMI Specification v1.0.2 – Section: 9.2.4>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: 9.2.5>
 Request Sample with Body:

```

PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2

{
}

```
4. Check for HTTP status code: **201 Created** returned.
 <Reference :CDMI Specification v1.0.2 – Section: 9.2.8>
5. Container created.
6. Container cleanup process : <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.1.1.2 Digest Authentication (1C_L_1.2)

CDMI Container Create Logging Credentials Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_L_1.2
Test Case Name	containerCreate_LoginAuthentication_Success
Test Case Description	To create a container with correct credentials
Pre-Test Dependencies	This test case will run if CDMI server implements Digest (WWW- Authenticate).

DESCRIPTION**<Test Case : Start>**

1. Set valid credentials in Request.

UserName : <Correct user name>

Password : <Correct password>

2. Set Digest Authorization header:

read "realm", "nonce", "qop" and "opaque" from Server Response Header
"WWW-Authenticate"

nc= 00000001

cnonce="0a4f113b"

String HA1 = MD5(UserName:realm:Password)

String HA2 = MD5(Method:URI)

String responseStr = MD5(HA1:nonce:nc:cnonce:qop:HA2)

Authorization: Digest username= UserName,
realm=realm,

nonce=nonce,

uri="<root URI>/TestContainer1/",

qop=qop,

nc=nc,

cnonce=cnonce,

response=responseStr,

opaque=opaque

3. Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.1>**
Request Header: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.4>**
Request Body: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.5>**

Request Sample with Body:

PUT <root URI>/TestContainer1/ HTTP/1.1

Host: cloud.example.com

Accept: application/cdmi-container

Content-Type: application/cdmi-container

X-CDMI-Specification-Version: 1.0.2

```
{
}
```

4. Check for HTTP status code: **201 Created** returned.
<Reference :CDMI Specification v1.0.2 – Section: 9.2.8>
5. Container created.
6. Container cleanup process : **<Reference: 6.5.1 Container Cleanup>**

<Test Case : End>

6.1.2 Server Authorization failure (1C_L_2)

6.1.2.1 Basic Authentication (1C_L_2.1)

CDMI Container Create Logging Credentials Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_L_2.1
Test Case Name	containerCreate_LoginAuthentication_Fail
Test Case Description	To not permit creation of a container when supplied with invalid credentials
Pre-Test Dependencies	This test case will run if CDMI server implements basic (WWW- Authenticate).

DESCRIPTION

<Test Case : Start>

1. Set invalid credentials in Request.
 UserName : <invalid username>
 Password : <invalid password>
2. Set Basic Authorization header:
 Authorization: "Basic " + Base64(UserName:Password) + "!"
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **9.2.1**>
 Request Header: <Reference: CDMI Specification v1.0.2 – Section: **9.2.4**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **9.2.5**>
 Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2

{
}
```
4. Check for HTTP status code: **401 Unauthorized** returned.
 <Reference: CDMI Specification v1.0.2 – Section: **9.2.8**>
5. Container creation failed.

<Test Case : End>

6.1.2.2 Digest Authentication (1C_L_2.2):

CDMI Container Create Logging Credentials Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_L_2.2
Test Case Name	containerCreate_LoginAuthentication_Fail

Test Case Description	To not permit creation of a container when supplied with invalid credentials
Pre-Test Dependencies	This test case will run if CDMI server implements Digest (WWW- Authenticate).
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set invalid credentials in Request. UserName : <invalid user name> Password : <invalid password> 2. Set Digest Authorization header: read "realm", "nonce", "qop" and "opaque" from Server Response Header "WWW-Authenticate" nc= 00000001 cnonce="0a4f113b" String HA1 = MD5(UserName:realm:Password) String HA2 = MD5(Method:URI) String responseStr = MD5(HA1:nonce:nc:cnonce:qop:HA2) Authorization: Digest username= UserName, realm=realm, nonce=nonce, uri="<root URI>/TestContainer1/", qop=qop, nc=nc, cnonce=cnonce, response=responseStr, opaque=opaque 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.2.1> Request Header: <Reference: CDMI Specification v1.0.2 – Section: 9.2.4> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 9.2.5> Request Sample with Body: PUT <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-container Content-Type: application/cdmi-container X-CDMI-Specification-Version: 1.0.2 { } 4. Check for HTTP status code: 401 Unauthorized returned. <Reference :CDMI Specification v1.0.2 – Section: 9.2.8> 5. Container creation failed. <p><Test Case : End></p>	

6.2 Capability

6.2.1 CDMI_Capability_Read

6.2.1.1 CDMI_Capability_Read_3R_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 12.2
Test Specification Id	3R_1
Test Case Name	capabilityRead_3R_1
Test Case Description	To read system-wide capabilities
Pre-Test Dependencies	None
<i>D E S C R I P T I O N</i>	
<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 12.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 12.2.3> Request Sample: GET <root URI>/cdmi_capabilities/ HTTP/1.1 Host: cloud.example.com Content-Type: application/cdmi-capability X-CDMI-Specification-Version: 1.0.2 Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 12.2.6> <ul style="list-style-type: none"> – objectType – objectID – objectName – parentURI – parentID – capabilities – childrenrange – children Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 12.2.7> Expected Result: System-wide Capabilities should be listed. <p><Test Case : End></p>	

6.2.1.2 CDMI_Capability_Read_3R_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 12.2
Test Specification Id	3R_2
Test Case Name	capabilityRead_3R_2
Test Case Description	To read a capability object.
Pre-Test Dependencies	None

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference:** Login Test Spec Id : **1C_L_1>**
2. Request URL: **<Reference:** CDMI Specification v1.0.2 – Section: 12.2.1>
 Request Headers: **<Reference:** CDMI Specification v1.0.2 – Section: 12.2.3>
 Request Sample:

```
GET <root URI>/cdmi_capabilities/<child capability>/ HTTP/1.1
Host: cloud.example.com
Content-Type: application/cdmi-capability
X-CDMI-Specification-Version: 1.0.2
```
3. Verify that all mandatory fields exist:
<Reference: CDMI Specification v1.0.2 – Section: **12.2.6>**
 - objectType
 - objectID
 - objectName
 - parentURI
 - parentID
 - capabilities
 - childrenrange
 - children
4. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: **12.2.7>**
5. Expected Result: Capabilities for the object should be listed.

<Test Case : End>

6.2.1.3 CDMI_Capability_Read_3R_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 12.2
Test Specification Id	3R_3
Test Case Name	capabilityRead_3R_3
Test Case Description	To read a capability object using objectID.
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2–Section: 12.1.1>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference: Login Test Spec Id : 1C_L_1>**
2. Get "cdmi_capabilities" and note its objectID. **<Reference: Test Case Id: 3R_1>**
3. Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 12.2.1>**
Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 12.2.3>**

Request Sample:

```
GET <root URI>/cdmi_objectid/<objectID>/ HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdmi-capability
X-CDMI-Specification-Version: 1.0.2
```

4. Verify that all mandatory fields exist:
<Reference: CDMI Specification v1.0.2 – Section: 12.2.6>
 - objectType
 - objectID
 - objectName
 - parentURI
 - parentID
 - capabilities
 - childrenrange
 - children
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 12.2.7>
6. Expected Result: Capabilities for the object should be listed.

<Test Case : End>

6.2.1.4 CDMI_Capability_Read_3R_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 12.2
Test Specification Id	3R_4
Test Case Name	capabilityRead_3R_4
Test Case Description	To update an existing capability object.
Pre-Test Dependencies	None

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 12.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 12.2.3> Request Sample: <pre>PUT <root URI>/cdmi_capabilities/<child capability>/ HTTP/1.1 Host: cloud.example.com Content-Type: application/cdmi-capability X-CDMI-Specification-Version: 1.0.2 { }</pre> 3. Check for HTTP status code: 405 Not Allowed returned. <Reference: CDMI Specification v1.0.2 – Section: 12.2.7> 4. Expected Result: Capabilities for the object should be listed. <p><Test Case : End></p>

6.2.1.5 CDMI_Capability_Read_3R_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 12.2
Test Specification Id	3R_5
Test Case Name	capabilityRead_3R_5
Test Case Description	To delete a capability object.
Pre-Test Dependencies	None

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 12.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 12.2.3> Request Sample: <pre>DELETE <root URI>/cdmi_capabilities/<child capability>/ HTTP/1.1 Host: cloud.example.com Content-Type: application/cdmi-capability X-CDMI-Specification-Version: 1.0.2</pre> 3. Check for HTTP status code: 405 Not Allowed returned. <Reference: CDMI Specification v1.0.2 – Section: 12.2.7> 4. Expected Result: Capabilities for the object should be listed. <p><Test Case : End></p>

6.3 Container

6.3.1 CDMI_Container_Create

6.3.1.1 CDMI_Container_Create_1C_1

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_1
Test Case Name	containerCreate_1C_1
Test Case Description	To create a container successfully
Pre-Test Dependencies	Capability on Parent Container Capability name: cdmi_create_container < Reference :CDMI Specification v1.0.2 – Section: 9.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Request URL <**Reference**: CDMI Specification v1.0.2 – Section: **9.2.1**>
Request Headers: <**Reference**: CDMI Specification v1.0.2 – Section: **9.2.4**>
Request Body: <**Reference**: CDMI Specification v1.0.2 – Section: **9.2.5**>

Request Sample with Body:

PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-container
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2

{
}

3. Check for HTTP status code: **201 Created** returned.
<**Reference** :CDMI Specification v1.0.2 – Section: **9.2.8**>
4. Verify that all mandatory fields exist:
<**Reference** :CDMI Specification v1.0.2 – Section: **9.2.7**>
 - objectType
 - objectID
 - objectName
 - parentURI
 - domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true)
 - parentID
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
5. Container cleanup process : <**Reference**: 6.5.1 Container Cleanup>

<Test Case : End>

6.3.1.2 CDMI_Container_Create_1C_3

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_3
Test Case Name	containerCreate_1C_3
Test Case Description	To create a container within an existing container
Pre-Test Dependencies	Capability in parent container: cdmi_create_container <Reference: CDMI Specification v1.0.2 – Section: 9.2.3 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials. <Reference Test Case Id : **1C_L_1**>
2. Request URL <Reference: CDMI Specification v1.0.2 – Section: **9.2.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **9.2.4**>
Request Body: <Reference: CDMI Specification v1.0.2 – Section: **9.2.5**>

Request Sample with Body:


```
PUT <root URI>/TestContainer1/TestContainer2/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-container
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
{
}
```
3. Check for HTTP status code: **201 Created** returned.
<Reference: CDMI Specification v1.0.2 – Section: **9.2.8**>
4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: **9.2.7**>
List of mandatory fields:
 - objectType
 - objectID
 - objectName
 - parentURI
 - domainURI (**Conditional:** if system-wide capability "cdmi_domains" is present and true)
 - parentID
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
5. Expected Result: Container should be created successfully.

<Test Case : End>

6.3.1.3 CDMI_Container_Create_1C_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_4
Test Case Name	containerCreate_1C_4
Test Case Description	To not permit creation of a container with an invalid container name
Pre-Test Dependencies	Capability in parent container: cdmi_create_container < Reference: CDMI Specification v1.0.2 – Section: 9.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <**Reference** Test Case Id : 1C_L_1>
2. Request URL <**Reference:** CDMI Specification v1.0.2 – Section: **9.2.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.2.4**>
Request Body: <**Reference:** CDMI Specification v1.0.2 – Section: **9.2.5**>

Request Sample with Body:

```
PUT <root URI>/cdmi_TestContainer1/ HTTP/1.1
Host: cloud.example.com
```

```
Accept: application/cdmi-container
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
```

```
{
}
```

3. Check for HTTP status code: **400 Bad Request** returned.
4. <**Reference:** CDMI Specification v1.0.2 – Section: **9.2.8**>
5. Expected Result: Container should not be created

<Test Case : End>

6.3.1.4 CDMI_Container_Create_1C_5

6.3.1.4.1 CDMI_Container_Create_1C_5.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_5.1
Test Case Name	containerCreate_1C_5.1
Test Case Description	To not permit creation of a container with an invalid header field
Pre-Test Dependencies	Capability in parent container: Capability name: cdmi_create_container < Reference: CDMI Specification v1.0.2 – Section: 9.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id : 1C_L_1>**
2. Request URL **<Reference: CDMI Specification v1.0.2 – Section: 9.2.1>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.4>**
 Request Body: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.5>**

Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com
```

```
Accept: application/container
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```

```
{
}
```

3. Check for HTTP status code: **406 Not Acceptable** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.2.8>
4. Container should not be created.

<Test Case : End>

6.3.1.4.2 CDMI_Container_Create_1C_5.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_5.2
Test Case Name	containerCreate_1C_5.2
Test Case Description	To not permit creation of a container with an invalid header field
Pre-Test Dependencies	Capability in parent container: Capability name: cdm_create_container <Reference: CDMI Specification v1.0.2 – Section: 9.2.3>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id : 1C_L_1>**
2. Request URL **<Reference: CDMI Specification v1.0.2 – Section: 9.2.1>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.4>**
 Request Body: **<Reference: CDMI Specification v1.0.2 – Section: 9.2.5>**

Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com
```

```
Accept: application/cdm-container
```

```
Content-Type: application/cdmi_container
X-CDMI-Specification-Version: 1.0.2

{
}
```

3. Check for HTTP status code: **400 Bad Request** returned.
<Reference: CDMI Specification v1.0.2 – Section: **9.2.8**>
4. Container should not be created.

<Test Case : End>

6.3.1.5 CDMI_Container_Create_1C_9

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_9
Test Case Name	containerCreate_1C_9
Test Case Description	To fail the creation of a container by using the reference field with other fields
Pre-Test Dependencies	Capability in parent container: Capability name: <ul style="list-style-type: none"> • cdmi_create_container • cdmi_create_reference • cdmi_copy_container <Reference: CDMI Specification v1.0.2 – Section: 9.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Login Id: **1C_L_1**>
2. To create container "TestContainer1," see Container_Create_1C_1 test.
3. Create a container named "TestContainer2" <Test Id: **1C_1**>
4. Request URL <Reference: CDMI Specification v1.0.2 – Section: **9.2.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **9.2.4**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **9.2.5**>

 Request Sample with Body:


```
PUT <root URI>/TestContainer2/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-container
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
{
  "reference" : "/TestContainer1/",
  "copy" : "/TestContainer1/"
}
```
5. Check for HTTP status code: **400 Bad Request** returned.

<p><Reference: CDMI Specification v1.0.2 – Section: 9.2.8></p> <p>6. Expected Result: "TestContainer2" should not be created.</p> <p><Test Case : End></p>

6.3.1.6 CDMI _Container_ Create_1C_10

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_10
Test Case Name	containerCreate_1C_10
Test Case Description	To not permit creation of a container without slashes in the URI <Reference: CDMI Specification v1.0.2 – Section: 9.1 >
Pre-Test Dependencies	Capability in parent container: Capability name: cdmi_create_container <Reference: CDMI Specification v1.0.2 – Section: 9.2.3 > Check: TestContainer1 does not exist

D E S C R I P T I O N

<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Login Id: 1C_L_1> Request URL <Reference: CDMI Specification v1.0.2 – Section: 9.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.2.4> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 9.2.5> Request Sample with Body: PUT <root URI>/TestContainer1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-container Content-Type: application/cdmi-container X-CDMI-Specification-Version: 1.0.2 { } Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 9.2.8> Expected Result: "TestContainer1" should not be created. <p><Test Case : End></p>

6.3.1.7 CDMI _Container_ Create_1C_11

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_11
Test Case Name	containerCreate_1C_11
Test Case Description	To create a container by copying an existing container

<p>Pre-Test Dependencies</p>	<p>Capability in parent container: Capability name:</p> <ul style="list-style-type: none"> • cdmi_create_container • cdmi_copy_container <p><Reference: CDMI Specification v1.0.2 – Section: 9.2.3> Check: TestContainer1 does not exist</p>
<p>D E S C R I P T I O N</p>	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Login Id : 1C_L_1> 2. Create a container named "TestContainer1" <Test Id: 1C_1> 3. Request URL <Reference: CDMI Specification v1.0.2 – Section: 9.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.2.4> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 9.2.5> <p>Request Sample with Body:</p> <pre>PUT <root URI>/TestContainer2/ HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-container Content-Type: application/cdmi-container X-CDMI-Specification-Version: 1.0.2 { "copy" : "/TestContainer1/" }</pre> <ol style="list-style-type: none"> 4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 9.2.7> List of mandatory fields for "TestContainer2": <ul style="list-style-type: none"> – objectType – objectID – objectName – parentURI – domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true) – parentID – capabilitiesURI – metadata – completionStatus – children – childrenrange 5. Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 9.2.8> 6. Expected Result: "TestContainer2" should be created successfully which is a copy of "TestContainer1". 7. Check that all fields and their values are reproduced correctly. <p><Test Case : End></p>	

6.3.1.8 CDMI_Container_Create_1C_12

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.2
Test Specification Id	1C_12
Test Case Name	containerCreate_1C_12
Test Case Description	To move a container to a new URI
Pre-Test Dependencies	Capability in parent container: Capability name: <ul style="list-style-type: none"> • cdmi_create_container • cdmi_move_container <Reference: CDMI Specification v1.0.2 – Section: 9.2.3 > Check: TestContainer1/TestContainer2/ does not exist

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Login Id : **1C_L_1**>
2. Create a container named "TestContainer2" inside "TestContainer1" <Test Id: **1C_3**>
3. Request URL <Reference: CDMI Specification v1.0.2 – Section: **9.2.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **9.2.4**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **9.2.5**>

 Request Sample with Body:


```

PUT <root URI>/TestContainer1/TestContainer3/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-container
Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
{
  "move" : "/TestContainer1/TestContainer2/"
}
```
4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: **9.2.7**>
 List of mandatory fields for "TestContainer3":
 - objectType
 - objectID
 - objectName
 - parentURI
 - domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true)
 - parentID
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
5. Check for HTTP status code: **201 Created** returned.
 <Reference: CDMI Specification v1.0.2 – Section: **9.2.8**>

6. Expected Result: "TestContainer3" should be created successfully and "TestContainer2" deleted.
7. Check that all fields and their values are reproduced correctly.

<Test Case : End>

6.3.2 CDMI_Container_Read

6.3.2.1 CDMI_Container_Read_1R_1

CDMI Specification Reference	CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_1
Test Case Name	containerRead_1R_1
Test Case Description	To read a container successfully
Pre-Test Dependencies	Container should be created < Reference: 7.3.1 CDMI_Container_Create_1C_1>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Request URL<**Reference** :CDMI Specification v1.0.2 – Section: **9.4.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.4.3**>

Request Sample:

GET <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
3. Check for HTTP status code: **200 OK** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **9.4.7**>

'TestContainer1' should be read at the desired location.
4. Verify that all mandatory fields exist: <**Reference** :CDMI Specification v1.0.2 – Section: **9.2.7**>
List of mandatory fields:
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - parentID
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange

<Test Case : End>

6.3.2.2 CDMI_Container_Read_1R_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2–Section: 9.4
Test Specification Id	1R_2
Test Case Name	containerRead_1R_2
Test Case Description	To read a container by objectID
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID < Reference: CDMI Specification v1.0.2–Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Login Id : **1C_L_1**>
2. Create a container named "TestContainer1" and read its 'objectID' <Test Id: **1C_1**>
3. Request URL<**Reference** :CDMI Specification v1.0.2 – Section: **9.4.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.4.3**>

Request Sample:

GET <root URI>/cdmi_objectid/<objectID of container>/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
4. Check for HTTP status code: **200 OK** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **9.4.7**>
5. Verify all mandatory fields exist: <**Reference:** CDMI Specification v1.0.2 – Section: **9.4.6**>
List of mandatory fields for "TestContainer1":
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
6. Expected Result: "TestContainer1" should be read successfully.

<Test Case : End>

6.3.2.3 CDMI_Container_Read_1R_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_4
Test Case Name	containerRead_1R_4
Test Case Description	To read a container with an invalid header field
Pre-Test Dependencies	NA

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Login Id : 1C_L_1> 2. Create a container named "TestContainer1" <Test Id: 1C_1> 3. Request URL<Reference :CDMI Specification v1.0.2 – Section: 9.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.4.3> Request Sample: GET <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com Accept: application/cdmi_container X-CDMI-Specification-Version: 1.0.2 4. Check for HTTP status code: 406 Not Acceptable returned. <Reference: CDMI Specification v1.0.2 – Section: 9.4.7> 5. Expected Result: "TestContainer1" should not be listed. <p><Test Case : End></p>

6.3.2.4 CDMI _Container_Read_1R_6

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_6
Test Case Name	containerRead_1R_6
Test Case Description	To read a non-existent container
Pre-Test Dependencies	"Test Container1" should not exist

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Login Id : 1C_L_1> 2. Request URL<Reference :CDMI Specification v1.0.2 – Section: 9.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.4.3> Request Sample: GET <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-container X-CDMI-Specification-Version: 1.0.2 3. Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 9.4.7> 4. Expected Result: "TestContainer1" should not be found. <p><Test Case : End></p>

6.3.2.5 CDMI_Container_Read_1R_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_7
Test Case Name	containerRead_1R_7
Test Case Description	To read a container with limited fields
Pre-Test Dependencies	TestContainer1 shall exist

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id : 1C_L_1>**
2. Create a container named "TestContainer1" **<Test Id: 1C_1>**
3. Request URL**<Reference :CDMI Specification v1.0.2 – Section: 9.4.1>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.3>**
 Request Sample:

```
GET<root URI>/TestContainer1/?objectType;objectId HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```
4. Verify that all mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.6>**
 List of mandatory fields for "TestContainer1":
 - objectType
 - objectID
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.4.7>
6. Expected Result: "TestContainer1" should be read successfully.

<Test Case : End>

CDMI _

6.3.2.6 Container_Read_1R_9

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_9
Test Case Name	containerRead_1R_9
Test Case Description	To read a container without a "/" (slash) in URI
Pre-Test Dependencies	
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference Test Case Id: 1C_L_1> 2. Create a container named "TestContainer1" <Test Id: 1C_1> 3. Request URL<Reference :CDMI Specification v1.0.2 – Section: 9.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.4.3> Request Sample: GET <root URI>/TestContainer1 HTTP/1.1 Host: cloud.example.com Accept: application/cdm-container X-CDMI-Specification-Version: 1.0.2 4. Check for HTTP status code: 301 Moved permanently returned. <Reference: CDMI Specification v1.0.2 – Section: 9.1> 5. Expected Result: "TestContainer1" should not be listed. <p><Test Case : End></p>	

6.3.2.7 CDMI_Container_Read_1R_10

6.3.2.7.1 CDMI_Container_Read_1R_10.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_10.1
Test Case Name	containerRead_1R_10.1
Test Case Description	To read a container using percent escaping of reserved characters <Reference: CDMI Specification v1.0.2 – Section: 5.13.4>
Pre-Test Dependencies	
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference Test Case Id: 1C_L_1> 2. Create a container named "@TestContainer1" <Test Id: 1C_1> 3. Request URL:<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4> 	

Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.4.3**>

Request Sample:

```
GET <root URI>/%40TestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
```

4. Verify that all mandatory fields exist: <**Reference:** CDMI Specification v1.0.2 – Section: **9.4.6**>
List of mandatory fields for "@TestContainer1":
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
5. Check for HTTP status code: **200 OK** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **9.4.7**>
6. Expected Result: "@TestContainer1" should be read successfully.

<Test Case : End>

6.3.2.7.2 CDMI_Container_Read_1R_10.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_10.2
Test Case Name	containerRead_1R_10.2
Test Case Description	To read a container using percent escaping of reserved characters < Reference: CDMI Specification v1.0.2 – Section: 5.13.4 >
Pre-Test Dependencies	

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id: 1C_L_1>**
2. Create a container named ":TestContainer1" **<Test Id: 1C_1>**
3. Request URL:**<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.3>**

 Request Sample:


```
GET <root URI>/%3ATestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```
4. Verify that all mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.6>**
 List of mandatory fields for ":TestContainer1":
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.4.7>
6. Expected Result: ":TestContainer1" should be read successfully.

<Test Case : End>

6.3.2.7.3 CDMI_Container_Read_1R_10.4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_10.4
Test Case Name	containerRead_1R_10.4
Test Case Description	To read a container using percent escaping of reserved characters <Reference: CDMI Specification v1.0.2 – Section: 5.13.4>
Pre-Test Dependencies	

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id: 1C_L_1>**
2. Create a container named "?TestContainer1" **<Test Id: 1C_1>**
3. Request URL:**<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.3>**

 Request Sample:


```
GET <root URI>/%3FTestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```
4. Verify that all mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.6>**
 List of mandatory fields for "?TestContainer1":
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.4.7>
6. Expected Result: "?TestContainer1" should be read successfully.

<Test Case : End>

6.3.2.7.4 CDMI_Container_Read_1R_10.5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_10.5
Test Case Name	containerRead_1R_10.5
Test Case Description	To read a container using percent escaping of reserved characters <Reference: CDMI Specification v1.0.2 – Section: 5.13.4>
Pre-Test Dependencies	

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id: 1C_L_1>**
2. Create a container named "#TestContainer1" **<Test Id: 1C_1>**
3. Request URL:**<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.3>**

 Request Sample:


```
GET <root URI>/%23TestContainer1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```
4. Verify that all mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.6>**
 List of mandatory fields for "#TestContainer1":
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - capabilitiesURI
 - metadata
 - completionStatus
 - children
 - childrenrange
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.4.7>
6. Expected Result: "#TestContainer1" should be read successfully.

<Test Case : End>

6.3.2.8 CDMI_Container_Read_1R_11

6.3.2.8.1 CDMI_Container_Read_1R_11.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_11.1
Test Case Name	containerRead_1R_11.1
Test Case Description	To read a container using percent escaping of reserved characters in metadata. <Reference: CDMI Specification v1.0.2 – Section: 5.13.4>
Pre-Test Dependencies	

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id: 1C_L_1>**
2. Create a container named "@TestContainer1" with metadata "@user" **<Test Id: 1C_1>**
3. Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.3>**

 Request Sample:


```
GET <root URI>/%40TestContainer1/?metadata:%40user HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```
4. Verify that all mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.6>**
 List of mandatory fields for "@TestContainer1": metadata
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.4.7>
6. Expected Result: "@TestContainer1" should be read successfully.

<Test Case : End>

6.3.2.8.2 CDMI_Container_Read_1R_11.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_11.2
Test Case Name	containerRead_1R_11.2
Test Case Description	To read a container using percent escaping of reserved characters in metadata. <Reference: CDMI Specification v1.0.2 – Section: 5.13.4>
Pre-Test Dependencies	

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference Test Case Id: 1C_L_1>**
2. Create a container named "TestContainer1" with metadata ":user" **<Test Id: 1C_1>**
3. Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.4.3>**

 Request Sample:


```
GET <root URI>/TestContainer1/?metadata:%3Auser HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-container
```

<p>X-CDMI-Specification-Version: 1.0.2</p> <ol style="list-style-type: none"> Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 9.4.6> List of mandatory fields for "TestContainer1": metadata Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 9.4.7> Expected Result: "TestContainer1" should be read successfully. <p><Test Case : End></p>
--

6.3.2.8.3 CDMI_Container_Read_1R_11.4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_11.4
Test Case Name	containerRead_1R_11.4
Test Case Description	To read a container using percent escaping of reserved characters in metadata. <Reference: CDMI Specification v1.0.2 – Section: 5.13.4>
Pre-Test Dependencies	

D E S C R I P T I O N

<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference Test Case Id: 1C_L_1> Create a container named "TestContainer1" with metadata "@user" <Test Id: 1C_1> Request URL:<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.4.3> Request Sample: GET <root URI>/TestContainer1/?metadata:%40user HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-container X-CDMI-Specification-Version: 1.0.2 Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 9.4.6> List of mandatory fields for "TestContainer1": metadata Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 9.4.7> Expected Result: "TestContainer1" should be read successfully. <p><Test Case : End></p>

6.3.2.8.4 CDMI_Container_Read_1R_11.5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_11.5
Test Case Name	containerRead_1R_11.5
Test Case Description	To read a container using percent escaping of reserved characters in metadata <Reference: CDMI Specification v1.0.2 – Section: 5.13.4 >
Pre-Test Dependencies	
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference Test Case Id: 1C_L_1> 2. Create a container named "TestContainer1" with metadata "?user"<Test Id: 1C_1> 3. Request URL:<Reference: CDMI Specification v1.0.2 – Section: 9.4.1 & 5.13.4> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.4.3> Request Sample: GET <root URI>/TestContainer1/?metadata:%3Fuser HTTP/1.1 Host: cloud.example.com Accept: application/cdm-container X-CDMI-Specification-Version: 1.0.2 4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 9.4.6> List of mandatory fields for "TestContainer1": metadata 5. Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 9.4.7> 6. Expected Result: "TestContainer1" should be read successfully. <p><Test Case : End></p>	

6.3.2.9 CDMI_Container_Read_1R_12

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.4
Test Specification Id	1R_12
Test Case Name	containerRead_1R_12
Test Case Description	To create a container via non-CDMI and read it via CDMI
Pre-Test Dependencies	NA

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Create a container named "TestContainer1". <Reference: Test Case Id: NC_1C_1> Request URL<Reference: CDMI Specification v1.0.2 – Section: 9.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.4.3> Request Sample: <pre>GET <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com Accept: application/cdm-container X-CDMI-Specification-Version: 1.0.2</pre> 3. Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 9.4.7> 4. Expected Result: Container should be created and read successfully. 5. Container cleanup process: <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>

6.3.3 CDMI_Container_Update

6.3.3.1 CDMI_Container_Update_1U_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2–Section:9.5
Test Specification Id	1U_1
Test Case Name	containerUpdate_1U_1
Test Case Description	To update a container by objectID
Pre-Test Dependencies	Capability in parent container: cdm_modify_metadata <Reference: CDMI Specification v1.0.2–Section:9.5.> System-wide capability: cdm_object_access_by_ID <Reference: CDMI Specification v1.0.2–Section: 12.1.1>

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a container named "TestContainer1" and note its objectID<Test Id: 1C_6> 3. Request URL:<Reference: CDMI Specification v1.0.2 – Section: 9.5.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.5.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 9.5.5> Request Sample with Body: <pre>PUT <root URI>/cdmi_objectid/<objectedID of "TestContainer1">/ HTTP/1.1</pre>

```
Host: cloud.example.com

Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2

{
  "metadata": {
    "Color" : "Green"
  }
}
```

4. Check for HTTP status code: **204 No Content** returned.
<Reference: CDMI Specification v1.0.2 – Section: **9.5.8**>
5. Expected Result: "TestContainer1" should be updated.

<Test Case : End>

6.3.3.2 CDMI_Container_Update_1U_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_2
Test Case Name	containerUpdate_1U_2
Test Case Description	To update a container with invalid header field
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 9.5.3 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials.<Reference Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Request URL:<Reference: CDMI Specification v1.0.2 – Section: **9.5.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **9.5.4**>
Request Body:<Reference: CDMI Specification v1.0.2 – Section: **9.5.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdmi_container
X-CDMI-Specification-Version: 1.0.2
{
  "metadata" : {
    "Color" : "Green"
  }
}
```

4. Check for HTTP status code: **400 Bad Request** returned.
<Reference: CDMI Specification v1.0.2 – Section: **9.5.8**>

5. Expected Result: "TestContainer1" should not be updated.

<Test Case : End>

6.3.3.3 CDMI_Container_Update_1U_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_5
Test Case Name	containerUpdate_1U_5
Test Case Description	To update a container with valid fields in request message body
Pre-Test Dependencies	System-wide capability: cdmi_domains < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Check cross-domain privilege for user: cdmi_member_privileges < Reference: CDMI Specification v1.0.2 – Section: 10.1.3 > Capability in parent domain: cdmi_create_domain < Reference: CDMI Specification v1.0.2 – Section: 10.2.2 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Create a domain named "MyNewDomain1" <Test ID: **7.5.1**>
4. Request URL:<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.5.4**>
Request Body:<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2

{
  "domainURI" = "/cdmi_domains/MyNewDomain1/"
}
```

5. Check for HTTP status code: **204 No Content** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.8**>
6. Expected Result: Valid fields of "TestContainer1" should be updated.

<Test Case : End>

6.3.3.4 CDMI_Container_Update_1U_6

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_6
Test Case Name	containerUpdate_1U_6
Test Case Description	To update metadata of container by passing in request message body with URL instead of ObjectId
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata < Reference: CDMI Specification v1.0.2 – Section: 9.5.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Request URL:<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.5.4**>
Request Body:<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
{
  "metadata" : {
    "Color" : "Green"
  }
}
```

4. Check for HTTP status code: **204 No Content** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.8**>
5. Expected Result: Valid metadata of "TestContainer1" should be updated.

<Test Case : End>

6.3.3.5 CDMI_Container_Update_1U_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_7
Test Case Name	containerUpdate_1U_7
Test Case Description	To update a container with valid fields in URI

Pre-Test Dependencies	System-wide capability: cdmi_domains <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Check cross-domain privilege for user: cdmi_member_privileges <Reference: CDMI Specification v1.0.2 – Section: 10.1.3 > Capability in parent domain: cdmi_create_domain <Reference: CDMI Specification v1.0.2 – Section: 10.2.2 > Note: This test case will not run if 'cdmi_domains' & 'cdmi_create_domain' capabilities are not supported and user does not have cross-domain privilege.
------------------------------	---

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<Reference Test Case Id: **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Create a domain named "MyNewDomain1" <Test ID: 7.5.1>
4. Request URL:<Reference: CDMI Specification v1.0.2 – Section: **9.5.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **9.5.4**>
 Request Body:<Reference: CDMI Specification v1.0.2 – Section: **9.5.5**>

 Request Sample with Body:


```

PUT <root URI>/TestContainer1/?domainURI HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdmi-container
X-CDMI-Specification-Version: 1.0.2
{
  "domainURI" = "/cdmi_domains/MyNewDomain1/"
}
            
```
5. Check for HTTP status code: **204 No Content** returned.
 <Reference: CDMI Specification v1.0.2 – Section: **9.5.8**>
6. Expected Result: Valid fields of "TestContainer1" should be updated.

<Test Case : End>

6.3.3.6 CDMI_Container_Update_1U_8

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_8
Test Case Name	containerUpdate_1U_8
Test Case Description	To update metadata of container by passing it in URI
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 9.5.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Request URL:<**Reference**: CDMI Specification v1.0.2 – Section: **9.5.1**>
Request Headers: <**Reference**: CDMI Specification v1.0.2 – Section: **9.5.4**>
Request Body:<**Reference**: CDMI Specification v1.0.2 – Section: **9.5.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/?metadata:Color HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```

```
{
  "metadata" : {
    "Color" : "Green"
  }
}
```

4. Check for HTTP status code: **204 No Content** returned.
<**Reference**: CDMI Specification v1.0.2 – Section: **9.5.8**>
5. Expected Result: Metadata of "TestContainer1" should be updated.

<Test Case : End>

6.3.3.7 CDMI_Container_Update_1U_9

6.3.3.7.1 CDMI_Container_Update_1U_9.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_9.1
Test Case Name	containerUpdate_1U_9.1
Test Case Description	To not permit update of a system defined/reserved container
Pre-Test Dependencies	

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference: Test Case Id : 1C_L_1>**
2. Request URL **<Reference: CDMI Specification v1.0.2 – Section: 9.5.1>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 9.5.4>**
 Request Body: **<Reference: CDMI Specification v1.0.2 – Section: 9.5.5>**

Request Sample with Body:

```
PUT <root URI>/cdmi_capabilities/ HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```

```
{
}
```

3. Check for HTTP status code: **405 Method Not Allowed** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.5.8>
4. Expected Result: Container should not be updated.

<Test Case : End>

6.3.3.8 CDMI _Container_Update_1U_12

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_12
Test Case Name	containerUpdate_1U_12
Test Case Description	To create a snapshot of existing container
Pre-Test Dependencies	Capability in container: cdmi_create_container <Reference :CDMI Specification v1.0.2 – Section: 9.2.3> System-wide capability: cdmi_snapshots <Reference: CDMI Specification v1.0.2 – Section: 9.5.1>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Request URL:<**Reference**: CDMI Specification v1.0.2 – Section: **9.5.1**>
Request Headers: <**Reference**: CDMI Specification v1.0.2 – Section: **9.5.4**>
Request Body:<**Reference**: CDMI Specification v1.0.2 – Section: **9.5.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2
```

```
{
    "snapshot" : "test_snapshot"
}
```

4. Check for HTTP status code: **204 No Content** returned.
<**Reference**: CDMI Specification v1.0.2 – Section: **9.5.8**>
5. Expected Result: Snapshot of "TestContainer1" should be created at URI:
"/TestContainer1/cdm_snapshots/test_snapshot"

<Test Case : End>

6.3.3.9 CDMI_Container_Update_1U_13

CDMI Specification Reference	Reference : CDMI Specification v1.0.2 – Section: 9.5
Test Specification Id	1U_13
Test Case Name	containerUpdate_1U_13
Test Case Description	To update a container without slash in URI
Pre-Test Dependencies	Capability in parent container: cdm_modify_metadata < Reference : CDMI Specification v1.0.2 – Section: 9.5.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<Test Id: **1C_6**>
3. Request URL:<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.5.4**>
Request Body:<**Reference:** CDMI Specification v1.0.2 – Section: **9.5.5**>

Request Sample with Body:

PUT <root URI>/TestContainer1 HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdm-container
X-CDMI-Specification-Version: 1.0.2


```
{
  "metadata" : {
    "Color" : "Green"
  }
}
```
4. Check for HTTP status code: **301 Moved Permanently** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **9.1**>
5. Expected Result: "TestContainer1" should not be updated.

<Test Case : End>

6.3.4 CDMI_Container_Delete

6.3.4.1 CDMI_Container_Delete_1D_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.6
Test Specification Id	1D_1
Test Case Name	containerDelete_1D_1
Test Case Description	If container exists then delete the same
Pre-Test Dependencies	Capability on Parent Container Capability name: cdm_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.6.2 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference:** Test Case Id : **1C_L_1**>
2. Create a container named "TestContainer1"<**Reference:** Test Id: **1C_6**>
3. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **9.6.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **9.6.3**>

<p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned <Reference: CDMI Specification v1.0.2 – Section: 9.6.7> 5. 'TestContainer1' should be deleted at the desired location. <p><Test Case : End></p>

6.3.4.2 CDMI_Container_Delete_1D_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.6
Test Specification Id	1D_2
Test Case Name	containerDelete_1D_2
Test Case Description	If container exists then delete the same via objectid
Pre-Test Dependencies	Parent Container Capability: cdm_delete_container <Reference :CDMI Specification v1.0.2 – Section: 9.6.2 > System-wide capability: cdm_object_access_by_ID <Reference: CDMI Specification v1.0.2–Section: 12.1.1 >

D E S C R I P T I O N

<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a container named "TestContainer1" and note its objectid 3. <Reference: Test Id: 1C_6> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.6.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.6.3> <p>Request Sample:</p> <pre>DELETE <root URI>/cdmi_objectid/<objectID of TestContainer1>/ HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned <Reference: CDMI Specification v1.0.2 – Section: 9.6.7> 5. 'TestContainer1' should be deleted at the desired location. <p><Test Case : End></p>
--

6.3.4.3 CDMI_Container_Delete_1D_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.6
Test Specification Id	1D_3
Test Case Name	containerDelete_1D_3
Test Case Description	To delete a container that does not exist
Pre-Test Dependencies	Parent Container Capability: cdmi_delete_container < Reference :CDMI Specification v1.0.2 – Section: 9.6.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.6.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.6.3> Request Sample: DELETE <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2 3. Check for HTTP status code: 404 Not Found returned <Reference: CDMI Specification v1.0.2 – Section: 9.6.7> 4. 'TestContainer1' does not exist. <p><Test Case : End></p>	

6.3.4.4 CDMI_Container_Delete_1D_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.6
Test Specification Id	1D_4
Test Case Name	containerDelete_1D_4
Test Case Description	To create a container via non-CDMI and delete it via CDMI
Pre-Test Dependencies	Container Capability: Capability name: cdmi_delete_container < Reference :CDMI Specification v1.0.2 – Section: 9.6.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a container named "TestContainer1". <Reference: Test Case Id: NC_1C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.6.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.6.3> 	

<p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned <Reference: CDMI Specification v1.0.2 – Section: 9.6.7> 5. 'TestContainer1' should be deleted successfully. <p><Test Case : End></p>
--

6.4 Data Object

6.4.1 CDMI_DataObject_Create

6.4.1.1 CDMI_DataObject_Create_2C_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_1
Test Case Name	dataObjectCreate_2C_1
Test Case Description	To create a data object with invalid header
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

DESCRIPTION

<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> <p>Request Sample with Body:</p> <pre>PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host: cloud.example.com Accept: application/cdm-object Content-Type: application/cdm-object X-CDMI-Specification-Version: 1.0.2 { }</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 406 Not Acceptable returned. <Reference: CDMI Specification v1.0.2 – Section: 8.2.8>
--

5. Expected Result: Data object "data_object1" should not be created.
6. Container cleanup process : <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.2 CDMI_DataObject_Create_2C_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_3
Test Case Name	dataObjectCreate_2C_3
Test Case Description	To create a data object successfully
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> Request Sample with Body: <pre>PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 { }</pre> 4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.2.7> List of mandatory fields for "data_object1": <ul style="list-style-type: none"> – objectType – objectID – objectName – parentURI – domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true) – parentID – capabilitiesURI – metadata – completionStatus – mimetype 	

5. Check for HTTP status code: **201 Created** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
6. Expected Result: Data object "data_object1" should be created.
7. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.3 CDMI_DataObject_Create_2C_6

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_6
Test Case Name	dataObjectCreate_2C_6
Test Case Description	To create a data object using reference field with other fields that are incorrect
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Reference: Test Spec Id: **1C_L_1**>
2. Create a data object "data_object1" into container "TestContainer1" <Reference: Test Spec Id: **2C_3**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<Reference: CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/dataobject2 HTTP/1.1
Host: cloud.example.com
```

```
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2
```

```
{
  "reference" : "/TestContainer1/data_object1",
  "copy" : "/TestContainer1/data_object1"
}
```

4. Check for HTTP status code: **400 Bad Request** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
5. Expected Result: Data object "data_object1" should not be created.
6. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.4 CDMI_DataObject_Create_2C_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_7
Test Case Name	dataObjectCreate_2C_7
Test Case Description	To create a data object by moving an existing data object
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 > Capability in parent container: cdmi_move_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <**Reference:** Test Spec Id : **1C_L_1**>
2. Create a data object "data_object1" in container named "TestContainer1" <**Reference:** Test Spec Id: **2C_3**>
3. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<**Reference:** CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:

PUT <root URI>/TestContainer1/data_object2 HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2

{
 "move" : "/TestContainer1/data_object1"
}
4. Verify that all mandatory fields exist: <**Reference:** CDMI Specification v1.0.2 – Section: **8.2.7**>
List of mandatory fields for "data_object2":
 - objectType
 - objectID
 - objectName
 - parentURI
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - parentID
 - capabilitiesURI
 - metadata
 - completionStatus
 - mimetype
5. Check for HTTP status code: **201 Created** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **8.2.8**>

6. Expected Result: Data object "data_object2" should be created with object ID of "data_object1".
7. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.5 CDMI_DataObject_Create_2C_8

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_8
Test Case Name	dataObjectCreate_2C_8
Test Case Description	To create data object with incorrect combination of 'Value Transfer Encoding' and 'Value'
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials. <Reference: Test Spec Id: **1C_L_1**>
2. Create a container named "TestContainer1" <Reference: Test Spec Id: **1C_1**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<Reference: CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/data_object1 HTTP/1.1
Host: cloud.example.com
```

```
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2
```

```
{
  "valuetransferencoding" : "base64",
  "value": " A non-Base64 String!"
}
```

4. Check for HTTP status code **400 Bad Request** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
5. Expected Result: Data object "data_object1" should not be created.
6. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.6 CDMI_DataObject_Create_2C_9

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_9
Test Case Name	dataObjectCreate_2C_9
Test Case Description	To create a data object by defining user-defined metadata, when user defined metadata items are less than or equal to the value of system-wide capability (cdmi_metadata_maxitems)
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_metadata_maxitems >=1 < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1> 3. Retrieve value of capability "cdmi_metadata_maxitems" in 'no_of_metadata_maxitems' 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> Request Sample with Body: PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 <pre> { "metadata" : { "Test Field" : "Test Value" } } </pre> 5. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.2.7> List of mandatory fields for "data_object1": <ul style="list-style-type: none"> – objectType – objectID – objectName – parentURI – domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true) – parentID – capabilitiesURI – metadata 	

- completionStatus
 - mimetype
 - Check for HTTP status code: **201 Created** returned.
 - <Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
 - Expected Result: Data object "data_object1" should be created.
 - Container cleanup process: <Reference: 6.5.1 Container Cleanup>
- <Test Case : End>

6.4.1.7 CDMI_DataObject_Create_2C_10

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_10
Test Case Name	dataObjectCreate_2C_10
Test Case Description	To create a data object by defining user-defined metadata, when user defined metadata items are greater than the value of system-wide capability(cdmi_metadata_maxitems)
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_metadata_maxitems=N <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Conditional: If "cdmi-metadata_maxitems" capability not exists or If "cdmi-metadata_maxitems" is more than 1025, then this test case shall not run. Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Reference: Test Spec Id: **1C_L_1**>
2. Create a container named "TestContainer1" <Reference: Test Spec Id: **1C_1**>
3. Retrieve value of capability "cdmi_metadata_maxitems" in 'no_of_metadata_maxitems'. Assumed it is 'N'.
4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<Reference: CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/data_object1 HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-object
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.2
{
  "metadata" : {
    "Test Field1" : "Test Value1",
    "Test Field2" : "Test Value2",
```



```

        .
        .
        .
        "Test Field N" : "Test Value N",
        "Test Field N+1" : "Test Value N+1",
    }
}

```

5. Check for HTTP status code: **400 Bad Request** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
6. Expected Result: Data object "data_object1" should not be created.
7. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.8 CDMI_DataObject_Create_2C_11

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_11
Test Case Name	dataObjectCreate_2C_11
Test Case Description	Create a data object with slash
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Reference: Test Spec Id : **1C_L_1**>
2. Create a container named "TestContainer1" <Reference: Test Spec Id: **1C_1**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<Reference: CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:


```

PUT <root URI>/TestContainer1/data_object1/ HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2

{
}

```
4. Check for HTTP status code: **400 Bad Request** or **404 Not Found** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
5. Expected Results: Data object "data_object1" should not be created.
6. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.9 CDMI_DataObject_Create_2C_12

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_12
Test Case Name	dataObjectCreate_2C_12
Test Case Description	Create a data object when valuetransferencoding passed without a value
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials. <**Reference:** Test Spec Id: **1C_L_1**>
2. Create a container named "TestContainer1" <**Reference:** Test Spec Id: **1C_1**>
3. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<**Reference:** CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:

PUT <root URI>/TestContainer1/data_object1 HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2

{
 "valuetransferencoding" : "base64"
}
4. Check for HTTP status code: **201 Created** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **8.2.8**>
5. Expected Results: Data object "data_object1" should be created.
6. Container cleanup process: <**Reference:** 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.10 CDMI_DataObject_Create_2C_14

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_14
Test Case Name	dataObjectCreate_2C_14

Test Case Description	Create a data object when value passed without a value transfer encoding
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> Request Sample with Body: PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 { "value" : "This is the Value of this Data Object" } 4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.2.7> List of mandatory fields: – objectType – objectID – objectName – parentURI – parentID – domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true) – capabilitiesURI – completionStatus – mimetype – metadata 5. Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 8.2.8> 6. Expected Result: Data object "data_object1" should be created. 7. Container cleanup process: <Reference: 6.5.1 Container Cleanup> <p><Test Case : End></p>	

6.4.1.11 CDMI_DataObject_Create_2C_15

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_15
Test Case Name	dataObjectCreate_2C_15
Test Case Description	Create a data object with valid value transfer encoding.
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> Request Sample with Body: PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 { "mimetype" : "text/plain", "valuetransferencoding" : "base64", "value" : "VGhpcyBpcyB0aGUgVmFsdWUgb2YgdGhpcyBEYXRhIE9iamVjdA==" } Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.2.7> List of mandatory fields: <ul style="list-style-type: none"> – objectType – objectID – objectName – parentURI – parentID – domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true) – capabilitiesURI – completionStatus – mimetype – metadata Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 8.2.8> 	

6. Expected Result: Data object "data_object1" should be created.
7. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.12 CDMI_DataObject_Create_2C_16

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_16
Test Case Name	dataObjectCreate_2C_16
Test Case Description	Create a reference of non-existing data object
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 > Capability in parent container: cdmi_create_reference <Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <Reference: Test Spec Id: **1C_L_1**>
2. Create a container named "TestContainer1" <Reference: Test Spec Id: **1C_1**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.2.1**>
Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.2.4**>
Request Body:<Reference: CDMI Specification v1.0.2 – Section: **8.2.5**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/dataobject2 HTTP/1.1
Host: cloud.example.com
```

```
Accept: application/cdmi-object
Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2
```

```
{
  "reference" : "/TestContainer1/dataobject1"
}
```

4. Check for HTTP status code: **404 Not found** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.2.8**>
5. Expected Result: Reference Data object "dataobject2" should not be created
6. Container cleanup process: <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

6.4.1.13 CDMI_DataObject_Create_2C_17

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_17
Test Case Name	dataObjectCreate_2C_17
Test Case Description	Create a reference data object
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 > Capability in parent container: cdmi_create_reference < Reference: CDMI Specification v1.0.2 – Section: 8.2.3 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a data object "data_object1" into a container named "TestContainer1" <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> Request Sample with Body: PUT <root URI>/TestContainer1/dataobject2 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 { "reference" : "/TestContainer1/data_object1" } 4. Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 8.2.8> 5. Expected Result: Data object "dataobject2" should be created and no response body returned. 6. Container cleanup process: <Reference: 6.5.1 Container Cleanup> <p><Test Case : End></p>	

6.4.1.14 CDMI_DataObject_Create_2C_18

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.2
Test Specification Id	2C_18
Test Case Name	dataObjectCreate_2C_18
Test Case Description	Create a reference data object via objectID

<p>Pre-Test Dependencies</p>	<p>System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.2.3> Capability in parent container: cdmi_create_reference <Reference: CDMI Specification v1.0.2 – Section: 8.2.3></p>
<p>D E S C R I P T I O N</p>	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a data object "data_object1" into a container named "TestContainer1" <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.2.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.2.4> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.2.5> Request Sample with Body: <pre> PUT <root URI>/TestContainer1/dataobject2 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 { "reference" : "/TestContainer1/<objectID of data_object1>" } </pre> 4. Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 8.2.8> 5. Expected Result: Data object "dataobject2" should be created and no response body returned. 6. Container cleanup process: <Reference: 6.5.1 Container Cleanup> <p><Test Case : End></p>	

6.4.2 CDMI_DataObject_Read

6.4.2.1 CDMI_DataObject_Read_1

<p>CDMI Specification Reference</p>	<p>Reference: CDMI Specification v1.0.2 – Section: 8.4</p>
<p>Test Specification Id</p>	<p>2R_1</p>
<p>Test Case Name</p>	<p>dataObjectRead_2R_1</p>
<p>Test Case Description</p>	<p>To read a data object by objectID</p>

Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> Create a data object "DataObject1" and note its objectID. <Reference: Test Spec Id: 2C_3> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: GET <root URI>/cdmi_objectid/<objectID of "DataObject1"> HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 Verify all the mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.4.6> List of mandatory fields for "DataObject1": <ul style="list-style-type: none"> objectType objectID domainURI (Conditional: If system-wide capability "cdmi_domains" is present and true) capabilitiesURI completionStatus mimetype metadata valuerange valuetransferencoding Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> Expected Result: Data object "DataObject1" should be read successfully. <p><Test Case : End></p>	

6.4.2.2 CDMI_DataObject_Read_2

6.4.2.2.1 CDMI_DataObject_Read_2.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_2.1
Test Case Name	dataObjectRead_2R_2.1
Test Case Description	To read a data object with invalid header
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" into container named "TestContainer1". <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: GET <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com Accept: application/cdm_i_object X-CDMI-Specification-Version: 1.0.2 4. Check for HTTP status code: 406 Not Acceptable returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> 5. Expected Result: Data object "DataObject1" should not be read. <p><Test Case : End></p>

6.4.2.2.2 CDMI_DataObject_Read_2.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_2.2
Test Case Name	dataObjectRead_2R_2.2
Test Case Description	To read a data object with invalid header
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" into container named "TestContainer1". <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: GET <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com Accept: application/cdm_i-object X-CDMI-Specification-Version: 1.0.2 4. Check for HTTP status code: 400 Bad Request returned.

<p><Reference: CDMI Specification v1.0.2 – Section: 8.4.7></p> <p>5. Expected Result: Data object "DataObject1" should not be read.</p> <p><Test Case : End></p>

6.4.2.3 CDMI _DataObject_Read_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_4
Test Case Name	dataObjectRead_2R_4
Test Case Description	To read a data object that is not present
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> <p>Request Sample:</p> <pre>GET <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com Accept: application/cdm-object X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> Expected Result: Data object "DataObject1" should not be found. <p><Test Case : End></p>
--

6.4.2.4 CDMI _DataObject_Read_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_5
Test Case Name	dataObjectRead_2R_5
Test Case Description	To read a data object successfully
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference: Login Test Spec Id : 1C_L_1>**
2. Create a data object "DataObject1" in container named "TestContainer1".
<Reference: Test Spec Id: 2C_3>
 Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.1>**
 Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.3>**

 Request Sample:


```
GET <root URI>/TestContainer1/DataObject1 HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-object
X-CDMI-Specification-Version: 1.0.2
```
3. Verify all the mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.6>**
 List of mandatory fields for "DataObject1":
 - objectType
 - objectID
 - domainURI (**Conditional:** If system-wide capability "cdmi_domains" is present and true)
 - capabilitiesURI
 - completionStatus
 - mimetype
 - metadata
 - valuerange
 - valuetransferencoding
4. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 8.4.7>
5. Expected Result: Data object "DataObject1" should be read successfully.

<Test Case : End>

6.4.2.5 CDMI _DataObject_Read_6

6.4.2.5.1 CDMI _DataObject_Read_6.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_6.1
Test Case Name	dataObjectRead_2R_6.1
Test Case Description	To read value with range of data object that has a value transfer encoding of Base64
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Data object capability: cdmi_read_value_range <Reference: CDMI Specification v1.0.2 – Section: 8.4.2>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference: Login Test Spec Id: 1C_L_1>**
2. Create a data object "DataObject1" with valuetransferencoding as "Base64" in container named "TestContainer1"
<Reference: Test Spec Id: 2C_3>
3. Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.1>**
Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.3>**

Request Sample:

GET <root URI>/TestContainer1/DataObject1?value:0-10 HTTP/1.1
Host: cloud.example.com

Accept: application/cdm-object
X-CDMI-Specification-Version: 1.0.2
4. Verify that all mandatory fields exist: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.6>**
List of mandatory fields for "DataObject1": value
5. Check for HTTP status code: **200 OK** returned.
<Reference: CDMI Specification v1.0.2 – Section: 8.4.7>
6. Expected Result: Data object "DataObject1" should be read successfully.

<Test Case : End>

6.4.2.5.2 CDMI_DataObject_Read_6.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_6.2
Test Case Name	dataObjectRead_2R_6.2
Test Case Description	To read value with range of data object that has a value transfer encoding of UTF-8
Pre-Test Dependencies	System-wide capability: cdm_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Data object capability: cdm_read_value_range <Reference: CDMI Specification v1.0.2 – Section: 8.4.2>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. **<Reference: Login Test Spec Id: 1C_L_1>**
2. Create a data object "DataObject1" with valuetransferencoding as UTF-8 in container named "TestContainer1"
<Reference: Test Spec Id: 2C_3>
Request URL: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.1>**
Request Headers: **<Reference: CDMI Specification v1.0.2 – Section: 8.4.3>**

<p>Request Sample:</p> <pre>GET <root URI>/TestContainer1/DataObject1?value:0-10 HTTP/1.1 Host: cloud.example.com Accept: application/cdm-object X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.4.6> List of mandatory fields for "DataObject1": value Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> Expected Result: Data object "DataObject1" should be read successfully. <p><Test Case : End></p>

6.4.2.6 CDMI _DataObject_Read_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_7
Test Case Name	dataObjectRead_2R_7
Test Case Description	To read the data object with valid fields
Pre-Test Dependencies	System-wide capability: cdm_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1>

D E S C R I P T I O N

<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> Create a data object "DataObject1" in container named "TestContainer1" <Reference: Test Spec Id: 2C_3> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> <p>Request Sample:</p> <pre>GET <root URI>/TestContainer1/DataObject1?objectType;objectID;parentURI HTTP/1.1 Host: cloud.example.com Accept: application/cdm-object X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: 8.4.6> List of mandatory fields for "DataObject1": <ul style="list-style-type: none"> objectType objectID parentURI Check for HTTP status code: 200 OK returned.

<p><Reference: CDMI Specification v1.0.2 – Section: 8.4.7></p> <p>6. Expected Result: Data object "DataObject1" should be read successfully.</p> <p><Test Case : End></p>
--

6.4.2.7 CDMI _DataObject_Read_9

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_9
Test Case Name	dataObjectRead_2R_9
Test Case Description	Read data object with invalid header parameter by objectID
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> Create a data object "DataObject1" in container named "TestContainer1" and note its objectID. <Reference: Test Spec Id: 2C_3> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: GET <root URI>/cdmi_objectid/<objectID of "DataObject1"> HTTP/1.1 Host: cloud.example.com Accept: application/cdmi_object X-CDMI-Specification-Version: 1.0.2 Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> Expected Result: "DataObject1" should not be listed. <p><Test Case : End></p>
--

6.4.2.8 CDMI _DataObject_Read_10

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_10
Test Case Name	dataObjectRead_2R_10
Test Case Description	Read a data object with reference

Pre-Test Dependencies	Capability in parent container: cdmi_read_value <Reference: CDMI Specification v1.0.2 – Section: 8.4.2 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: 2C_3> 3. Create a reference data object "DataObject2" of data object "DataObject1" <Reference: Test Spec Id: 2C_17> 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: GET <root URI>/TestContainer1/DataObject2 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 5. Check for HTTP status code: 302 Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> 6. Check for Response Header "Location" value: "/TestContainer1/DataObject1" 7. Expected Result: "DataObject1" should be read. <p><Test Case : End></p>	

6.4.2.9 CDMI_DataObject_Read_11

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_11
Test Case Name	dataObjectRead_2R_11
Test Case Description	Read a data object with valid fields in URL by objectID
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1" and note its objectID. <Reference: Test Spec Id: 2C_3> 	

3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.4.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.4.3**>
 Request Sample:

```
GET <root URI>/cdmi_objectid/<objectID of
"DataObject1">?objectType;objectID;parentURI HTTP/1.1
Host: cloud.example.com

Accept: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2
```
4. Verify that all mandatory fields exist: <Reference: CDMI Specification v1.0.2 – Section: **8.4.6**>
 List of mandatory fields for "DataObject1":
 - objectType
 - objectID
 - parentURI
5. Check for HTTP status code: **200 OK** returned.
 <Reference: CDMI Specification v1.0.2 – Section: **8.4.7**>
6. Expected Result: "DataObject1" should be read successfully.

<Test Case : End>

6.4.2.10 CDMI_DataObject_Read_12

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_12
Test Case Name	dataObjectRead_2R_12
Test Case Description	Read a data object with invalid fields in URL by objectID
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1" and note its objectID. <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: GET <root URI>/cdmi_objectid/<objectID of "DataObject1">?object?Type HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object X-CDMI-Specification-Version: 1.0.2 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> 5. Expected Result: "DataObject1" should not be read. <p><Test Case : End></p>

6.4.2.11 CDMI_DataObject_Read_13

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_13
Test Case Name	dataObjectRead_2R_13
Test Case Description	Read a data object with invalid fields in URL
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: <pre>GET <root URI>/TestContainer1/DataObject1?object?Type HTTP/1.1 Host: cloud.example.com Accept: application/cdm-object X-CDMI-Specification-Version: 1.0.2</pre> 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7> 5. Expected Result: "DataObject1" should not be read. <p><Test Case : End></p>

6.4.2.12 CDMI_DataObject_Read_16

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_16
Test Case Name	dataObjectRead_2R_16
Test Case Description	Read a data object that does not exist by objectID
Pre-Test Dependencies	System-wide capability: cdm-object-access-by-ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> 2. Create a data object named "DataObject1" in container named "TestContainer1" and note its objectID. <Reference: Test Spec Id: 2C_3> 3. Delete above created data object "DataObject1" <Reference: Test Spec Id: 2D_5> 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3> Request Sample: <pre>GET <root URI>/cdmi-objectid/<objectID of "DataObject1"> HTTP/1.1 Host: cloud.example.com</pre> Accept: application/cdm-object

<p>X-CDMI-Specification-Version: 1.0.2</p> <p>5. Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7></p> <p>6. Expected Result: "DataObject1" should not be found.</p> <p><Test Case : End></p>
--

6.4.2.13 CDMI_DataObject_Read_18

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.4
Test Specification Id	2R_18
Test Case Name	dataObjectRead_2R_18
Test Case Description	Read a data object via reference by objectID
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<p><Test Case : Start></p> <p>1. Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1></p> <p>2. Create a reference data object "DataObject2" of data object "DataObject1" by using its objectID. <Reference: Test Id: 2C_18></p> <p>3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.4.3></p> <p>Request Sample:</p> <pre>Get <root URI>/TestContainer1/DataObject2 HTTP/1.1 Host: cloud.example.com Accept: application/cdm-object X-CDMI-Specification-Version: 1.0.2</pre> <p>4. Check for HTTP status code: 302 Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.4.7></p> <p>5. Check for Response Header "Location" value: "/TestContainer1/DataObject1"</p> <p>6. Expected Result: "DataObject1" should be read.</p> <p><Test Case : End></p>
--

6.4.2.14 CDMI_DataObject_Read_19

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 –Section: 8.4
Test Specification Id	2R_19
Test Case Name	dataObjectRead_2R_19
Test Case Description	To create a data object via non-CDMI and read it via CDMI

Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 –Section: 12.1.1 > Data object capability: cdmi_read_value <Reference: CDMI Specification v1.0.2 –Section:8.4.2>
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: NC_2C_1> <p>Request URL: <Reference: CDMI Specification v1.0.2 –Section: 8.4.1> Request Headers: <Reference: CDMI Specification v1.0.2 –Section: 8.4.3> Request Sample:</p> <pre>GET <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 3. Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 –Section: 8.4.7> 4. Expected Result: Data object "DataObject1" should be read successfully. 5. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

6.4.3 CDMI_DataObject_Update

6.4.3.1 CDMI_DataObject_Update_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_1
Test Case Name	dataObjectUpdate_2U_1
Test Case Description	To update a data object by objectID
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > cdmi_object_access_by_ID Note : Object Id should be present and true <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1" and note its objectID.<Test Id: 2C_3> 	

3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.6.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.6.3**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **8.6.4**>
 Request Sample with Body:

```
PUT <root URI>/cdmi_objectid/<objectid of "dataobject1"> HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.2

{
  "metadata" : {
    "Color": "Blue"
  }
}
```
4. Check for HTTP status code: **204 No Content** returned.
 <Reference: CDMI Specification v1.0.2 – Section: **8.6.7**>
5. Expected Result: "dataobject1" should be updated.

<Test Case : End>

6.4.3.2 CDMI_DataObject_Update_2

6.4.3.2.1 CDMI_DataObject_Update_2.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_2.1
Test Case Name	dataObjectUpdate_2U_2.1
Test Case Description	To update a data object with invalid header field
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials.<Reference Test Case Id : **1C_L_1**>
2. Create a data object "dataobject1".<Test Id: **2C_3**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.6.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.6.3**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **8.6.4**>
 Request Sample with Body:

```
PUT <root URI>/TestContainer1/dataobject1 HTTP/1.1
Host: cloud.example.com

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

<pre>X-CDMI-Specification-Version: 1.0.2</pre> <pre>{</pre> <pre>}</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.6.7> 5. Expected Result: "dataobject1" should not be updated. <p><Test Case : End></p>
--

6.4.3.2.2 CDMI_DataObject_Update_2.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_2.2
Test Case Name	dataObjectUpdate_2U_2.2
Test Case Description	To update a data object with invalid header field
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

DESCRIPTION

<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1".<Test Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.6.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.6.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.6.4> Request Sample with Body: <pre>PUT <root URI>/TestContainer1/dataobject1 HTTP/1.1</pre> <pre>Host: cloud.example.com</pre> <pre>Accept: application/cdm-object</pre> <pre>X-CDMI-Specification-Version: 1.0.2</pre> <pre>{</pre> <pre>}</pre> 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.6.7> 5. Expected Result: "dataobject1" should not be updated. <p><Test Case : End></p>
--

6.4.3.2.3 CDMI_DataObject_Update_2.3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_2.3
Test Case Name	dataObjectUpdate_2U_2.3
Test Case Description	To update a data object with invalid header field
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata < Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1".<Test Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.6.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.6.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.6.4> Request Sample with Body: PUT <root URI>/TestContainer1/dataobject1 HTTP/1.1 Host: cloud.example.com Accept: application/cdmi-object Content-Type: application/cdmi-object { } 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.6.7> 5. Expected Result: "dataobject1" should not be updated. <p><Test Case : End></p>	

6.4.3.2.4 CDMI_DataObject_Update_2.4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_2.4
Test Case Name	dataObjectUpdate_2U_2.4
Test Case Description	To update a data object with invalid header field
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata < Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<Reference Test Case Id : **1C_L_1**>
2. Create a data object "dataobject1".<Test Id: **2C_3**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.6.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.6.3**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **8.6.4**>

 Request Sample with Body:


```
PUT <root URI>/TestContainer1/dataobject1 HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdm-object

{
}
```
4. Check for HTTP status code: **400 Bad Request** returned.
 <Reference: CDMI Specification v1.0.2 – Section: **8.6.7**>
5. Expected Result: "dataobject1" should not be updated.

<Test Case : End>

6.4.3.2.5 CDMI_DataObject_Update_2.5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_2.5
Test Case Name	dataObjectUpdate_2U_2.5
Test Case Description	To update a data object with invalid header field
Pre-Test Dependencies	Capability in parent container: cdm_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdm_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<Reference Test Case Id : **1C_L_1**>
2. Create a data object "dataobject1".<Test Id: **2C_3**>
3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: **8.6.1**>
 Request Headers: <Reference: CDMI Specification v1.0.2 – Section: **8.6.3**>
 Request Body: <Reference: CDMI Specification v1.0.2 – Section: **8.6.4**>

 Request Sample with Body:


```
PUT <root URI>/TestContainer1/dataobject1 HTTP/1.1
Host: cloud.example.com

X-CDMI-Specification-Version: 1.0.2
```


<pre>{</pre>
<pre>}</pre>
<ol style="list-style-type: none"> 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.6.7> 5. Expected Result: "dataobject1" should not be updated.
<p><Test Case : End></p>

6.4.3.3 CDMI_DataObject_Update_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_4
Test Case Name	dataObjectUpdate_2U_4
Test Case Description	To update the data object by passing valid fields in the request message body
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata <Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

DESCRIPTION

<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1" <Test Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.6.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.6.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.6.4> Request Sample with Body: <pre>PUT <root URI>/TestContainer1/dataobject1 HTTP/1.1 Host: cloud.example.com Content-Type: application/cdm-object X-CDMI-Specification-Version: 1.0.2 { "mimetype": "text/plain", "metadata": { "Color": "Blue" }, "value": "This is the value of this Data Object" }</pre> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.6.7> 5. Expected Result: Valid fields of "dataobject1" should be updated. <p><Test Case : End></p>

6.4.3.4 CDMI_DataObject_Update_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_5
Test Case Name	dataObjectUpdate_2U_5
Test Case Description	To update data object metadata by passing valid metadata in the request message body
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata < Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : 1C_L_1>
2. Create a data object "dataobject1" <Test Id: **2C_3**>
3. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.3**>
Request Body: <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.4**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/data_object1?metadata HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.2
```

```
{
  "metadata": {
    "Color": "Blue"
  }
}
```

4. Check for HTTP status code: **204 No Content** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **8.6.7**>
5. Expected Result: Valid metadata of "dataobject1" should be updated.

<Test Case : End>

6.4.3.5 CDMI_DataObject_Update_6

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_6
Test Case Name	dataObjectUpdate_2U_6
Test Case Description	To update data object by valid fields in URI
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata < Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects

	<p><Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Check cross-domain privilege for user: cdmi_member_privileges <Reference: CDMI Specification v1.0.2 – Section: 10.1.3> Capability in parent domain: cdmi_create_domain <Reference: CDMI Specification v1.0.2 – Section: 10.2.2> Note: This test case will not run if 'cdmi_domains' & 'cdmi_create_domain' capabilities are not supported and user does not have cross-domain privilege.</p>
--	--

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : 1C_L_1>
2. Create a data object "dataobject1" <Test Id: **2C_3**>
3. Create a domain named "MyNewDomain1" <Test ID: 7.5.1>
4. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.1**>
 Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.3**>
 Request Body: <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.4**>

 Request Sample with Body:


```
PUT <root URI>/TestContainer1/dataobject1?domainURI HTTP/1.1
Host: cloud.example.com

Content-Type: application/cdmi-object
X-CDMI-Specification-Version: 1.0.2

{
    "domainURI" = "/cdmi_domains/MyNewDomain1/"
}
```
5. Check for HTTP status code: **204 No Content** returned.
 <**Reference:** CDMI Specification v1.0.2 – Section: **8.6.7**>
6. Expected Result: Valid fields of "dataobject1" should be updated.

<Test Case : End>

6.4.3.6 CDMI_DataObject_Update_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.6
Test Specification Id	2U_7
Test Case Name	dataObjectUpdate_2U_7
Test Case Description	To update data object with invalid fields in URI
Pre-Test Dependencies	Capability in parent container: cdmi_modify_metadata < Reference: CDMI Specification v1.0.2 – Section: 8.6.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference** Test Case Id : 1C_L_1>
2. Create a data object "dataobject1" <Test Id: **2C_3**>
3. Request URL: <**Reference**: CDMI Specification v1.0.2 – Section: **8.6.1**>
Request Headers: <**Reference**: CDMI Specification v1.0.2 – Section: **8.6.3**>
Request Body: <**Reference**: CDMI Specification v1.0.2 – Section: **8.6.4**>

Request Sample with Body:

```
PUT <root URI>/TestContainer1/dataobject1?mete?data HTTP/1.1
Host: cloud.example.com
```

```
Content-Type: application/cdm-object
X-CDMI-Specification-Version: 1.0.2
```

```
{
  "metadata1": {
    "TestField1": "TestValue1"
  }
}
```

4. Check for HTTP status code: **400 Bad Request** returned.
<**Reference**: CDMI Specification v1.0.2 – Section: **8.6.7**>
5. Expected Result: "dataobject1" should not be updated.

<Test Case : End>

6.4.4 CDMI_DataObject_Delete

6.4.4.1 CDMI_DataObject_Delete_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.8
Test Specification Id	2D_1
Test Case Name	dataObjectDelete_2D_1
Test Case Description	To delete data object by objectID
Pre-Test Dependencies	Capability in parent container: cdm_delete_dataobject < Reference : CDMI Specification v1.0.2 – Section: 8.8.2 > System-wide capability: cdm_object_access_by_ID < Reference : CDMI Specification v1.0.2 – Section: 12.1.1 > cdm_dataobjects < Reference : CDMI Specification v1.0.2 – Section: 12.1.1 >

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1" and note its objectID.<Test Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.8.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.8.3> <p>Request Sample:</p> <pre>DELETE <root URI>/cdmi_objectid/<objectid of "dataobject1"> HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.8.7> 5. Expected Result: "dataobject1" should be deleted. <p><Test Case : End></p>

6.4.4.2 CDMI_DataObject_Delete_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.8
Test Specification Id	2D_2
Test Case Name	dataObjectDelete_2D_2
Test Case Description	To delete data object with invalid header field
Pre-Test Dependencies	Capability in parent container: cdmi_delete_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.8.2 > System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1"<Test Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.8.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.8.3> <p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/dataobject1 HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.8.7>

- Expected Result: "dataobject1" should not be deleted.

<Test Case : End>

6.4.4.3 CDMI _DataObject_Delete_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.8
Test Specification Id	2D_3
Test Case Name	dataObjectDelete_2D_3
Test Case Description	To delete data object that does not exist
Pre-Test Dependencies	Capability in parent container: cdmi_delete_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.8.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

- Set valid login credentials.<**Reference** Test Case Id : **1C_L_1**>
- Create a container named "TestContainer1" <Test Id: **1C_2**>
- Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.8.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **8.8.3**>

Request Sample:

DELETE <root URI>/TestContainer1/dataobject1 HTTP/1.1
Host: cloud.example.com

X-CDMI-Specification-Version: 1.0.2
- Check for HTTP status code: **404 Not Found** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **8.8.7**>
- Expected Result: "dataobject1" should not be found.

<Test Case : End>

6.4.4.4 CDMI _DataObject_Delete_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.8
Test Specification Id	2D_5
Test Case Name	dataObjectDelete_2D_5
Test Case Description	To delete an existing data object
Pre-Test Dependencies	Capability in parent container: cdmi_delete_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.8.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference Test Case Id : 1C_L_1> 2. Create a data object "dataobject1"<Test Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.8.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.8.3> <p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/dataobject1 HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.8.7> 5. Expected Result: "dataobject1" should be deleted. <p><Test Case : End></p>

6.4.4.5 CDMI_DataObject_Delete_6

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 –Section: 8.8
Test Specification Id	2D_6
Test Case Name	dataObjectDelete_2D_6
Test Case Description	To create a data object via non-CDMI and delete it via CDMI
Pre-Test Dependencies	DataObject Capability: cdmi_delete_dataobject < Reference: CDMI Specification v1.0.2 –Section: 8.8.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 –Section: 12.1.1 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id: 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: NC_2C_1> Request URL: <Reference: CDMI Specification v1.0.2 –Section: 8.8.1> Request Headers: <Reference: CDMI Specification v1.0.2 –Section: 8.8.3> <p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/dataobject1 HTTP/1.1 Host: cloud.example.com X-CDMI-Specification-Version: 1.0.2</pre> <ol style="list-style-type: none"> 3. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 –Section: 8.8.7>

4. Expected Result: "dataobject1" should be deleted.
5. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup>

<Test Case : End>

6.5 Server Cleanup Process

After execution of test scripts, the server should be reverted to an initiated state for further testing. To complete this process after execution of each test case, a script will be run to delete the leftover resources produced by that test case.

6.5.1 Container Cleanup

Call Container Delete	Container should be deleted <Reference: 6.3.4 CDMI_Container_Delete_1D_1>
------------------------------	--

6.5.2 Data Object Cleanup

Call Data Object Delete	Data object should be deleted < Reference: 6.4.4 CDMI_DataObject_Delete_2D_1>
--------------------------------	--

7 Non-CDMI Support

Non-CDMI operation is specifically designed for non-CDMI-aware clients. Most of the existing clients are not aware of the CDMI standard. In such cases, the client has been given limited privilege (i.e., to create and delete containers and to create, delete, update, and read data objects) to perform the operation.

7.1 Server Authentication Test

See [6.1 Server Authentication Test](#)

7.2 Capability

See [6.2 Capability](#)

7.3 Non-CDMI Container

7.3.1 Non_CDMI_Container_Create

7.3.1.1 Container_Create_NC_1C_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.3
Test Specification Id	NC_1C_1
Test Case Name	containerCreate_NC_1C_1
Test Case Description	To create a container successfully using non-CDMI content type
Pre-Test Dependencies	Capability in parent container: cdmi_create_container < Reference: CDMI Specification v1.0.2 – Section: 9.3.2 >
<i>DESCRIPTION</i>	
<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Test Case Id: 1C_L_1> Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.3.1> Request Sample: <pre>PUT <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com</pre> Check for HTTP Status Code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 9.3.7> Expected Result: Container should be created successfully. Container cleanup process: <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.3.1.2 Container_Create_NC_1C_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.3
Test Specification Id	NC_1C_2
Test Case Name	containerCreate_NC_1C_2
Test Case Description	To create a container within existing container using non-CDMI content type
Pre-Test Dependencies	Capability in parent container: cdmi_create_container < Reference: CDMI Specification v1.0.2 – Section: 9.3.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id : 1C_L_1> 2. Create a container named "TestContainer1". <Reference: Test Case Id: NC_1C_1 > 3. Create a container named "TestContainer2": 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.3.1> Request Sample: PUT <root URI>/TestContainer1/TestContainer2/ HTTP/1.1 Host: cloud.example.com 5. Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 9.3.7> 6. Expected Result: Container should be created successfully. 7. Container cleanup process: <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.3.1.3 Container_Create_NC_1C_3

7.3.1.3.1 Container_Create_NC_1C_3.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.3
Test Specification Id	NC_1C_3.1
Test Case Name	containerCreate_NC_1C_3.1
Test Case Description	To create a container with an invalid container name.
Pre-Test Dependencies	Capability in parent container: cdmi_create_container < Reference: CDMI Specification v1.0.2 – Section: 9.3.2 >

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Create a container with invalid container name: 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.3.1> Request Sample: PUT <root URI>/cdmi_TestContainer1/ HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 9.3.7> 5. Expected Result: Container should not be created. <p><Test Case : End></p>

7.3.2 Non_CDMI_Container_Delete

7.3.2.1 Container_Delete_NC_1D_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_1
Test Case Name	containerDelete_NC_1D_1
Test Case Description	To delete a container successfully using a non-CDMI content type
Pre-Test Dependencies	Capability on Parent Container: cdmi_delete_container <Reference: CDMI Specification v1.0.2 – Section: 9.7.2>

<i>D E S C R I P T I O N</i>
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Create a container named "TestContainer1" <Reference: Test Case Id: NC_1C_1> 3. Delete container named "TestContainer1": Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample: DELETE <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 5. Expected Result: Container should be deleted successfully. <p><Test Case : End></p>

7.3.2.2 Container_Delete_NC_1D_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_2
Test Case Name	containerDelete_NC_1D_2
Test Case Description	To delete a container using object ID using a non-CDMI content type
Pre-Test Dependencies	System-wide capability: cdmi_object_access_by_ID < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability on Parent Container: cdmi_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.7.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id : 1C_L_1> 2. Create a container named "TestContainer1" and get its object ID <Reference: Test Case Id: NC_1C_1 > 3. Delete container named "TestContainer1": 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample: DELETE <root URI>/cdmi_objectid/<objectid of "TestContainer1">/HTTP/1.1 Host: cloud.example.com 5. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 6. Expected Result: Container should be deleted successfully. <p><Test Case : End></p>	

7.3.2.3 Container_Delete_NC_1D_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_4
Test Case Name	containerDelete_NC_1D_4
Test Case Description	To delete a container that does not exist
Pre-Test Dependencies	Capability on Parent Container: cdmi_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.7.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Delete container named "TestContainer1". 	

- Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1>

Request Sample:

```
DELETE <root URI>/TestContainer1/ HTTP/1.1
Host: cloud.example.com
```

- Check for HTTP status code: **404 Not Found** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.7.7>
- Expected Result: Container cannot be deleted at the desired location.

<Test Case : End>

7.3.2.4 Container_Delete_NC_1D_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_5
Test Case Name	containerDelete_NC_1D_5
Test Case Description	To delete a container that does not exist by objectID
Pre-Test Dependencies	System-wide capability: cdm_i_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Capability on Parent Container: cdm_i_delete_container <Reference: CDMI Specification v1.0.2 – Section: 9.7.2>

DESCRIPTION

<Test Case : Start>

- Set valid login credentials. <Reference: Test Case Id: 1C_L_1>
- Create a container named "TestContainer1" and get its object ID
<Reference: Test Case Id: NC_1C_1 >
- Delete the container "TestContainer1" <Reference: Test Case Id: NC_1D_1>
- Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1>

Request Sample:

```
DELETE <root URI>/cdm_i_objectid/<objectid of "TestContainer1">/
HTTP/1.1
Host: cloud.example.com
```

- Check for HTTP status code: **404 Not Found** returned.
<Reference: CDMI Specification v1.0.2 – Section: 9.7.7>
- Expected Result: Container cannot be deleted successfully.

<Test Case : End>

7.3.2.5 Container_Delete_NC_1D_6

7.3.2.5.1 Container_Delete_NC_1D_6.1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_6.1
Test Case Name	containerDelete_NC_1D_6.1
Test Case Description	To delete those containers which are system defined and reserved
Pre-Test Dependencies	Capability on Parent Container: cdm_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.7.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id : 1C_L_1> 2. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample: DELETE <root URI>/cdmi_capabilities/ HTTP/1.1 Host: cloud.example.com 3. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 4. Expected Result: Container cannot be deleted successfully. <p><Test Case : End></p>	

7.3.2.5.2 Container_Delete_NC_1D_6.2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_6.2
Test Case Name	containerDelete_NC_1D_6.2
Test Case Description	To delete those containers which are system defined and reserved.
Pre-Test Dependencies	Capability on Parent Container: cdm_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.7.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample: DELETE <root URI>/cdmi_objectid/ HTTP/1.1 Host: cloud.example.com 3. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 	

4. Expected Result: Container cannot be deleted successfully.

<Test Case : End>

7.3.2.5.3 Container_Delete_NC_1D_6.3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_6.3
Test Case Name	containerDelete_NC_1D_6.3
Test Case Description	To delete those containers which are system defined and reserved
Pre-Test Dependencies	Capability on Parent Container: cdmi_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.7.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample:Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample: DELETE <root URI>/cdmi_snapshots/ HTTP/1.1 Host: cloud.example.com 3. Check for HTTP status code: 400 Bad Request or 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 4. Expected Result: Container cannot be deleted successfully. <p><Test Case : End></p>	

7.3.2.6 Container_Delete_NC_1D_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_7
Test Case Name	containerDelete_NC_1D_7
Test Case Description	To delete a container without passing trailing slash in URI
Pre-Test Dependencies	Capability on Parent Container: cdmi_delete_container < Reference: CDMI Specification v1.0.2 – Section: 9.7.2 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Create a container named "TestContainer1" <Reference: Test Case Id: NC_1C_1 > 3. Delete container named "TestContainer1". 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Sample: <pre>DELETE <root URI>/TestContainer1 HTTP/1.1 Host: cloud.example.com</pre> 5. Check for HTTP status code: 301 Moved Permanently returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 6. Expected Result: Container should not be deleted and "Location" header should be returned with the value of "/TestContainer1/". 7. Container cleanup process: <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>

7.3.2.7 Container_Delete_NC_1D_8

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 9.7
Test Specification Id	NC_1D_8
Test Case Name	containerDelete_NC_1D_8
Test Case Description	To create a container via CDMI and delete it via Non-CDMI
Pre-Test Dependencies	Container Capability: cdmi_delete_container <Reference: CDMI Specification v1.0.2 – Section: 9.7.2>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Case Id: 1C_L_1> 2. Create a container named "TestContainer1"<Reference: Test Id: 1C_6> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 9.7.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 9.6.3> Request Sample: <pre>DELETE <root URI>/TestContainer1/ HTTP/1.1 Host: cloud.example.com</pre> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 9.7.7> 5. Expected Result: 'TestContainer1' should be deleted successfully. <p><Test Case : End></p>

7.4 Non CDMI Data Object

7.4.1 Non_CDMI_DataObject_Create

7.4.1.1 DataObject_Create_NC_2C_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.3
Test Specification Id	NC_2C_1
Test Case Name	dataobjectCreate_NC_2C_1
Test Case Description	To create a data object successfully using non-CDMI content type
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.3.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a container named "TestContainer1". < Reference: Test Case Id: NC_1C_1> 3. Create a data object named "data_object1": 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.3.1> Request Header: <Reference: CDMI Specification v1.0.2 – Section: 8.3.3> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.3.4> Request Sample with Body: <pre>PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host: cloud.example.com Content-Type: text/plain This is the value of this Data Object.</pre> 5. Check for HTTP status code: 201 Created returned. <Reference: CDMI Specification v1.0.2 – Section: 8.3.7> 6. Expected Result: Data object should be created and CDMI GET should return valuetransferencoding of base64. 7. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.1.2 DataObject_Create_NC_2C_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.3
Test Specification Id	NC_2C_2
Test Case Name	dataobjectCreate_NC_2C_2
Test Case Description	To create a data object successfully with charset = utf-8 using non-CDMI content type
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.3.2 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials. <**Reference:** Test Spec Id: 1C_L_1>
2. Create a container named "TestContainer1". < **Reference:** Test Case Id: NC_1C_1>
3. Create a data object named "data_object1":
4. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.3.1**>
Request Header: <**Reference:** CDMI Specification v1.0.2 – Section: **8.3.3**>
Request Body:<**Reference:** CDMI Specification v1.0.2 – Section: **8.3.4**>
Request Sample with Body:

```
PUT <root URI>/TestContainer1/data_object1 HTTP/1.1
Host:cloud.example.com

Content-Type: text/plain;charset=utf-8

This is the value of this Data Object.
```
5. Check for HTTP status code: **201 Created** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **8.3.7**>
6. Expected Result: Data object "data_object1"should be created and CDMI GET should return valuetransferencoding of utf-8.
7. Container cleanup process : <**Reference:** 7.5.1 Non-CDMI Container Cleanup>

<Test Case : End>

7.4.1.3 DataObject_Create_NC_2C_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.3
Test Specification Id	NC_2C_4
Test Case Name	dataobjectCreate_NC_2C_4
Test Case Description	To create a data object within a container that doesn't exist

Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.3.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a data object named "data_object1": 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.3.1> Request Header: <Reference: CDMI Specification v1.0.2 – Section: 8.3.3> Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.3.4> <p>Request Sample with Body:</p> <pre>PUT <root URI>/TestContainer1/data_object1 HTTP/1.1 Host:cloud.example.com Content-Type: text/plain This is the value of this Data Object.</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 400 Bad Request returned. <Reference: CDMI Specification v1.0.2 – Section: 8.3.7> 5. Expected Result: Data object cannot be created. <p><Test Case : End></p>	

7.4.1.4 DataObject_Create_NC_2C_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.3
Test Specification Id	NC_2C_5
Test Case Name	dataobjectCreate_NC_2C_5
Test Case Description	To create a data object within a container created using CDMI content type
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.3.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1> 2. Create a container named "TestContainer1" <Reference: Test Spec Id: 1C_1 > 3. Create a data object named "data_object1": 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.3.1> 	

Request Header: <Reference: CDMI Specification v1.0.2 – Section: 8.3.3>
 Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.3.4>
 Request Sample with Body:
 PUT <root URI>/TestContainer1/data_object1 HTTP/1.1
 Host:cloud.example.com
 Content-Type: text/plain
 This is the value of this Data Object.

5. Check for HTTP status code: 201 Created returned.
 <Reference: CDMI Specification v1.0.2 – Section: 8.3.7>
6. Expected Result: Data object should be created.
7. Container cleanup process : <Reference: 6.5.1 Container Cleanup>

<Test Case : End>

7.4.1.5 DataObject_Create_NC_2C_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.3
Test Specification Id	NC_2C_7
Test Case Name	dataobjectCreate_NC_2C_7
Test Case Description	To create a data object with incorrect header field
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Capability in parent container: cdmi_create_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.3.2>

DESCRIPTION

<Test Case : Start>

1. Set valid login credentials. <Reference: Test Spec Id: 1C_L_1>
2. Create a container named "TestContainer1". < Reference: Test Case Id: NC_1C_1>
3. Create a data object named "data_object1":
4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.3.1>
 Request Body:<Reference: CDMI Specification v1.0.2 – Section: 8.3.4>
 Request Sample with Body:
 PUT <root URI>/TestContainer1/data_object1 HTTP/1.1
 This is the value of this Data Object.
5. Check for HTTP status code: **400 Bad Request** returned.
 <Reference: CDMI Specification v1.0.2 – Section: 8.3.7>
6. Expected Result: Data object cannot be created.
7. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup>

<Test Case : End>

7.4.2 Non_CDMI_DataObject_Read

7.4.2.1 DataObject_Read_NC_2R_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_1
Test Case Name	dataobjectRead_NC_2R_1
Test Case Description	To read an existing data object successfully using a non-CDMI content type
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Data object capability: cdmi_read_value < Reference: CDMI Specification v1.0.2 – Section: 8.5.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: NC_2C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Sample: GET <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 8.5.7> 5. Expected Result: Data object should be read successfully. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.2.2 DataObject_Read_NC_2R_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_2
Test Case Name	dataobjectRead_NC_2R_2
Test Case Description	To read an existing data object via objectID successfully using a non-CDMI content type
Pre-Test Dependencies	System-wide capability: <ul style="list-style-type: none"> • cdmi_dataobjects • cdmi_object_access_by_ID < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Data object capability: cdmi_read_value < Reference: CDMI Specification v1.0.2 – Section: 8.5.2 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1" and read the object Id of DataObject1. <Reference: Test Spec Id: NC_2C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Sample: GET <root URI>/cdmi_objectid/<objectid of "DataObject1"> HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 – Section: 8.5.7> 5. Expected Result: Data object should be read successfully. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>

7.4.2.3 DataObject_Read_NC_2R_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_3
Test Case Name	dataobjectRead_NC_2R_3
Test Case Description	To read a particular range of data object successfully using a non-CDMI content type
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Data object capability: cdmi_read_value_range <Reference: CDMI Specification v1.0.2 – Section: 8.5.2>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id: 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: NC_2C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.5.3> Request Sample: GET <root URI>/Test_Container1/DataObject1 HTTP/1.1 Host: cloud.example.com Range: bytes=0-10 4. Check for HTTP status code: 206 Partial Content returned.

<p><Reference: CDMI Specification v1.0.2 – Section: 8.5.7></p> <p>5. Expected Result: Data object should be read successfully.</p> <p>6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup></p> <p><Test Case : End></p>

7.4.2.4 DataObject_Read_NC_2R_5

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_5
Test Case Name	dataobjectRead_NC_2R_5
Test Case Description	To read a data object that does not exist
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Data object capability: cdmi_read_value <Reference: CDMI Specification v1.0.2 – Section: 8.5.2 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> Create a container named "TestContainer1".<Reference: Test Spec Id: NC_1C_1 > Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Sample: GET <root URI>/Test_Container1/DataObject1 HTTP/1.1 Host: cloud.example.com Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.5.7> Expected Result: Data object should not be read successfully. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.2.5 DataObject_Read_NC_2R_6

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_6
Test Case Name	dataobjectRead_NC_2R_6
Test Case Description	To read a data object that is a reference to a CDMI data object
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Data object capability: cdmi_read_value <Reference: CDMI Specification v1.0.2 – Section: 8.5.2 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a reference data object "DataObject2" of data object "DataObject1". <Reference: Test Id: 2C_17> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Sample: GET <root URI>/Test_Container1/DataObject2 HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 302 Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.5.7> 5. Check for Response Header "Location" value: "/Test_Container1/DataObject1" 6. Expected Result: Data object can be read successfully. 7. Container cleanup process : <Reference: 6.5.1 Container Cleanup> <p><Test Case : End></p>

7.4.2.6 DataObject_Read_NC_2R_7

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_7
Test Case Name	dataobjectRead_NC_2R_7
Test Case Description	To read a data object (with trailing slash) that does not exist
Pre-Test Dependencies	System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section: 12.1.1> Data object capability: cdmi_read_value <Reference: CDMI Specification v1.0.2 – Section: 8.5.2>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a container named "TestContainer1".<Reference: Test Spec Id: NC_1C_1 > 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Sample: GET <root URI>/Test_Container1/DataObject1/ HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.5.7> 5. Expected Result: Data object should not be read successfully. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup>

<Test Case : End>

7.4.2.7 DataObject_Read_NC_2R_9

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.5
Test Specification Id	NC_2R_9
Test Case Name	dataobjectRead_NC_2R_9
Test Case Description	To read a non-existing data object by objectID
Pre-Test Dependencies	System-wide capability: <ul style="list-style-type: none"> • cdm_dataobjects • cdm_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section: 12.1.1 > Data object capability: cdm_read_value <Reference: CDMI Specification v1.0.2 – Section: 8.5.2 >
D E S C R I P T I O N	
<Test Case : Start>	
<ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1" and read the object Id of DataObject1. <Reference: Test Spec Id: NC_2C_1> 3. Delete data object "DataObject1" <Reference: Test Spec Id: NC_2D_1> 4. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.5.1> Request Sample: GET <root URI>/cdmi_objectid/<objectid of "DataObject1"> HTTP/1.1 Host: cloud.example.com 5. Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.5.7> 6. Expected Result: Data object should not be read successfully. 7. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> 	
<Test Case : End>	

7.4.2.8 DataObject_Read_NC_2R_10

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 –Section: 8.5
Test Specification Id	NC_2R_10
Test Case Name	dataobjectRead_NC_2R_10
Test Case Description	To create a data object via CDMI and read it via Non-CDMI
Pre-Test Dependencies	System-wide capability: cdm_dataobjects <Reference: CDMI Specification v1.0.2 –Section: 12.1.1 > Data object capability: cdm_read_value <Reference: CDMI Specification v1.0.2 –Section: 8.5.2 >

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials. <Reference: Login Test Spec Id : 1C_L_1> 2. Create a data object "DataObject1" in container named "TestContainer1". <Reference: Test Spec Id: 2C_3> 3. Request URL: <Reference: CDMI Specification v1.0.2 –Section: 8.5.1> Request Sample: GET <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 200 OK returned. <Reference: CDMI Specification v1.0.2 –Section: 8.5.7> 5. Expected Result: Data object should be read successfully. 6. Container cleanup process : <Reference: 6.5.1 Container Cleanup> <p><Test Case : End></p>

7.4.3 Non_CDMI_DataObject_Update

7.4.3.1 DataObject_Update_NC_2U_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.7
Test Specification Id	NC_2U_1
Test Case Name	dataobjectUpdate_NC_2U_1
Test Case Description	To update a data object by modifying its value.
Pre-Test Dependencies	Capability in data object: cdmi_modify_value <Reference: CDMI Specification v1.0.2 – Section: 8.7.2> System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section 12.1.1>

D E S C R I P T I O N
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a data object "DataObject1" <Reference: Test Id: NC_2C_1 > 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.7.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.7.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.7.4> Request Sample with Body: PUT <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com Content-Type: text/plain

This is the updated values of this data object.

4. Check for HTTP status code: **204 No Content** returned.
<Reference: CDMI Specification v1.0.2 – Section: **8.7.7**>
5. Expected Result: DataObject1 should be updated.
6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup>

<Test Case : End>

7.4.3.2 DataObject_Update_NC_2U_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.7
Test Specification Id	NC_2U_2
Test Case Name	dataobjectUpdate_NC_2U_2
Test Case Description	To update a data object via objectID using Non CDMI Content Type
Pre-Test Dependencies	Capability in data object: cdmi_modify_value <Reference: CDMI Specification v1.0.2 – Section: 8.7.2 > System-wide capability: <ul style="list-style-type: none"> • cdmi_dataobjects • cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a data object "DataObject1" <Reference: Test Id: NC_2C_1 > 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.7.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.7.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.7.4> <p>Request Sample with Body:</p> <pre>PUT <root URI>/cdmi_objectid/<objectid of "DataObject1"> HTTP/1.1 Host: cloud.example.com Content-Type: text/plain</pre> <p style="text-align: center;">This is the updated values of this data object.</p> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.7.7> 5. Expected Result: Data object "DataObject1" should be updated. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.3.3 DataObject_Update_NC_2U_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.7
Test Specification Id	NC_2U_3
Test Case Name	dataobjectUpdate_NC_2U_3
Test Case Description	To update a data object by modifying specific value range.
Pre-Test Dependencies	Capability in data object: cdmi_modify_value_range < Reference: CDMI Specification v1.0.2 – Section: 8.7.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a data object "DataObject1". <Reference: Test Id: NC_2C_1 > 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.7.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.7.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.7.4> <p>Request Sample with Body:</p> <pre>PUT <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com Content-Type: text/plain Content-Range: bytes=21-24 that</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.7.7> 5. Expected Result: Data object "DataObject1" should be updated. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.3.4 DataObject_Update_NC_2U_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.7
Test Specification Id	NC_2U_4
Test Case Name	dataobjectUpdate_NC_2U_4
Test Case Description	To update a data object by modifying specific value range by objectID using Non CDMI Content Type

<p>Pre-Test Dependencies</p>	<p>Capability in data object: cdmi_modify_value_range <Reference: CDMI Specification v1.0.2 – Section: 8.7.2> System-wide capability: cdmi_dataobjects cdmi_object_access_by_ID <Reference: CDMI Specification v1.0.2 – Section 12.1.1></p>
<p>D E S C R I P T I O N</p>	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id: 1C_L_1> 2. Create a data object "DataObject1" and note its objectId. <Reference: Test Id: NC_2C_1 > 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.7.1> Request Headers: <Reference: CDMI Specification v1.0.2 – Section: 8.7.3> Request Body: <Reference: CDMI Specification v1.0.2 – Section: 8.7.4> Request Sample with Body: PUT <root URI>/cdmi_objectid/<objectId of "dataobject1"> HTTP/1.1 Host: cloud.example.com Content-Type: text/plain Content-Range: bytes=21-24 that 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.7.7> 5. Expected Result: "DataObject1" should be updated successfully. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.3.5 DataObject_Update_NC_2U_7

<p>CDMI Specification Reference</p>	<p>Reference: CDMI Specification v1.0.2 – Section: 8.7</p>
<p>Test Specification Id</p>	<p>NC_2U_7</p>
<p>Test Case Name</p>	<p>dataobjectUpdate_NC_2U_7</p>
<p>Test Case Description</p>	<p>To update a data object that is a reference to a CDMI data object.</p>
<p>Pre-Test Dependencies</p>	<p>Capability in data object: cdmi_modify_value <Reference: CDMI Specification v1.0.2 – Section: 8.7.2> System-wide capability: cdmi_dataobjects <Reference: CDMI Specification v1.0.2 – Section 12.1.1></p>

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference:** Test Case Id : 1C_L_1>
2. Create a reference data object "DataObject2" of data object "DataObject1" <**Reference:** Test Id: 2C_17>
3. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.7.1**>
Request Headers: <**Reference:** CDMI Specification v1.0.2 – Section: **8.7.3**>
Request Body: <**Reference:** CDMI Specification v1.0.2 – Section: **8.7.4**>

Request Sample with Body:

PUT <root URI>/TestContainer1/DataObject2 HTTP/1.1
Host: cloud.example.com

Content-Type: text/plain

This is the updated values of this data object.
4. Check for HTTP status code: **302 Found** returned.
<**Reference:** CDMI Specification v1.0.2 – Section: **8.7.7**>
5. Expected Result: Data object should be updated.
6. Container cleanup process : <**Reference:** 6.5.1 Container Cleanup>

<Test Case : End>

7.4.4 Non_CDMI_DataObject_Delete

7.4.4.1 DataObject_Delete_NC_2D_1

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.9
Test Specification Id	NC_2D_1
Test Case Name	dataobjectDelete_NC_2D_1
Test Case Description	To delete data object successfully using a non-CDMI content type
Pre-Test Dependencies	Data object capability: cdmi_delete_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.9.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >

D E S C R I P T I O N

<Test Case : Start>

1. Set valid login credentials.<**Reference:** Test Case Id : 1C_L_1>
2. Create a data object "DataObject1" in container "TestContainer1".
<**Reference:** Test Case Id: NC_2C_1 >
3. Request URL: <**Reference:** CDMI Specification v1.0.2 – Section: **8.9.1**>

<p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.9.7> 5. Expected Result: Data object should be deleted. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>
--

7.4.4.2 DataObject_Delete_NC_2D_2

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.9
Test Specification Id	NC_2D_2
Test Case Name	dataobjectDelete_NC_2D_2
Test Case Description	To delete data object by objectId using a non-CDMI content type
Pre-Test Dependencies	<p>Capability in parent container: cdmi_delete_dataobject <Reference: CDMI Specification v1.0.2 – Section: 8.9.2></p> <p>System-wide capability:</p> <ul style="list-style-type: none"> • cdmi_dataobjects • cdmi_object_access_by_ID <p><Reference: CDMI Specification v1.0.2 – Section: 12.1.1></p>

DESCRIPTION

<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a data object "DataObject1" in container "TestContainer1" and note its objectId. <Reference: Test Case Id: NC_2C_1 > 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.9.1> <p>Request Sample:</p> <pre>DELETE <root URI>/cdmi_objectid/<objectid of "DataObject1"> HTTP/1.1 Host: cloud.example.com</pre> <ol style="list-style-type: none"> 4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 – Section: 8.9.7> 5. Expected Result: Data object should be deleted. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>
--

7.4.4.3 DataObject_Delete_NC_2D_3

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 – Section: 8.9
Test Specification Id	NC_2D_3
Test Case Name	dataobjectDelete_NC_2D_3
Test Case Description	To delete data object that does not exist
Pre-Test Dependencies	Capability in parent container: cdmi_delete_dataobject < Reference: CDMI Specification v1.0.2 – Section: 8.9.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 – Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a container named "TestContainer1". <Reference: Test Spec Id: NC_1C_1> 3. Request URL: <Reference: CDMI Specification v1.0.2 – Section: 8.9.1> Request Sample: DELETE <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com 4. Check for HTTP status code: 404 Not Found returned. <Reference: CDMI Specification v1.0.2 – Section: 8.9.7> 5. Expected Result: Data object cannot be deleted. 6. Container cleanup process : <Reference: 7.5.1 Non-CDMI Container Cleanup> <p><Test Case : End></p>	

7.4.4.4 DataObject_Delete_NC_2D_4

CDMI Specification Reference	Reference: CDMI Specification v1.0.2 –Section: 8.9
Test Specification Id	NC_2D_4
Test Case Name	dataobjectDelete_NC_2D_4
Test Case Description	To create a data object via CDMI and delete it via Non-CDMI
Pre-Test Dependencies	Data object capability: cdmi_delete_dataobject < Reference: CDMI Specification v1.0.2 –Section: 8.9.2 > System-wide capability: cdmi_dataobjects < Reference: CDMI Specification v1.0.2 –Section: 12.1.1 >
D E S C R I P T I O N	
<p><Test Case : Start></p> <ol style="list-style-type: none"> 1. Set valid login credentials.<Reference: Test Case Id : 1C_L_1> 2. Create a data object "DataObject1" in container "TestContainer1". <Reference: Test Case Id: 2C_3 > 	

<p>3. Request URL: <Reference: CDMI Specification v1.0.2 –Section: 8.9.1></p> <p>Request Sample:</p> <pre>DELETE <root URI>/TestContainer1/DataObject1 HTTP/1.1 Host: cloud.example.com</pre> <p>4. Check for HTTP status code: 204 No Content returned. <Reference: CDMI Specification v1.0.2 –Section: 8.9.7></p> <p>5. Expected Result: Data object should be deleted.</p> <p>6. Container cleanup process : <Reference: 6.5.1 Container Cleanup></p> <p><Test Case : End></p>
--

7.5 Server Cleanup Process

7.5.1 Non-CDMI Container Cleanup

Call Container Delete	Container created using Non CDMI Content Type should be deleted <Reference: 7.3.2.1 Container_Delete_NC_1D_1>
------------------------------	--

7.5.2 Non-CDMI Data Object Cleanup

Call Data Object Delete	Data object created using Non CDMI Content Type should be deleted < Reference: 7.4.4.1 DataObject_Delete_NC_2D_1>
--------------------------------	--

8 Profile-Based Support

SNIA has published profiles in the CDMI international standard, which an implementation can support to provide specific functionalities. This specification uses these profiles to scope the testing of CDMI implementations.

8.1 Basic Service-Based Profile

See http://snia.org/sites/default/files/CDMI_Profile-BasicStorageServiceV1_0e.pdf

8.1.1 Mandatory Tests

The mandatory tests include the following:

- Container Create using a Non-CDMI Content Type - See [7.3.1 Non_CDMI_Container_Create](#)
- Container Read using CDMI Content Type - See [6.3.2 CDMI Container Read](#)
- Container Delete using a Non-CDMI Content Type - See [7.3.2 Non_CDMI_Container_Delete](#)
- Data Object Create using a Non-CDMI Content Type - See [7.4.1 Non_CDMI_DataObject_Create](#)
- Data Object Read using a Non-CDMI Content Type - See [7.4.2 Non_CDMI_DataObject_Read](#)
- Data Object Update using a Non-CDMI Content Type - See [7.4.3 Non_CDMI_DataObject_Update](#)
- Data Object Delete using a Non-CDMI Content Type - See [7.4.4 Non_CDMI_DataObject_Delete](#)

8.1.2 Optional Tests

The optional tests include the following:

- Container Create using CDMI Content Type - See [6.3.1 CDMI Container Create](#)
- Container Update using CDMI Content Type - See [6.3.3 CDMI Container Update](#)
- Container Delete using CDMI Content Type - See [6.3.4 CDMI Container Delete](#)
- Data Object Create using CDMI Content Type - See [6.4.1 CDMI DataObject Create](#)
- Data Object Read using CDMI Content Type - See [6.4.2 CDMI DataObject Read](#)
- Data Object Update using CDMI Content Type - See [6.4.3 CDMI DataObject Update](#)
- Data Object Delete using CDMI Content Type - See [6.4.4 CDMI DataObject Delete](#)

8.2 Self -Storage Management Profile

See <http://snia.org/sites/default/files/Self-Storage%20Management%20Profile%20v1.0e.pdf>

8.2.1 Mandatory Tests

The mandatory tests include the following:

- Container Create using CDMI Content Type - See [6.3.1 CDMI Container Create](#)
- Container Read using CDMI Content Type - See [6.3.2 CDMI Container Read](#)
- Container Update using CDMI Content Type - See [6.3.3 CDMI Container Update](#)
- Container Delete using CDMI Content Type - See [6.3.4 CDMI Container Delete](#)

8.2.2 Optional Tests

The optional tests include the following:

- Data Object Create using CDMI Content Type - See [6.4.1 CDMI DataObject Create](#)
- Data Object Read using CDMI Content Type - See [6.4.2 CDMI DataObject Read](#)
- Data Object Update using CDMI Content Type - See [6.4.3 CDMI DataObject Update](#)
- Data Object Delete using CDMI Content Type - See [6.4.4 CDMI DataObject Delete](#)

9 Test Standards Conventions

The Test Case ID should be in the following format:

<Feature Identifier (optional)><Object Identifier><Operation Identifier>_<TestCaseCount>

Where:

- **<Feature Identifier (optional)>** represents features. This field is optional and will be used for the following test cases:

Feature	Feature Identifier
Non-CDMI Test Cases	NC

- **<Object Identifier>** is a numeric value and should be used as follows:

Object Name	Object Identifier
Container	1
Data Object	2
Capability	3

- **<Operation Identifier>** is a character set that represents what operation this test case will perform and should be used as follows:

Operation Name	Operation Identifier
Create	C
Read	R
Update	U
Delete	D