ETSI TS 103 142 V1.1.1 (2013-04)

Technical Specification

CLOUD;
Test Descriptions for Cloud Interoperability

Reference

DTS/CLOUD-0014_tests_Descript

Keywords

CLOUD, interoperability, testing

ETSI

650 Route des Lucioles
F-06921 Sophia Antipolis Cedex - FRANCE

Tel.: +33 4 92 94 42 00 Fax: +33 4 93 65 47 16

Siret N° 348 623 562 00017 - NAF 742 C
Association à but non lucratif enregistrée à la
Sous-Préfecture de Grasse (06) N° 7803/88

Important notice

Copyright Notification

Contents

Intellectual Property Rights

IPRs essential or potentially essential to the present document may have been declared to ETSI. The information pertaining to these essential IPRs, if any, is publicly available for **ETSI members and non-members**, and can be found in ETSI SR 000 314: *"Intellectual Property Rights (IPRs); Essential, or potentially Essential, IPRs notified to ETSI in respect of ETSI standards"*, which is available from the ETSI Secretariat. Latest updates are available on the ETSI Web server (http://ipr.etsi.org).

Pursuant to the ETSI IPR Policy, no investigation, including IPR searches, has been carried out by ETSI. No guarantee can be given as to the existence of other IPRs not referenced in ETSI SR 000 314 (or the updates on the ETSI Web server) which are, or may be, or may become, essential to the present document.

Foreword

This Technical Specification (TS) has been produced by ETSI Technical Committee CLOUD (CLOUD).

1 Scope

The present document specifies Interoperability Test Descriptions (TDs) for OCCI and CDMI standards. The Test Descriptions cover the OCCI and CDMI protocol specifications where relevant and more specifically:

1) OCCI interoperability testing, to prove that end-to-end functionality is as required by the standard.

2) CDMI interoperability testing, to prove that end-to-end functionality is as required by the standard.

3) OCCI + CDMI interworking testing, to prove that end-to-end functionality is as required by the standards.

2 References

References are either specific (identified by date of publication and/or edition number or version number) or non-specific. For specific references, only the cited version applies. For non-specific references, the latest version of the referenced document (including any amendments) applies.

Referenced documents which are not found to be publicly available in the expected location might be found at http://docbox.etsi.org/Reference.

NOTE: While any hyperlinks included in this clause were valid at the time of publication ETSI cannot guarantee their long term validity.

2.1 Normative references

The following referenced documents are necessary for the application of the present document.

[1] OGF GFD.183: "Open Cloud Computing Interface - Core".

[2] OGF GFD.184: "Open Cloud Computing Interface - Infrastructure".

[3] OGF GFD.185: "Open Cloud Computing Interface - RESTful HTTP Rendering".

[4] ISO/IEC 17826: "Information technology -- Cloud Data Management Interface (CDMI)".

2.2 Informative references

The following referenced documents are not necessary for the application of the present document but they assist the user with regard to a particular subject area.

[i.1] IETF RFC 2046: "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types".

[i.2] IETF RFC 2616: "Hypertext Transfer Protocol -- HTTP/1.1".

3 Abbreviations

For the purposes of the present document, the abbreviations given in GFD.183 [1], GFD.184 [2], GFD.185 [3], ISO/IEC 17826 [4] and the following apply:

SUT	System Under Test

4 Conventions

4.1 Interoperability test process

4.1.1 Introduction

The goal of interoperability testing is to check that services implemented according to protocol specifications are able to interwork and to provide at least the mandatory features specified in the protocol specification. In addition, optional features may be checked when all services involved in a test support them.

Detailed protocol conformance checks may be performed during the interoperability test sessions but are not the focus of the interoperability test event.

The test session will be mainly executed between two systems from different vendors. For some test descriptions, it may be necessary to have more than two systems involved. The information about the test configuration like the number of systems or the roles required are indicated in the test description tables 2 to 10.

4.1.2 Test description proforma

The test descriptions are provided in proforma tables. The test description header specifies a unique test identifier, the test objective, the test configuration to be used and references to the protocol specification(s). The pre-condition row defines conditions that need to apply before starting the test.

The following different types of test operator actions are considered during the test execution:

- A **stimulus** corresponds to an event that enforces an SUT to proceed with a specific protocol action, like sending a message.
- A **verify** consists of verifying that the SUT behaves according to the expected behaviour (for instance the SUT behaviour shows that it receives the expected message).
- A **configure** corresponds to an action to modify the SUT configuration.
- A **check** ensures the receipt of protocol messages on reference points, with valid content. This "check" event type corresponds to the interoperability testing with conformance check method.

For the execution of the interoperability test sessions, the following conventions apply:

- Every 'Check' step of a test description should be performed by verifying a trace created with a monitoring tool (see clause 'Tooling' below) and may be skipped due to time restrictions.

4.2 Tooling

- Participant will use their own tools (e.g. tcpdump, wireshark, ngrep) for logging and analyzing messages for the "check" purposes.
- Participants will be given the opportunity to upload their log files to a central server for later offline conformance review.
- Except for the "check" events, the verification of the message conformity is not part of the Interoperability test process.

4.3 Test Description naming convention

Table 1: TD naming convention

TD/<root>/<gr1>/<gr2>/<nnn>		
<root> = root	OCCI CDMI INTER	Open Cloud Computing Interface Cloud Data Management Interface Multi-specification interworking
<gr1> = outer group	CORE INFRA DATA CONTAINER DOMAIN QUEUE CAPABILITIES OCCI+CDMI	OCCI Core model OCCI Infrastructure model CDMI Data Objects CDMI Container Objects CDMI Domain Objects CDMI Queue Objects CDMI Capabilities OCCI + CDMI Interworking
<gr2> = inner group	DISCOVERY CREATE READ UPDATE DELETE MISC	Resource discovery Resource creation Resource reading Resource update Resource deletion Miscellaneous functions
<nnn> = sequential number		001 to 999

4.4 Test Summary - Mandatory Tests

4.4.1 OCCI Mandatory Tests

Table 2: OCCI Mandatory Tests

1	TD/OCCI/CORE/DISCOVERY/001	Retrieving all OCCI Categories supported by the OCCI Server

4.4.2 CDMI Mandatory Tests

Table 3: CDMI Mandatory Tests

1	TD/CDMI/CAPABILITIES/READ/001	Retrieve root CDMI Capability Object
2	TD/CDMI/CAPABILITIES/READ/002	List children of the root CDMI Capability Object
3	TD/CDMI/CAPABILITIES/READ/003	Read capabilities field from existing CDMI Capability Object
4	TD/CDMI/CAPABILITIES/READ/004	Retrieve the Capabilities of a CDMI object

4.5 Test Summary - Optional Tests

4.5.1 OCCI Optional Tests

Table 4: OCCI Core Optional Tests

	TD/OCCI/CORE/DISCOVERY/002	Retrieving the OCCI Categories with an OCCI Category filter from the OCCI Server
2	TD/OCCI/CORE/CREATE/001	Create an OCCI Resource
3	TD/OCCI/CORE/CREATE/002	Create an OCCI Resource with an OCCI Mixin
4	TD/OCCI/CORE/CREATE/003	Create an OCCI Resource with an OCCI Link to an existing OCCI Resource
5	TD/OCCI/CORE/CREATE/004	Create an OCCI Link
6	TD/OCCI/CORE/CREATE/005	Create an OCCI Link with an OCCI Mixin
7	TD/OCCI/CORE/CREATE/006	Add an OCCI Mixin definition
8	TD/OCCI/CORE/READ/001	Retrieve the URLs of all OCCI Entities belonging to an OCCI Kind or OCCI Mixin
9	TD/OCCI/CORE/READ/002	Retrieve the URLs of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin and related to an OCCI Category filter
10	TD/OCCI/CORE/READ/003	Retrieve the URLs of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin which contain a specific Attribute
11	TD/OCCI/CORE/READ/004	Retrieve the descriptions of all OCCI Entities belonging to an OCCI Kind or Mixin
12	TD/OCCI/CORE/READ/005	Retrieve the descriptions of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin and related to an OCCI Category filter
13	TD/OCCI/CORE/READ/006	Retrieve the description of an OCCI Entity
14	TD/OCCI/CORE/UPDATE/001	Full update of a specific OCCI Entity
15	TD/OCCI/CORE/UPDATE/002	Partial update of a specific OCCI Entity
16	TD/OCCI/CORE/UPDATE/003	Full update of a specific OCCI Mixin Collection
17	TD/OCCI/CORE/DELETE/001	Delete an OCCI Entity
18	TD/OCCI/CORE/DELETE/002	Delete all OCCI Entities belonging to an OCCI Kind
19	TD/OCCI/CORE/DELETE/003	Delete an OCCI Mixin
20	TD/OCCI/CORE/MISC/001	Trigger OCCI Action on existing OCCI Entity
21	TD/OCCI/CORE/MISC/002	Trigger OCCI Action on all OCCI Entities belonging to an OCCI Kind or OCCI Mixin
22	TD/OCCI/CORE/MISC/003	Associate OCCI Entities with OCCI Mixin
23	TD/OCCI/CORE/MISC/004	Disassociate OCCI Entities from OCCI Mixin

Table 5: OCCI Infrastructure Optional Tests

1	TD/OCCI/INFRA/CREATE/001	Create an OCCI Compute Resource
2	TD/OCCI/INFRA/CREATE/002	Create an OCCI Storage Resource
3	TD/OCCI/INFRA/CREATE/003	Create an OCCI Network Resource
4	TD/OCCI/ INFRA/CREATE/004	Create an OCCI Compute Resource using an OS and resource template
5	TD/OCCI/INFRA/CREATE/005	Create an OCCI Compute Resource with an OCCI Storagelink and an OCCI Networkinterface
6	TD/OCCI/INFRA/CREATE/006	Create an OCCI Storagelink between an existing OCCI Compute and OCCI Storage Resource
7	TD/OCCI/INFRA/CREATE/007	Create an OCCI Networkinterface between an existing OCCI Compute and OCCI Network Resource

4.5.2 CDMI Optional Tests

Table 6: CDMI Data Object Optional Tests

1	TD/CDMI/DATA/CREATE/001	Create a new CDMI Data Object
2	TD/CDMI/DATA/CREATE/002	Create a reference to an existing CDMI Data Object
3	TD/CDMI/DATA/CREATE/003	Copy an existing CDMI Data Object or CDMI Queue to a new OCCI Data Object
4	TD/CDMI/DATA/CREATE/004	Move a CDMI Data Object
5	TD/CDMI/DATA/CREATE/005	Create a new CDMI Data Object by deserializing an existing CDMI Data Object
6	TD/CDMI/DATA/CREATE/006	Create a new CDMI Data Object by serializing an existing CDMI object
7	TD/CDMI/DATA/READ/001	Read all fields from existing CDMI Data Object
8	TD/CDMI/DATA/READ/002	Read metadata from existing CDMI Data Object
9	TD/CDMI/DATA/READ/003	Read value from existing CDMI Data Object
10	TD/CDMI/DATA/READ/004	Read first 10 bytes from the value of an existing CDMI Data Object
11	TD/CDMI/DATA/UPDATE/001	Modify an existing CDMI Data Object
12	TD/CDMI/DATA/UPDATE/002	Modify the metadata of an existing CDMI Data Object
13	TD/CDMI/DATA/UPDATE/003	Modify the value of an existing CDMI Data Object
14	TD/CDMI/DATA/UPDATE/004	Modify the first 10 bytes of the value of an existing CDMI Data Object
15	TD/CDMI/DATA/DELETE/001	Delete an existing CDMI Data Object

Table 7: CDMI Container Optional Tests

1	TD/CDMI/CONTAINER/CREATE/001	Create a new CDMI Container
2	TD/CDMI/CONTAINER/CREATE/002	Create a reference to an existing CDMI Container
3	TD/CDMI/CONTAINER/CREATE/003	Copy a CDMI Container
4	TD/CDMI/CONTAINER/CREATE/004	Move an existing CDMI Container
5	TD/CDMI/CONTAINER/CREATE/005	Create a new CDMI Container by deserializing an existing CDMI Data Object
6	TD/CDMI/CONTAINER/READ/001	Read all fields from existing CDMI Container
7	TD/CDMI/CONTAINER/READ/002	Read metadata from existing CDMI Container
8	TD/CDMI/CONTAINER/READ/003	List children of an existing CDMI Container
9	TD/CDMI/CONTAINER/READ/004	List first 2 children of an existing CDMI Container
10	TD/CDMI/CONTAINER/UPDATE/001	Modify an existing CDMI Container
11	TD/CDMI/CONTAINER/UPDATE/002	Modify the metadata of an existing CDMI Container
12	TD/CDMI/CONTAINER/UPDATE/003	Create a snapshot of the contents of an existing CDMI Container
13	TD/CDMI/CONTAINER/UPDATE/004	Add an export protocol to an existing CDMI Container
14	TD/CDMI/CONTAINER/DELETE/001	Delete an existing CDMI Container

Table 8: CDMI Domain Optional Tests

1	TD/CDMI/DOMAIN/CREATE/001	Create a new CDMI Domain
2	TD/CDMI/DOMAIN/CREATE/002	Copy an existing CDMI Domain
3	TD/CDMI/DOMAIN/CREATE/003	Create a new CDMI Domain by deserializing an existing CDMI Data Object
4	TD/CDMI/DOMAIN/READ/001	Read all fields from existing CDMI Domain
5	TD/CDMI/DOMAIN/READ/002	Read metadata from existing CDMI Domain
6	TD/CDMI/DOMAIN/READ/003	List children of existing CDMI Domain
7	TD/CDMI/DOMAIN/UPDATE/001	Modify an existing CDMI Domain
8	TD/CDMI/DOMAIN/UPDATE/002	Modify the metadata of an existing CDMI Domain
9	TD/CDMI/DOMAIN/DELETE/001	Delete an existing CDMI Domain

Table 9: CDMI Queue Optional Tests

1	TD/CDMI/QUEUE/CREATE/001	Create a new CDMI Queue
2	TD/CDMI/QUEUE/CREATE/002	Create a reference to an existing CDMI Queue
3	TD/CDMI/QUEUE/CREATE/003	Copy an existing CDMI Queue
4	TD/CDMI/QUEUE/CREATE/004	Move an existing CDMI Queue
5	TD/CDMI/QUEUE/CREATE/005	Create a new CDMI Queue by deserializing an existing CDMI Data Object
6	TD/CDMI/QUEUE/READ/001	Read all fields from existing CDMI Queue
7	TD/CDMI/QUEUE/READ/002	Read metadata from existing CDMI Queue
8	TD/CDMI/QUEUE/READ/003	Read value of oldest enqueued object of existing CDMI Queue
9	TD/CDMI/QUEUE/READ/004	Read first 10 bytes of oldest enqueued object value of existing CDMI Queue
10	TD/CDMI/QUEUE/READ/005	Read queue values from existing CDMI Queue
11	TD/CDMI/QUEUE/UPDATE/001	Modify an existing CDMI Queue
12	TD/CDMI/QUEUE/UPDATE/002	Modify the metadata of an existing CDMI Queue
13	TD/CDMI/QUEUE/DELETE/001	Delete an existing CDMI Queue
14	TD/CDMI/QUEUE/ENQUEUE/001	Enqueue a data value to an existing CDMI Queue
15	TD/CDMI/QUEUE/ENQUEUE/002	Copy an existing CDMI Data Object or CDMI Queue to an existing CDMI Queue
16	TD/CDMI/QUEUE/ENQUEUE/003	Move an existing CDMI Data Object or CDMI Queue to an existing CDMI Queue
17	TD/CDMI/QUEUE/DEQUEUE/001	Dequeue oldest data value from an existing CDMI Queue
18	TD/CDMI/QUEUE/DEQUEUE/002	Dequeue the two oldest values from existing CDMI Queue

4.5.3 Interworking Optional Tests

Table 10: OCCI+CDMI Optional Tests

1	TD/INTER/OCCI+CDMI/CREATE/001	Create an OCCI Storagelink between an existing OCCI Compute Resource and existing CDMI Container
2	TD/INTER/OCCI+CDMI/CREATE/002	Create an OCCI Compute Resource with an OCCI Storagelink to an existing CDMI Container
3	TD/INTER/OCCI+CDMI/CREATE/003	Create a CDMI Container and connect it to an existing OCCI Compute Resource using an OCCI Storagelink
4	TD/INTER/OCCI+CDMI/READ/001	Retrieve the description of an OCCI Compute Resource with an OCCI Storagelink to a CDMI Container
5	TD/INTER/OCCI+CDMI/READ/002	Read OCCI export protocol field from existing CDMI Container
6	TD/INTER/OCCI+CDMI/UPDATE/001	Add permission for an existing OCCI Compute Resource to access an existing CDMI Container
7	TD/INTER/OCCI+CDMI/UPDATE/002	Remove permission for an existing OCCI Compute Resource to access an existing CDMI Container
8	TD/INTER/OCCI+CDMI/DELETE/001	Delete an OCCI Compute Resource with an OCCI Storagelink to a CDMI Container
9	TD/INTER/OCCI+CDMI/DELETE/002	Delete an existing CDMI Container with access permission for an OCCI Compute Resource
10	TD/INTER/OCCI+CDMI/DELETE/003	Delete the OCCI Storagelink between an OCCI Compute Resource and a CDMI Container

5 Test Configurations

This section defines roles and the different test configurations.

5.1 Roles

Equipment under test can take one of the following roles:

- OCCI Server
- OCCI Client

- CDMI Server
- CDMI Client

5.2 Test Configuration 1 (OCCI_CFG_01)

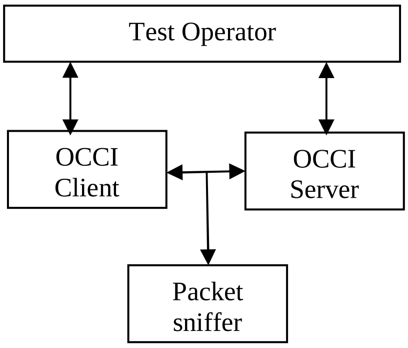

Figure 1: Basic Face 2 Face OCCI Configuration

5.3 Test Configuration 2 (CDMI_CFG_01)

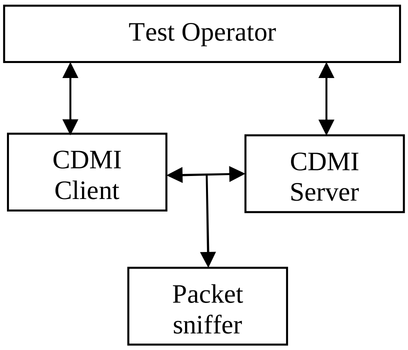

Figure 2: Basic Face 2 Face CDMI Configuration

5.4 Test Configuration 3 (OCCI_CDMI_CFG_01)

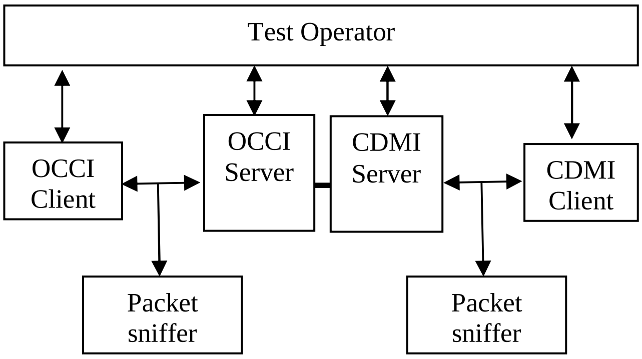

Figure 3: OCCI+CDMI Configuration

6 Feature List

In order to ease test setup and execution, participants are requested to fill in the following feature tables. Information in the tables will be used for selection/de-selection of tests related to optional features. It is highly recommended that **Bold** features are supported to enable a minimum set of interoperability among implementations.

6.1 OCCI Server

Table 11: OCCI Core features supported by OCCI Server

Feature	Support [Yes/No]	Dependent test descriptions
Create an OCCI Resource		**TD/OCCI/CORE/CREATE/002**
Create an OCCI Link		**TD/OCCI/CORE/CREATE/004**
Retrieval of OCCI Entity URLs		**TD/OCCI/CORE/READ/001** TD/OCCI/CORE/READ/002 TD/OCCI/CORE/READ/003
Deletion of OCCI Entities		**TD/OCCI/CORE/DELETE/001** **TD/OCCI/CORE/DELETE/002**
OCCI Category filter		TD/OCCI/CORE/DISCOVERY/002 TD/OCCI/CORE/READ/002 TD/OCCI/CORE/READ/005
OCCI Attribute filter		TD/OCCI/CORE/READ/003 TD/OCCI/CORE/READ/006
Create an OCCI Entity with an OCCI Mixin		TD/OCCI/CORE/CREATE/002
Create an OCCI Resource with an OCCI Link		TD/OCCI/CORE/CREATE/003
Create an OCCI Link with an OCCI Mixin		TD/OCCI/CORE/CREATE/005
Support for user-defined OCCI Mixins		TD/OCCI/CORE/CREATE/006 TD/OCCI/CORE/DELETE/003
Retrieval of multiple OCCI Entity descriptions		TD/OCCI/CORE/READ/004 TD/OCCI/CORE/READ/005 TD/OCCI/CORE/READ/006
Full update of OCCI Entity		TD/OCCI/CORE/UPDATE/001
Partial update of OCCI Entity		TD/OCCI/CORE/UPDATE/002
Managing OCCI Mixin Collections		TD/OCCI/CORE/UPDATE/003 TD/OCCI/CORE/MISC/003 TD/OCCI/CORE/MISC/004
Triggering Actions		TD/OCCI/CORE/MISC/001 TD/OCCI/CORE/MISC/002

Table 12: OCCI Infrastructure features supported by OCCI Server

Feature	Support [Yes/No]	Dependent test descriptions
Create an OCCI Compute Resource		**TD/OCCI/INFRA/CREATE/001**
Create an OCCI Storage Resource		TD/OCCI/INFRA/CREATE/002
Create an OCCI Network Resource		TD/OCCI/INFRA/CREATE/003
Create an OCCI Compute Resource using an OS and resource template		TD/OCCI/ INFRA/CREATE/004
Support for OCCI Storagelink		TD/OCCI/ INFRA/CREATE/005 TD/OCCI/INFRA/CREATE/006
Support for OCCI Networkinterface		TD/OCCI/ INFRA/CREATE/005 TD/OCCI/INFRA/CREATE/007
Support for OCCI Storagelink to CDMI Container		TD/INTER/OCCI+CDMI/CREATE/001 TD/INTER/OCCI+CDMI/CREATE/002 TD/INTER/OCCI+CDMI/CREATE/003 TD/INTER/OCCI+CDMI/READ/001 TD/INTER/OCCI+CDMI/DELETE/001 TD/INTER/OCCI+CDMI/DELETE/002 TD/INTER/OCCI+CDMI/DELETE/003

6.2 OCCI Client

Table 13: OCCI Core features supported by OCCI Client

Feature	Support [Yes/No]	Dependent test descriptions
Create an OCCI Resource		**TD/OCCI/CORE/CREATE/002**
Create an OCCI Link		**TD/OCCI/CORE/CREATE/004**
Retrieval of OCCI Entity URLs		**TD/OCCI/CORE/READ/001** TD/OCCI/CORE/READ/002 TD/OCCI/CORE/READ/003
Deletion of OCCI Entities		**TD/OCCI/CORE/DELETE/001** **TD/OCCI/CORE/DELETE/002**
OCCI Category filter		TD/OCCI/CORE/DISCOVERY/002 TD/OCCI/CORE/READ/002 TD/OCCI/CORE/READ/005
OCCI Attribute filter		TD/OCCI/CORE/READ/003 TD/OCCI/CORE/READ/006
Create an OCCI Entity with an OCCI Mixin		TD/OCCI/CORE/CREATE/002
Create an OCCI Resource with an OCCI Link		TD/OCCI/CORE/CREATE/003
Create an OCCI Link with an OCCI Mixin		TD/OCCI/CORE/CREATE/005
Support for user-defined OCCI Mixins		TD/OCCI/CORE/CREATE/006 TD/OCCI/CORE/DELETE/003
Retrieval of multiple OCCI Entity descriptions		TD/OCCI/CORE/READ/004 TD/OCCI/CORE/READ/005 TD/OCCI/CORE/READ/006
Full update of OCCI Entity		TD/OCCI/CORE/UPDATE/001
Partial update of OCCI Entity		TD/OCCI/CORE/UPDATE/002
Managing OCCI Mixin Collections		TD/OCCI/CORE/UPDATE/003 TD/OCCI/CORE/MISC/003 TD/OCCI/CORE/MISC/004
Triggering Actions		TD/OCCI/CORE/MISC/001 TD/OCCI/CORE/MISC/002

Table 14: OCCI Infrastructure features supported by OCCI Client

Feature	Support [Yes/No]	Dependent test descriptions
Create an OCCI Compute Resource		**TD/OCCI/INFRA/CREATE/001**
Create an OCCI Storage Resource		TD/OCCI/INFRA/CREATE/002
Create an OCCI Network Resource		TD/OCCI/INFRA/CREATE/003
Create an OCCI Compute Resource using an OS and resource template		TD/OCCI/ INFRA/CREATE/004
Support for OCCI Storagelink		TD/OCCI/ INFRA/CREATE/005 TD/OCCI/INFRA/CREATE/006
Support for OCCI Networkinterface		TD/OCCI/ INFRA/CREATE/005 TD/OCCI/INFRA/CREATE/007
Support for OCCI Storagelink to CDMI Container		TD/INTER/OCCI+CDMI/CREATE/001 TD/INTER/OCCI+CDMI/CREATE/002 TD/INTER/OCCI+CDMI/CREATE/003 TD/INTER/OCCI+CDMI/READ/001 TD/INTER/OCCI+CDMI/DELETE/001 TD/INTER/OCCI+CDMI/DELETE/002 TD/INTER/OCCI+CDMI/DELETE/003

6.3 CDMI Server

Table 15: CDMI Data Object features supported by CDMI Server

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Data Object		**TD/CDMI/DATA/CREATE/001**
Create a reference to a CDMI Data Object		TD/CDMI/DATA/CREATE/002
Copy a CDMI Data Object		TD/CDMI/DATA/CREATE/003
Move a CDMI Data Object		TD/CDMI/DATA/CREATE/004
Deserialize a CDMI Data Object		TD/CDMI/DATA/CREATE/005
Serialize a CDMI Data Object into a CDMI Data Object		TD/CDMI/DATA/CREATE/006
Serialize a CDMI Container		TD/CDMI/DATA/CREATE/006
Serialize a CDMI Domain		TD/CDMI/DATA/CREATE/006
Serialize a CDMI Queue		TD/CDMI/DATA/CREATE/006
Retrieve a CDMI Data Object		**TD/CDMI/DATA/READ/001**
Read the metadata of a CDMI Data Object		TD/CDMI/DATA/READ/002
Read the value of a CDMI Data Object		TD/CDMI/DATA/READ/003
Read specific byte range from the value of a CDMI Data Object		TD/CDMI/DATA/READ/004
Modify a CDMI Data Object		TD/CDMI/DATA/UPDATE/001
Modify the metadata of a CDMI Data Object		TD/CDMI/DATA/UPDATE/002
Modify the value of a CDMI Data Object		TD/CDMI/DATA/UPDATE/003
Modify specific byte range from the value of a CDMI Data Object		TD/CDMI/DATA/UPDATE/004
Delete a CDMI Data Object		**TD/CDMI/DATA/DELETE/001**

Table 16: CDMI Container features supported by CDMI Server

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Container		**TD/CDMI/CONTAINER/CREATE/001**
Create a reference to a CDMI Container		TD/CDMI/CONTAINER/CREATE/002
Copy a CDMI Container		TD/CDMI/CONTAINER/CREATE/003
Move a CDMI Container		TD/CDMI/CONTAINER/CREATE/004
Deserialize a CDMI Container		TD/CDMI/CONTAINER/CREATE/005
Retrieve a CDMI Container		TD/CDMI/CONTAINER/READ/001
Read the metadata of a CDMI Container		TD/CDMI/CONTAINER/READ/002
List children of a CDMI Container		TD/CDMI/CONTAINER/READ/003
List range of children of a CDMI Container		TD/CDMI/CONTAINER/READ/004
Modify a CDMI Container		TD/CDMI/CONTAINER/UPDATE/001
Modify the metadata of a CDMI Container		TD/CDMI/CONTAINER/UPDATE/002
Snapshot support for CDMI Container		TD/CDMI/CONTAINER/UPDATE/003
Support for NFS export		TD/CDMI/CONTAINER/UPDATE/004
Support for CIFS export		TD/CDMI/CONTAINER/UPDATE/004
Support for OCCI export		TD/CDMI/CONTAINER/UPDATE/004 TD/INTER/OCCI+CDMI/CREATE/001 TD/INTER/OCCI+CDMI/CREATE/002 TD/INTER/OCCI+CDMI/CREATE/003 TD/INTER/OCCI+CDMI/READ/002 TD/INTER/OCCI+CDMI/UPDATE/001 TD/INTER/OCCI+CDMI/UPDATE/002 TD/INTER/OCCI+CDMI/DELETE/001 TD/INTER/OCCI+CDMI/DELETE/002
Support for iSCSI export		TD/CDMI/CONTAINER/UPDATE/004
Support for WebDav export		TD/CDMI/CONTAINER/UPDATE/004
Delete a CDMI Container		**TD/CDMI/CONTAINER/DELETE/001**

Table 17: CDMI Domain features supported by CDMI Server

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Domain		TD/CDMI/DOMAIN/CREATE/001
Copy a CDMI Domain		TD/CDMI/DOMAIN/CREATE/002
Deserialize a CDMI Domain		TD/CDMI/DOMAIN/CREATE/003
Retrieve a CDMI Domain		TD/CDMI/DOMAIN/READ/001
Read the metadata of a CDMI Domain		TD/CDMI/DOMAIN/READ/002
List children of a CDMI Object		TD/CDMI/DOMAIN/READ/003
Modify a CDMI Domain		TD/CDMI/DOMAIN/UPDATE/001
Modify the metadata of a CDMI Domain		TD/CDMI/DOMAIN/UPDATE/002
Delete a CDMI Domain		TD/CDMI/DOMAIN/DELETE/001

Table 18: Features supported by CDMI Server

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Queue		TD/CDMI/QUEUE/CREATE/001
Create a reference to a CDMI Queue		TD/CDMI/QUEUE/CREATE/002
Copy a CDMI Queue		TD/CDMI/QUEUE/CREATE/003
Move a CDMI Queue		TD/CDMI/QUEUE/CREATE/004
Deserialize a CDMI Queue		TD/CDMI/QUEUE/CREATE/005
Retrieve a CDMI Queue		TD/CDMI/QUEUE/READ/001
Read the metadata of a CDMI Queue		TD/CDMI/QUEUE/READ/002
Read a value from a CDMI Queue		TD/CDMI/QUEUE/READ/003 TD/CDMI/QUEUE/READ/004 TD/CDMI/QUEUE/READ/005
Modify a CDMI Queue		TD/CDMI/QUEUE/UPDATE/002
Modify the metadata of a CDMI Queue		TD/CDMI/QUEUE/UPDATE/002
Modify a value in a CDMI Queue		TD/CDMI/QUEUE/ENQUEUE/001 TD/CDMI/QUEUE/ENQUEUE/002 TD/CDMI/QUEUE/ENQUEUE/003 TD/CDMI/QUEUE/DEQUEUE/001 TD/CDMI/QUEUE/DEQUEUE/002
Delete a CDMI Queue		TD/CDMI/QUEUE/DELETE/001

6.4 CDMI Client

Table 19: CDMI Data Object features supported by CDMI Client

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Data Object		**TD/CDMI/DATA/CREATE/001**
Create a reference to a CDMI Data Object		TD/CDMI/DATA/CREATE/002
Copy a CDMI Data Object		TD/CDMI/DATA/CREATE/003
Move a CDMI Data Object		TD/CDMI/DATA/CREATE/004
Deserialize a CDMI Data Object		TD/CDMI/DATA/CREATE/005
Serialize a CDMI Data Object into a CDMI Data Object		TD/CDMI/DATA/CREATE/006
Serialize a CDMI Container		TD/CDMI/DATA/CREATE/006
Serialize a CDMI Domain		TD/CDMI/DATA/CREATE/006
Serialize a CDMI Queue		TD/CDMI/DATA/CREATE/006
Retrieve a CDMI Data Object		**TD/CDMI/DATA/READ/001**
Read the metadata of a CDMI Data Object		TD/CDMI/DATA/READ/002
Read the value of a CDMI Data Object		TD/CDMI/DATA/READ/003
Read specific byte range from the value of a CDMI Data Object		TD/CDMI/DATA/READ/004
Modify a CDMI Data Object		TD/CDMI/DATA/UPDATE/001
Modify the metadata of a CDMI Data Object		TD/CDMI/DATA/UPDATE/002
Modify the value of a CDMI Data Object		TD/CDMI/DATA/UPDATE/003
Modify specific byte range from the value of a CDMI Data Object		TD/CDMI/DATA/UPDATE/004
Delete a CDMI Data Object		**TD/CDMI/DATA/DELETE/001**

Table 20: CDMI Container features supported by CDMI Client

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Container		**TD/CDMI/CONTAINER/CREATE/001**
Create a reference to a CDMI Container		TD/CDMI/CONTAINER/CREATE/002
Copy a CDMI Container		TD/CDMI/CONTAINER/CREATE/003
Move a CDMI Container		TD/CDMI/CONTAINER/CREATE/004
Deserialize a CDMI Container		TD/CDMI/CONTAINER/CREATE/005
Retrieve a CDMI Container		TD/CDMI/CONTAINER/READ/001
Read the metadata of a CDMI Container		TD/CDMI/CONTAINER/READ/002
List children of a CDMI Container		TD/CDMI/CONTAINER/READ/003
List range of children of a CDMI Container		TD/CDMI/CONTAINER/READ/004
Modify a CDMI Container		TD/CDMI/CONTAINER/UPDATE/001
Modify the metadata of a CDMI Container		TD/CDMI/CONTAINER/UPDATE/002
Snapshot a CDMI Container		TD/CDMI/CONTAINER/UPDATE/003
Delete a CDMI Container		**TD/CDMI/CONTAINER/DELETE/001**

Table 21: CDMI Domain features supported by CDMI Client

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Domain		TD/CDMI/DOMAIN/CREATE/001
Copy a CDMI Domain		TD/CDMI/DOMAIN/CREATE/002
Deserialize a CDMI Domain		TD/CDMI/DOMAIN/CREATE/003
Retrieve a CDMI Domain		TD/CDMI/DOMAIN/READ/001
Read the metadata of a CDMI Domain		TD/CDMI/DOMAIN/READ/002
List children of a CDMI Object		TD/CDMI/DOMAIN/READ/003
Modify a CDMI Domain		TD/CDMI/DOMAIN/UPDATE/001
Modify the metadata of a CDMI Domain		TD/CDMI/DOMAIN/UPDATE/002
Delete a CDMI Domain		TD/CDMI/DOMAIN/DELETE/001

Table 22: Features supported by CDMI Client

Feature	Support [Yes/No]	Dependent test descriptions
Create a CDMI Queue		TD/CDMI/QUEUE/CREATE/001
Create a reference to a CDMI Queue		TD/CDMI/QUEUE/CREATE/002
Copy a CDMI Queue		TD/CDMI/QUEUE/CREATE/003
Move a CDMI Queue		TD/CDMI/QUEUE/CREATE/004
Deserialize a CDMI Queue		TD/CDMI/QUEUE/CREATE/005
Retrieve a CDMI Queue		TD/CDMI/QUEUE/READ/001
Read the metadata of a CDMI Queue		TD/CDMI/QUEUE/READ/002
Read a value from a CDMI Queue		TD/CDMI/QUEUE/READ/003 TD/CDMI/QUEUE/READ/004 TD/CDMI/QUEUE/READ/005
Modify a CDMI Queue		TD/CDMI/QUEUE/UPDATE/002
Modify the metadata of a CDMI Queue		TD/CDMI/QUEUE/UPDATE/002
Modify a value in a CDMI Queue		TD/CDMI/QUEUE/ENQUEUE/001 TD/CDMI/QUEUE/ENQUEUE/002 TD/CDMI/QUEUE/ENQUEUE/003 TD/CDMI/QUEUE/DEQUEUE/001 TD/CDMI/QUEUE/DEQUEUE/002
Delete a CDMI Queue		TD/CDMI/QUEUE/DELETE/001

7 OCCI

This section provides the test descriptions for the different OCCI features.

7.1 OCCI Core

7.1.1 Discovery Interface

7.1.1.1 TD/OCCI/CORE/DISCOVERY/001

Interoperability Test Description			
Identifier:	TD/OCCI/CORE/DISCOVERY/001		
Objective:	Retrieving all OCCI Categories supported by the OCCI Server		
Configuration:	OCCI_CFG_01		
References:	GFD.185 [3], clause 3.4.1		
Pre-test conditions:			
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests all OCCI Categories supported by the OCCI Server**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is /-/ or /.well-known/org/ogf/occi/-/ • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • application/occi+json
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Body contains all OCCI Categories supported by the OCCI Server and at least the following categories • http://schemas.ogf.org/occi/core#entity • http://schemas.ogf.org/occi/core#resource • http://schemas.ogf.org/occi/core#link • The format of all OCCI Categories is compliant with the requested MIME type and the OCCI format restrictions
	4	**verify**	**OCCI Client displays the OCCI Categories received from the OCCI Server**

7.1.1.2 TD/OCCI/CORE/DISCOVERY/002

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/DISCOVERY/002**		
Objective:	Retrieving the OCCI Categories with an OCCI Category filter from the OCCI Server		
Configuration:	OCCI_CFG_01		
References:	GFD.185 [3], clause 3.4.1		
Pre-test conditions:	OCCI Client selects an OCCI Category provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests from the OCCI Server the OCCI Categories related to the OCCI Category retrieved in the pre-test conditions**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is /-/ or /.well-known/org/ogf/occi/-/ • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Content-Type header is one of the following MIME types • text/occi • text/plain • application/occi+json • The OCCI Category description is compliant with the requested MIME type and the OCCI format restrictions
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Body contains the OCCI Categories related to the OCCI Category filter • The format of all OCCI Categories is compliant with the requested MIME type and the OCCI format restrictions
	4	**verify**	**OCCI Client displays the OCCI Categories received from the OCCI Server**

7.1.2 Create

7.1.2.1 TD/OCCI/CORE/CREATE/001

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/CREATE/001**		
Objective:	Create an OCCI Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client selects an OCCI Kind describing an OCCI Resource as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Resource to be created • HTTP Content-Type header is one of the following MIME types • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Resource description • The OCCI Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Resource**
	5	**verify**	**OCCI Resource has been successfully created by OCCI Server**

7.1.2.2 TD/OCCI/CORE/CREATE/002

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/CREATE/002**		
Objective:	Create an OCCI Resource with an OCCI Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client selects an OCCI Kind describing an OCCI Resource as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client selects an OCCI Mixin as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind and OCCI Mixin to define an OCCI Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Resource to be created • HTTP Content-Type header is one of the following MIME types • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Resource description • The OCCI Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Resource**
	5	**verify**	**OCCI Resource has been successfully created by OCCI Server** **OCCI Resource contains OCCI Mixin**

7.1.2.3 TD/OCCI/CORE/CREATE/003

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/CREATE/003**		
Objective:	Create an OCCI Resource with an OCCI Link to an existing OCCI Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Resource to be linked e.g. by using the OCCI Resource URL as returned in TD/OCCI/CORE/CREATE/001 or TD/OCCI/CORE/READ/001 OCCI Client selects an OCCI Kind describing an OCCI Resource and an OCCI Kind describing an OCCI Link as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Resource with a link to the existing OCCI Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Resource description • The OCCI Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Resource**
	5	**verify**	**OCCI Resource has been successfully created by OCCI Server** **OCCI Resource has been successfully linked with existing OCCI Resource**

7.1.2.4 TD/OCCI/CORE/CREATE/004

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/CREATE/004**		
Objective:	Create an OCCI Link		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.5		
Pre-test conditions:	OCCI Client selects an OCCI Kind describing an OCCI Link as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Link Two existing OCCI Resources to be linked with each other		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Link as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Link to be created • HTTP Content-Type header is one of the following MIME types • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Link description with the two OCCI Resources to be linked as source and target • The OCCI Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Link
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Link**
	5	**verify**	**OCCI Link has been successfully created by OCCI Server**

7.1.2.5 TD/OCCI/CORE/CREATE/005

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/CREATE/005**		
Objective:	Create an OCCI Link with an OCCI Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.5		
Pre-test conditions:	OCCI Client selects an OCCI Kind describing an OCCI Link as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client selects an OCCI Mixin as provided by the discovery interface as described in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind and OCCI Mixin to define an OCCI Link Two existing OCCI Resources to be linked with each other		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Link as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Link to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Link description with the two OCCI Resources to be linked as source and target • The OCCI Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Link
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Link**
	5	**verify**	**OCCI Link has been successfully created by OCCI Server** **OCCI Link contains OCCI Mixin**

7.1.2.6 TD/OCCI/CORE/CREATE/006

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier:</b></td><td colspan="3"><b>TD/OCCI/CORE/CREATE/006</b></td></tr>
<tr><td><b>Objective:</b></td><td colspan="3">Add an OCCI Mixin definition</td></tr>
<tr><td><b>Configuration:</b></td><td colspan="3">OCCI_CFG_01</td></tr>
<tr><td><b>References:</b></td><td colspan="3">OCCI - GFD.185 [3], clause 3.4.1</td></tr>
<tr><td><b>Pre-test conditions:</b></td><td colspan="3"></td></tr>
<tr><td rowspan="6"><b>Test Sequence:</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td><b>1</b></td><td><b>stimulus</b></td><td><b>OCCI Client requests OCCI Server to add an OCCI Mixin definition</b></td></tr>
<tr><td>2</td><td>check</td><td>OCCI Client sends a HTTP POST request
• Request-URI is /-/ or /.well-known/org/ogf/occi/-/
• HTTP Content-Type header is one of the following MIME types:
  • text/occi
  • text/plain
  • application/occi+json
• HTTP Body contains the OCCI Mixin description
• The OCCI Mixin description is compliant with the requested MIME type and the OCCI format restrictions</td></tr>
<tr><td>3</td><td>check</td><td>OCCI Server sends a HTTP 200 (OK) response</td></tr>
<tr><td><b>4</b></td><td><b>verify</b></td><td><b>OCCI Client reports success of adding OCCI Mixin</b></td></tr>
<tr><td><b>5</b></td><td><b>verify</b></td><td><b>OCCI Mixin has been added to OCCI Category Collection on OCCI Server</b></td></tr>
</table>

7.1.3 Read

7.1.3.1 TD/OCCI/CORE/READ/001

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier:</b></td><td colspan="3"><b>TD/OCCI/CORE/READ/001</b></td></tr>
<tr><td><b>Objective:</b></td><td colspan="3">Retrieve the URLs of all OCCI Entities belonging to an OCCI Kind or OCCI Mixin</td></tr>
<tr><td><b>Configuration:</b></td><td colspan="3">OCCI_CFG_01</td></tr>
<tr><td><b>References:</b></td><td colspan="3">OCCI - GFD.185 [3], clause 3.4.4
OCCI - GFD.185 [3], clause 3.4.5</td></tr>
<tr><td><b>Pre-test conditions:</b></td><td colspan="3">OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001
OCCI Client extracts the location of the OCCI Kind or Mixin from the description</td></tr>
<tr><td rowspan="5"><b>Test Sequence:</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td><b>1</b></td><td><b>stimulus</b></td><td><b>OCCI Client requests OCCI Server to send the URLs of all OCCI Entities belonging to the OCCI Kind or OCCI Mixin</b></td></tr>
<tr><td>2</td><td>check</td><td>OCCI Client sends a HTTP GET request
• Request-URI is the location of the OCCI Kind or OCCI Mixin
• If HTTP Accept header is present it is containing at least one of the following MIME types:
  • text/plain
  • text/occi
  • text/uri-list</td></tr>
<tr><td>3</td><td>check</td><td>OCCI Server sends a HTTP 200 (OK) response
• HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6)
• HTTP header or HTTP body contains the rendering of the URLs of the OCCI Entity according to the MIME type specified in the HTTP Content-type header</td></tr>
<tr><td><b>4</b></td><td><b>verify</b></td><td><b>OCCI Client displays the URLs of the OCCI Entities</b></td></tr>
</table>

7.1.3.2 TD/OCCI/CORE/READ/002

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/READ/002**		
Objective:	Retrieve the URLs of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin and related to an OCCI Category filter		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.185 [3], clause 3.4.5		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location of the OCCI Kind or OCCI Mixin from the description OCCI Client selects an OCCI Category filter		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to send the URLs of the OCCI Entities belonging to the OCCI Kind or OCCI Mixin and related to the OCCI Category filter**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is the location of the OCCI Kind or OCCI Mixin • If HTTP Accept header is present it is containing at least one of the following MIME types: ◦ text/plain ◦ text/occi ◦ text/uri-list • HTTP Content-Type header is one of the following MIME types: ◦ text/occi ◦ text/plain ◦ application/occi+json • HTTP header or HTTP body contains the OCCI Category rendering according to the requested MIME type and the OCCI format restrictions
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP header or HTTP body contains the rendering of the URLs of the OCCI Entities according to the MIME type specified in the HTTP Content-type header
	4	**verify**	**OCCI Client only displays the URLs of the OCCI Entities which belong to the OCCI Kind or OCCI Mixin and are related to the OCCI Category specified as filter**

7.1.3.3 TD/OCCI/CORE/READ/003

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/READ/003**		
Objective:	Retrieve the URLs of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin which contain a specific Attribute		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.185 [3], clause 3.4.5		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location of the OCCI Kind or OCCI Mixin from the description OCCI Client selects an OCCI Category and extracts an Attribute from it as filter		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to send the URLs of the OCCI Entities belonging to the OCCI Kind or OCCI Mixin which contain a specific Attribute**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is the location of the OCCI Kind • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/plain • text/occi • text/uri-list • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • HTTP header or HTTP body contains the Attribute rendering according to the requested MIME type and the OCCI format restrictions
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP header or HTTP body contains the rendering of the URLs of the OCCI Entities according to the MIME type specified in the HTTP Content-type header
	4	**verify**	**OCCI Client only displays the URLs of the OCCI Entities which belong to the OCCI Kind or OCCI Mixin and contain the specified attribute**

7.1.3.4 TD/OCCI/CORE/READ/004

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/READ/004**		
Objective:	Retrieve the descriptions of all OCCI Entities belonging to an OCCI Kind or Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.3		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location of the OCCI Kind or OCCI Mixin from its description		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to send the descriptions of all OCCI Resources belonging to the OCCI Kind or Mixin**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is the location of the OCCI Kind or OCCI Mixin • HTTP Accept header is: • application/occi+json
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header is: • application/occi+json • HTTP Body contains the json rendering of the description of the OCCI Resources
	4	**verify**	**OCCI Client displays the descriptions of the OCCI Resources which belong to the OCCI Kind or OCCI Mixin**

7.1.3.5 TD/OCCI/CORE/READ/005

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier:</b></td><td colspan="3"><b>TD/OCCI/CORE/READ/005</b></td></tr>
<tr><td><b>Objective:</b></td><td colspan="3">Retrieve the descriptions of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin and related to an OCCI Category filter</td></tr>
<tr><td><b>Configuration:</b></td><td colspan="3">OCCI_CFG_01</td></tr>
<tr><td><b>References:</b></td><td colspan="3">OCCI - GFD.185 [3], clause 3.4.3</td></tr>
<tr><td><b>Pre-test conditions:</b></td><td colspan="3">OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001
OCCI Client extracts the location of the OCCI Kind or OCCI Mixin from the description
OCCI Client selects an OCCI Category filter</td></tr>
<tr><td rowspan="5"><b>Test Sequence:</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td><b>1</b></td><td><b>stimulus</b></td><td><b>OCCI Client requests OCCI Server to send the descriptions of the OCCI Entities belonging to the OCCI Kind or OCCI Mixin and related to the OCCI Category filter</b></td></tr>
<tr><td>2</td><td>check</td><td>OCCI Client sends a HTTP GET request
Request-URI is the location of the OCCI Kind or OCCI Mixin
HTTP Accept header is:
application/occi+json
HTTP Content-Type header is one of the following MIME types
text/occi
text/plain
application/occi+json
HTTP header or HTTP body contains the OCCI Category rendering according to the requested MIME type and the OCCI format restrictions</td></tr>
<tr><td>3</td><td>check</td><td>OCCI Server sends a HTTP 200 (OK) response
HTTP Content-Type header is
application/occi+json
HTTP Body contains the descriptions of the OCCI Entities according to the MIME type specified in the HTTP Content-type header</td></tr>
<tr><td><b>4</b></td><td><b>verify</b></td><td><b>OCCI Client only displays the descriptions of the OCCI Entities which belong to the OCCI Kind or OCCI Mixin and are related to the OCCI Category specified as filter</b></td></tr>
</table>

7.1.3.6 TD/OCCI/CORE/READ/006

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/READ/006**		
Objective:	Retrieve the descriptions of the OCCI Entities belonging to an OCCI Kind or OCCI Mixin and containing the Attribute filter		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.2		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location of the OCCI Kind or OCCI Mixin from the description OCCI Client selects an OCCI Category and extracts an Attribute from it as filter		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to send the descriptions of the OCCI Entities belonging to the OCCI Kind and containing the Attribute filter**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is the location of the OCCI Kind or OCCI Mixin • HTTP Accept header is: • application/occi+json • HTTP Content-Type header is one of the following MIME types • text/occi • text/plain • HTTP header or HTTP body contains the Attribute rendering according to the requested MIME type and the OCCI format restrictions
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header is: • application/occi+json • HTTP Body contains the descriptions of the OCCI Resources according to the MIME type specified in the HTTP Content-type header
	4	**verify**	**OCCI Client only displays the descriptions of the OCCI Entities which belong to the OCCI Kind or OCCI Mixin and contain the specified attribute**

7.1.3.7 TD/OCCI/CORE/READ/007

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/READ/007**		
Objective:	Retrieve the description of an OCCI Entity		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.185 [3], clause 3.4.5		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Entity as returned in TD/OCCI/CORE/READ/001		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to send the description of the OCCI Entity**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is the location of the OCCI Entity If HTTP Accept header is present it is containing at least one of the following MIME types: • text/plain • text/occi • application/occi+json
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP header or HTTP body message contains the rendering of the OCCI Entity according to the MIME type specified in the HTTP Content-type header
	4	**verify**	**OCCI Client displays the description of the OCCI Entity**

7.1.4 Update

7.1.4.1 TD/OCCI/CORE/UPDATE/001

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/UPDATE/001**		
Objective:	Full update of a specific OCCI Entity		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Entity as returned in TD/OCCI/CORE/READ/001		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to replace the description of the OCCI Entity**
	2	check	OCCI Client sends a HTTP PUT request • Request-URI is the location of the OCCI Entity • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Entity description • The OCCI Entity description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	If OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Header or HTTP Body contains the full, updated description of the OCCI Entity If OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the updated OCCI Entity
	4	**verify**	**OCCI Client displays success of update operation**
	5	**verify**	**OCCI Server has replaced the description of the OCCI Entity**

7.1.4.2 TD/OCCI/CORE/UPDATE/002

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/UPDATE/002**		
Objective:	Partial update of a specific OCCI Entity		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Entity as returned in TD/OCCI/CORE/READ/001		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to update the description of the OCCI Entity**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Entity • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the part of the OCCI Entity description which is to be updated • The OCCI Entity description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	If OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Header or HTTP Body contains the full, updated description of the OCCI Entity If OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the updated OCCI Entity
	4	**verify**	**OCCI Client displays success of update operation**
	5	**verify**	**OCCI Server has updated the description of the OCCI Entity**

7.1.4.3 TD/OCCI/CORE/UPDATE/003

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/UPDATE/003**		
Objective:	Full update of a specific OCCI Mixin Collection		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.3		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location from the OCCI Mixin		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests the OCCI Server to replace the OCCI Entities associated with the OCCI Mixin**
	2	check	OCCI Client sends a HTTP PUT request • Request-URI is the location of the OCCI Mixin • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • text/uri-list • HTTP Body contains a list of the URIs of the OCCI Entities which are to be associated with the OCCI Mixin • The list of URIs is compliant with the requested MIME type and the OCCI format restrictions
	3	check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays success of update operation**
	5	**verify**	**OCCI Server has disassociated all OCCI Entities from the OCCI Mixin** **OCCI Server has associated OCCI Entities from the request with the OCCI Mixin**

7.1.5 Delete

7.1.5.1 TD/OCCI/CORE/DELETE/001

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/DELETE/001**		
Objective:	Delete an OCCI Entity		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Entity as returned in TD/OCCI/CORE/READ/001		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to delete the OCCI Entity**
	2	check	OCCI Client sends a HTTP DELETE request • Request-URI is the location of the OCCI Entity
	3	check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays success message**
	5	**verify**	**OCCI Server has deleted OCCI Entity**

7.1.5.2 TD/OCCI/CORE/DELETE/002

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/DELETE/002**		
Objective:	Delete all OCCI Entities belonging to an OCCI Kind		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.2		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Kind as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location of the OCCI Kind from the description		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to delete all OCCI Entities belonging to the OCCI Kind**
	2	check	OCCI Client sends a HTTP DELETE request • Request-URI is the location of the OCCI Kind
	3	check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays success message**
	5	**verify**	**OCCI Server has deleted all OCCI Entities belonging to the OCCI Kind**

7.1.5.3 TD/OCCI/CORE/DELETE/003

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/DELETE/003**		
Objective:	Delete an OCCI Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.1		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to delete the OCCI Mixin**
	2	check	OCCI Client sends a HTTP DELETE request • Request-URI is /-/ • HTTP header or HTTP body contains the description of the OCCI Mixin
	3	check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays success message**
	5	**verify**	**OCCI Server removes all associations to the OCCI Mixin from OCCI Entity instances**

7.1.6 Miscellaneous

7.1.6.1 TD/OCCI/CORE/MISC/001

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/MISC/001**		
Objective:	Trigger OCCI Action on existing OCCI Entity		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Entity as returned in TD/OCCI/CORE/READ/001 OCCI Client retrieves the description of the OCCI Entity as returned in TD/OCCI/CORE/READ/007		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client selects an applicable OCCI Action from the OCCI Entity description and requests OCCI Server to trigger that OCCI Action**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the URI of the OCCI Resource with query string ?action=$ACTION where $ACTION corresponds to the term of the OCCI Action to be triggered • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Action description • The OCCI Action description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6)
	4	**verify**	**OCCI Client reports that action was triggered successfully**
	5	**verify**	**OCCI Action was triggered successfully by the OCCI Server on the OCCI Entity**

7.1.6.2 TD/OCCI/CORE/MISC/002

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/MISC/002**		
Objective:	Trigger OCCI Action on all OCCI Entities belonging to an OCCI Kind or OCCI Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.3		
Pre-test conditions:	OCCI Client retrieves the description of an OCCI Kind or OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location of the OCCI Kind or OCCI Mixin from its description		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	OCCI Client selects an OCCI Action from the OCCI Kind or OCCI Mixin description and requests the OCCI Server to trigger that OCCI Action on all OCCI Entities belonging to the OCCI Kind or OCCI Mixin
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind or OCCI Mixin with query string ?action=$ACTION where $ACTION corresponds to the term of the OCCI Action to be triggered • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Action description • The OCCI Action description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6)
	4	**verify**	**OCCI Client reports that action was triggered successfully**
	5	**verify**	**OCCI Action was triggered successfully by the OCCI Server on all OCCI Entities belonging to the OCCI Kind or OCCI Mixin**

7.1.6.3 TD/OCCI/CORE/MISC/003

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/MISC/003**		
Objective:	Associate OCCI Entities with OCCI Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.3		
Pre-test conditions:	OCCI Client retrieves URLs of OCCI Entities as returned in TD/OCCI/CORE/READ/001 OCCI Client retrieves the description of an OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location from the OCCI Mixin description		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to associate the OCCI Entities with the OCCI Mixin**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Mixin • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • text/uri-list • HTTP Body contains the URLs of the OCCI Entities
	3	check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays successful association**
	5	**verify**	**OCCI Server associated OCCI Entities with OCCI Mixin**

7.1.6.4 TD/OCCI/CORE/MISC/004

Interoperability Test Description			
Identifier:	**TD/OCCI/CORE/MISC/004**		
Objective:	Disassociate OCCI Entities from OCCI Mixin		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.3		
Pre-test conditions:	OCCI Client retrieves URLs of OCCI Entities as returned in TD/OCCI/CORE/READ/001 OCCI Client retrieves the description of an OCCI Mixin as returned in TD/OCCI/CORE/DISCOVERY/001 OCCI Client extracts the location from the OCCI Mixin description		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to disassociate the OCCI Entities from the OCCI Mixin**
	2	check	OCCI Client sends a HTTP DELETE request • Request-URI is the location of the OCCI Mixin • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • text/uri-list • HTTP Body contains the URLs of the OCCI Entities
	3	check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays successful disassociation**
	5	**verify**	**OCCI Server disassociated OCCI Entities from OCCI Mixin**

7.2 OCCI Infrastructure

7.2.1 Create

7.2.1.1 TD/OCCI/INFRA/CREATE/001

Interoperability Test Description			
Identifier:	**TD/OCCI/INFRA/CREATE/001**		
Objective:	Create an OCCI Compute Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.184 [2], clause 3.1		
Pre-test conditions:	OCCI Client selects the OCCI Kind of the OCCI Compute Resource as provided by the discovery interface TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Compute Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Compute Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Compute Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Compute Resource description • The OCCI Compute Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json

	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Compute Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Compute Resource**
	5	**verify**	**OCCI Compute Resource has been successfully created by OCCI Server**

7.2.1.2 TD/OCCI/INFRA/CREATE/002

Interoperability Test Description			
Identifier:	**TD/OCCI/INFRA/CREATE/002**		
Objective:	Create an OCCI Storage Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.184 [2], clause 3.3		
Pre-test conditions:	OCCI Client selects the OCCI Kind of the OCCI Storage Resource as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Storage Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Storage Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Storage Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Storage Resource description • The OCCI Storage Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Storage Resource**
	5	**verify**	**OCCI Storage Resource has been successfully created by OCCI Server**

7.2.1.3 TD/OCCI/INFRA/CREATE/003

Interoperability Test Description			
Identifier:	**TD/OCCI/INFRA/CREATE/003**		
Objective:	Create an OCCI Network Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.184 [2], clause 3.2		
Pre-test conditions:	OCCI Client selects the OCCI Kind of the OCCI Network Resource as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Network Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Network Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Network Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Header or Body contains the OCCI Network Resource description • The OCCI Network Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Network Resource**
	5	**verify**	**OCCI Network Resource has been successfully created by OCCI Server**

7.2.1.4 TD/OCCI/INFRA/CREATE/004

Interoperability Test Description			
Identifier:	**TD/OCCI/ INFRA/CREATE/004**		
Objective:	Create an OCCI Compute Resource using an OS and resource template		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.184 [2], clause 3.1 OCCI - GFD.184 [2], clause 3.5		
Pre-test conditions:	OCCI Client selects the OCCI Kind of the OCCI Compute Resource as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/001 OCCI Client selects an OCCI Mixin related to the OCCI OS Template Mixin and an OCCI Mixin related to the OCCI Resource Template Mixin as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/002 OCCI Client uses the information provided by the selected OCCI Kind and OCCI Mixins to define an OCCI Compute Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Compute Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Compute Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Header or Body contains the OCCI Compute Resource description • The OCCI Compute Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Compute Resource**
	5	**verify**	**OCCI Compute Resource has been successfully created by OCCI Server** **OCCI Compute Resource is associated with OCCI OS Template Mixin** **OCCI Compute Resource is associated with OCCI Resource Template Mixin**

7.2.1.5 TD/OCCI/INFRA/CREATE/005

Interoperability Test Description			
Identifier:	**TD/OCCI/INFRA/CREATE/005**		
Objective:	Create an OCCI Compute Resource with an OCCI Storagelink and an OCCI Networkinterface		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.4 OCCI - GFD.185 [3], clause 3.4.5 OCCI - GFD.184 [2], clause 3.1 OCCI - GFD.184 [2], clause 3.4.1 OCCI - GFD.184 [2], clause 3.4.2		
Pre-test conditions:	OCCI Client retrieves the URL of an OCCI Storage Resource as returned in TD/OCCI/CORE/READ/002 OCCI Client retrieves the URL of an OCCI Network Resource as returned in TD/OCCI/CORE/READ/002 OCCI Client selects the OCCI Kind of the OCCI Compute Resource as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Resource including an OCCI Storagelink with the URL of the OCCI Storage Resource as target and an OCCI Networkinterface with the URL of the OCCI Network Resource as target		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Compute Resource as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Compute Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Compute Resource description • The OCCI Compute Resource description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Compute Resource**
	5	**verify**	**OCCI Compute Resource has been successfully created by OCCI Server** **OCCI Compute Resource is linked with OCCI Storage Resource** **OCCI Compute Resource is linked with OCCI Network Resource**

7.2.1.6 TD/OCCI/INFRA/CREATE/006

Interoperability Test Description			
Identifier:	**TD/OCCI/INFRA/CREATE/006**		
Objective:	Create an OCCI Storagelink between an existing OCCI Compute and OCCI Storage Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.5 OCCI - GFD.184 [2], clause 3.4.2		
Pre-test conditions:	OCCI Client retrieves the URLs of an OCCI Compute Resource and an OCCI Storage Resource as returned in TD/OCCI/CORE/READ/002 OCCI Client selects the OCCI Kind of the OCCI Storagelink as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Storagelink between the OCCI Compute and OCCI Storage Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Storagelink as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Storagelink to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Storagelink description • The OCCI Storagelink description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Storagelink**
	5	**verify**	**OCCI Storagelink has been successfully created by OCCI Server** **OCCI Compute Resource is linked with OCCI Storage Resource**

7.2.1.7 TD/OCCI/INFRA/CREATE/007

Interoperability Test Description			
Identifier:	**TD/OCCI/INFRA/CREATE/007**		
Objective:	Create an OCCI Networkinterface between an existing OCCI Compute and OCCI Network Resource		
Configuration:	OCCI_CFG_01		
References:	OCCI - GFD.185 [3], clause 3.4.5 OCCI - GFD.184 [2], clause 3.4.2		
Pre-test conditions:	OCCI Client retrieves the URLs of an OCCI Compute Resource and an OCCI Network Resource as returned in TD/OCCI/CORE/READ/002 OCCI Client selects the OCCI Kind of the OCCI Networkinterface as provided by the discovery interface in TD/OCCI/CORE/DISCOVERY/001 OCCI Client uses the information provided by the selected OCCI Kind to define an OCCI Networkinterface between the OCCI Compute and OCCI Network Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create OCCI Networkinterface as defined in Pre-test conditions**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Networkinterface to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Networkinterface description • The OCCI Networkinterface description is compliant with the requested MIME type and the OCCI format restrictions • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports success of create operation and may display URL of created OCCI Networkinterface**
	5	**verify**	**OCCI Networkinterface has been successfully created by OCCI Server** **OCCI Compute is linked with OCCI Network**

8 CDMI

This section provides the test descriptions for the different CDMI features.

8.1 Capabilities

8.1.1 Read

8.1.1.1 TD/CDMI/CAPABILITIES/READ/001

Interoperability Test Description			
Identifier:	**TD/CDMI/CAPABILITIES/READ/001**		
Objective:	Retrieve root CDMI Capability Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 12.1.1 CDMI - ISO/IEC 17826 [4], clause 12.2		
Pre-test conditions:			
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests the root CDMI Capability Object from the CDMI Server**
	2	check	CDMI Client sends a HTTP GET request Request URI is <root URI>/cdmi_capabilities/ according to clause 12.2.1 HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) If HTTP Accept header is present it is containing the MIME type `application/cdmi-capability`
	3	check	CDMI Server sends a HTTP 200 (OK) HTTP Content-Type header is `application/cdmi-capability` HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) HTTP Body consists of a JSON object containing the fields defined in clause 12.2.6 Field capabilities of JSON object in HTTP Body contains entries according to 12.1.1 Field children of JSON object in HTTP Body contains entries "domain/", "container/", "dataobject/", and "queue/"
	4	**verify**	**CDMI Client displays all fields of the root CDMI Capability Object**

8.1.1.2 TD/CDMI/CAPABILITIES/READ/002

Interoperability Test Description			
Identifier:	**TD/CDMI/CAPABILITIES/READ/002**		
Objective:	List children of the root CDMI Capability Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 12.2		
Pre-test conditions:			
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests children of the root CDMI Capability Object from the CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/cdmi_capabilities/?children according to clause 12.2.1 • If the client uses pagination the request URI contains a specific range e.g. for the first two children <root URI>/cdmi_capabilities/?children?0:1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-capability
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-capability • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the children field • Field children of JSON object in HTTP Body contains the first two children of the root CDMI Capability Object
	4	**verify**	**CDMI Client displays children of root CDMI Capability Object**

8.1.1.3 TD/CDMI/CAPABILITIES/READ/003

Interoperability Test Description			
Identifier:	**TD/CDMI/CAPABILITIES/READ/003**		
Objective:	Read capabilities field from existing CDMI Capability Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 12.1.1 CDMI - ISO/IEC 17826 [4], clause 12.2		
Pre-test conditions:			
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests capabilities field of the root CDMI Capability Object from the CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/cdmi_capabilities/?capabilities according to clause 12.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-capability
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-capability • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the capabilities field • Field capabilities of JSON object in HTTP Body contains entries according to clause 12.1.1
	4	**verify**	**CDMI Client displays capabilities of CDMI Server**

8.1.1.4 TD/CDMI/CAPABILITIES/READ/004

Interoperability Test Description	
Identifier:	**TD/CDMI/CAPABILITIES/READ/004**
Objective:	Retrieve the Capabilities of a CDMI object
Configuration:	CDMI_CFG_01
References:	CDMI - ISO/IEC 17826 [4], clause 8.4.1 CDMI - ISO/IEC 17826 [4], clause 9.4.1 CDMI - ISO/IEC 17826 [4], clause 10.3.1 CDMI - ISO/IEC 17826 [4], clause 11.3.1 CDMI - ISO/IEC 17826 [4], clause 12.1 CDMI - ISO/IEC 17826 [4], clause 12.2
Pre-test conditions:	Existing CDMI Data Object, CDMI Container, CDMI Domain or CDMI Queue

Test Sequence:	Step	Type	Description
	1	**stimulus**	**CDMI Client requests the capabilities URI of an existing CDMI Data Object, CDMI Container, CDMI Domain or CDMI Queue**
	2	check	CDMI Client sends a HTTP GET request • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-capability For CDMI Data Object • Request URI is <root URI>/<ContainerName>/<DataObjectName>?capabilitiesURI according to clause 8.4.1 For CDMI Container Object • Request URI is <root URI>/<ContainerName>/<TheContainerName>/?capabilitiesURI according to clause 9.4.1 For CDMI Domain Object • Request URI is <root URI>/ cdmi_domains/<DomainName>/<TheDomainName>/?capabilitiesURI according to clause 10.3.1 For CDMI Queue Object • Request URI is <root URI>/<ContainerName>/<QueueName>?capabilitiesURI according to clause 11.3.1
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-capability • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the capabilitiesURI field
	4	**stimulus**	**CDMI Client uses the capabilities URI to retrieve the CDMI Capability Object for the CDMI object**
	5	check	CDMI Client sends a HTTP GET request • Request URI is the capabilities URI retrieved in step 1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-capability
	6	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-capability • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the capabilities field • Field capabilities of JSON object in HTTP Body contains entries according to clause 12.1.2 and 12.1.3 • For CDMI Data Object o Field capabilities of JSON object in HTTP Body may additionally contain entries according to clause 12.1.4 • For CDMI Containers o Field capabilities of JSON object in HTTP Body may additionally contain entries according to clause 12.1.5 • For CDMI Domains o Field capabilities of JSON object in HTTP Body may additionally contain entries according to clause 12.1.6 • For CDMI Queues o Field capabilities of JSON object in HTTP Body may additionally contain entries according to clause 12.1.7
	6	**verify**	**CDMI Client displays capabilities of CDMI Object**

8.2 Data Objects

8.2.1 Create

8.2.1.1 TD/CDMI/DATA/CREATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/CREATE/001**		
Objective:	Create a new CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.2 CDMI - ISO/IEC 17826 [4], clause 8.3 TD/CDMI/CAPABILITIES/READ/004		
Pre-test conditions:	Existing CDMI Container with capability cdmi_create_dataobject		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a new CDMI Data Object with value "just some test data, can be removed" and metadata "key1":"value1" and "key2":"value2"**

	2	check	CDMI Client sends a HTTP PUT request
			If HTTP Header includes X-CDMI-Specification-Version
			• CDMI Client creates CDMI Data Object using the CDMI Content Type
			• Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.2.1
			• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
			• If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message
			• HTTP Content-Type header is application/cdmi-object
			• If HTTP Accept header is present it is containing the MIME type application/cdmi-object
			If HTTP Header does not include X-CDMI-Specification-Version
			• CDMI Client creates CDMI Data Object using non-CDMI Content Type
			• HTTP Content-Type header includes the MIME type of the data object to be created. It may include the charset of the data object (e.g. ;charset=utf-8 or ;charset=base64) as specified in RFC 2046 [i.1]
			• If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message
			HTTP Body consists of a JSON object containing the fields defined in clause 8.2.5
			• The value field of the JSON object in the HTTP Body contains the contents of the CDMI Data Object to be created e.g.
			{
			  "mimetype" : "text/plain",
			  "metadata" : {
			    "key1": "value1",
			    "key2":"value2"
			  },
			  "value": "just some test data, can be removed"
			}

	3	check	If Request HTTP Header included X-CDMI-Specification-Version CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.7 • If HTTP status code is 201 o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202 o Field completionStatus of JSON object in HTTP Body is Processing • Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of create operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of create operation**
	5	**verify**	**CDMI Server has successfully created data object**

8.2.1.2 TD/CDMI/DATA/CREATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/CREATE/002**		
Objective:	Create a reference to an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_create_reference Existing CDMI Data Object		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a reference to an existing CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message • HTTP Content-Type header is application/cdmi-object • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • HTTP Body consists of a JSON object containing the fields defined in clause 8.2.5. o Field reference contains the URI of an existing CDMI Data Object
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-object • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 8.2.7 • If HTTP status code is 201 o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202 o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** **• CDMI Client reports success of create reference operation** **If CDMI Server sends HTTP 202 status code** **• CDMI Client reports delayed completion of create reference operation**
	5	**verify**	**CDMI Server has successfully created a reference to an existing CDMI Data Object**

8.2.1.3 TD/CDMI/DATA/CREATE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/CREATE/003**		
Objective:	Copy an existing CDMI Data Object or CDMI Queue to a new OCCI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_copy_dataobject Existing CDMI Data Object or CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to copy an existing CDMI Data Object or CDMI Queue into a new CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message • HTTP Content-Type header is application/cdmi-object • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • HTTP Body consists of a JSON object containing the fields defined in clause 8.2.5 . The copy field contains the URI of a CDMI Data Object or queue
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-object • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 8.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of copy operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of copy operation**
	5	**verify**	**CDMI Server has successfully copied CDMI Data Object or queue**

8.2.1.4 TD/CDMI/DATA/CREATE/004

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/CREATE/004**		
Objective:	Move a CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_move_dataobject Existing CDMI Object		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to move a CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message • HTTP Content-Type header is application/cdmi-object • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • HTTP Body consists of a JSON object containing the fields defined in clause 8.2.5 . The move field contains the URI of a CDMI Data Object
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-object • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 8.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of move operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of move operation**
	5	**verify**	**CDMI Server has successfully moved CDMI Data Object**

8.2.1.5 TD/CDMI/DATA/CREATE/005

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td>**Identifier:**</td><td colspan="3">**TD/CDMI/DATA/CREATE/005**</td></tr>
<tr><td>**Objective:**</td><td colspan="3">Create a new CDMI Data Object by deserializing an existing CDMI Data Object</td></tr>
<tr><td>**Configuration:**</td><td colspan="3">CDMI_CFG_01</td></tr>
<tr><td>**References:**</td><td colspan="3">CDMI - ISO/IEC 17826 [4], clause 8.2</td></tr>
<tr><td>**Pre-test conditions:**</td><td colspan="3">Existing CDMI Container with capability cdmi_deserialize_dataobject
Existing CDMI Data Object containing the serialization of a CDMI Data Object</td></tr>
<tr><td>**Test Sequence:**</td><td>**Step**</td><td>**Type**</td><td>**Description**</td></tr>
<tr><td></td><td>**1**</td><td>**stimulus**</td><td>**CDMI Client requests CDMI Server to deserialize a serialized CDMI Data Object**</td></tr>
<tr><td></td><td>2</td><td>check</td><td>CDMI Client sends a HTTP PUT request
• Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.2.1
• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
• If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message
• HTTP Content-Type header is application/cdmi-object
• If HTTP Accept header is present it is containing the MIME type application/cdmi-object
• HTTP Body consists of a JSON object containing the fields defined in clause 8.2.5 . The deserialize field contains the URI of the serialized CDMI Data Object</td></tr>
<tr><td></td><td>3</td><td>check</td><td>CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED)
• HTTP Content-Type header is application/cdmi-object
• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0)
• HTTP Body consists of a JSON object containing the fields defined in clause 8.2.7
• If HTTP status code is 201:
o Field completionStatus of JSON object in HTTP Body is Complete
• If HTTP status code is 202:
o Field completionStatus of JSON object in HTTP Body is Processing
o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100</td></tr>
<tr><td></td><td>**4**</td><td>**verify**</td><td>**If CDMI Server sends HTTP 201 status code**
• **CDMI Client reports success of deserialize operation**
If CDMI Server sends HTTP 202 status code
• **CDMI Client reports delayed completion of deserialize operation**</td></tr>
<tr><td></td><td>**5**</td><td>**verify**</td><td>**CDMI Server has successfully deserialized CDMI Data Object**</td></tr>
</table>

8.2.1.6 TD/CDMI/DATA/CREATE/006

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier:</b></td><td colspan="3"><b>TD/CDMI/DATA/CREATE/006</b></td></tr>
<tr><td><b>Objective:</b></td><td colspan="3">Create a new CDMI Data Object by serializing an existing CDMI object</td></tr>
<tr><td><b>Configuration:</b></td><td colspan="3">CDMI_CFG_01</td></tr>
<tr><td><b>References:</b></td><td colspan="3">CDMI - ISO/IEC 17826 [4], clause 8.2</td></tr>
<tr><td><b>Pre-test conditions:</b></td><td colspan="3">Existing CDMI Container with capability cdmi_serialize_dataobject, cdmi_serialize_container, cdmi_serialize_domain, or cdmi_serialize_queue
Existing CDMI Data Object, CDMI Container, CDMI Domain or CDMI Queue</td></tr>
<tr><td rowspan="6"><b>Test Sequence:</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td><b>1</b></td><td><b>stimulus</b></td><td><b>CDMI Client requests CDMI Server to serialize the existing CDMI Data Object, CDMI Container, CDMI Domain or CDMI Queue into a CDMI Data Object</b></td></tr>
<tr><td>2</td><td>check</td><td>CDMI Client sends a HTTP PUT request
• Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.2.1
• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
• If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message
• HTTP Content-Type header is application/cdmi-object
• If HTTP Accept header is present it is containing the MIME type application/cdmi-object
• HTTP Body consists of a JSON object containing the fields defined in clause 8.2.5 . The serialize field contains the URI of the CDMI Data Object, CDMI Container, CDMI Domain or CDMI Queue</td></tr>
<tr><td>3</td><td>check</td><td>CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED)
• HTTP Content-Type header is application/cdmi-object
• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0)
• HTTP Body consists of a JSON object containing the fields defined in clause 8.2.7
• If HTTP status code is 201:
o Field completionStatus of JSON object in HTTP Body is Complete
• If HTTP status code is 202:
o Field completionStatus of JSON object in HTTP Body is Processing
o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100</td></tr>
<tr><td><b>4</b></td><td><b>verify</b></td><td><b>If CDMI Server sends HTTP 201 status code</b>
<b>• CDMI Client reports success of serialize operation</b>
<b>If CDMI Server sends HTTP 202 status code</b>
<b>• CDMI Client reports delayed completion of serialize operation</b></td></tr>
<tr><td><b>5</b></td><td><b>verify</b></td><td><b>CDMI Server has successfully serialized CDMI Data Object, CDMI Container, CDMI Domain or CDMI Queue</b></td></tr>
</table>

8.2.2 Read

8.2.2.1 TD/CDMI/DATA/READ/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/READ/001**		
Objective:	Read all fields from existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.4		
Pre-test conditions:	Existing CDMI Data Object		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to describe CDMI Data Object**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-object
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-object • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 8.4.6 • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**CDMI Client displays all fields of the CDMI Data Object**

8.2.2.2 TD/CDMI/DATA/READ/002

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/READ/002**		
Objective:	Read metadata from existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.4 CDMI - ISO/IEC 17826 [4], clause 16		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_read_metadata		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests metadata of the CDMI Data Object from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<DataObjectName>?metadata according to clause 8.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-object
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-object • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the metadata field • Field metadata of JSON object in HTTP Body contains entries according to clause 16
	4	**verify**	**CDMI Client displays metadata of the CDMI Data Object**

8.2.2.3 TD/CDMI/DATA/READ/003

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/READ/003**		
Objective:	Read value from existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.4 CDMI - ISO/IEC 17826 [4], clause 8.5		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_read_value		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests value of the CDMI Data Object from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • If HTTP header includes X-CDMI-Specification-Version: o CDMI Client requests the CDMI Data Object with CDMI Content Type o Request URI is <root URI>/<ContainerName>/<DataObjectName>?value according to clause 8.4.1 o HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) o If HTTP Accept header is present it is containing the MIME type application/cdmi-object • If HTTP header does not include X-CDMI-Specification-Version: o CDMI Client requests the CDMI Data Object with non-CDMI Content Type
	3	check	CDMI Server sends a HTTP 200 (OK) • If Request HTTP Header contains X-CDMI-Specification-Version: o HTTP Content-Type header is application/cdmi-object o HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) o HTTP Body consists of a JSON object containing only the value field according to clause 8.4.6 • If Request HTTP Header does not contain X-CDMI-Specification-Version: o HTTP Header Content-Type corresponds to the mimetype field in the data object according to clause 8.5.5 o HTTP Header Location is the URI that the reference redirects to if the CDMI Data Object is a reference according to clause 8.5.5 o HTTP Body contains the value of the CDMI Data Object according to clause 8.5.6
	4	**verify**	**CDMI Client displays value of the CDMI Data Object**

8.2.2.4 TD/CDMI/DATA/READ/004

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/READ/004**		
Objective:	Read first 10 bytes from the value of an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.4 CDMI - ISO/IEC 17826 [4], clause 8.5 RFC 2616 [i.2], clause 14.35.1		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_read_value_range		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests first 10 bytes from the value of an existing CDMI Data Object from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • If HTTP header includes X-CDMI-Specification-Version: o Request URI is <root URI>/<ContainerName>/<DataObjectName>?value:0-9 according to clause 8.4.1 o HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) o If HTTP Accept header is present it is containing the MIME type application/cdmi-object • If HTTP header does not include X-CDMI-Specification-Version: o CDMI Client requests the CDMI Data Object using a non-CDMI content type o HTTP header Range is 0-9 according to clause 14.35.1 of RFC 2616 [i.2]
	3	check	CDMI Server sends a HTTP 200 (OK) • If Request HTTP Header contains X-CDMI-Specification-Version: o HTTP Content-Type header is application/cdmi-object o HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) o HTTP Body consists of a JSON object containing only the value field according to clause 8.4.6 o Field value of JSON object in HTTP Body contains the first 10 bytes from the value of the CDMI Data Object • If Request HTTP Header does not contain X-CDMI-Specification-Version: o HTTP Header Content-Type is the mimetype field in the data object according to clause 8.5.5 o HTTP Header Location is the URI that the reference redirects to if the CDMI Data Object is a reference according to clause 8.5.5 • HTTP Body contains the first 10 bytes from the value of the CDMI Data Object according to clause 8.5.6
	4	**verify**	**CDMI Client displays first 10 bytes from the value of the CDMI Data Object**

8.2.3 Update

8.2.3.1 TD/CDMI/DATA/UPDATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/UPDATE/001**		
Objective:	Modify an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.6		
Pre-test conditions:	Existing CDMI Data Object		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • If HTTP X-CDMI-Partial is present it is set to true and the create request continues in another HTTP message • HTTP Content-Type header is application/cdmi-object • HTTP Body consists of a JSON object containing the fields defined in clause 8.6.4
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.2.3.2 TD/CDMI/DATA/UPDATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/UPDATE/002**		
Objective:	Modify the metadata of an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.6		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_modify_metadata		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update metadata of an existing CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<DataObjectName>?metadata according to clause 8.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • HTTP Content-Type header is application/cdmi-object • HTTP Body consists of a JSON object containing only the metadata field as defined in clause 8.6.4
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.2.3.3 TD/CDMI/DATA/UPDATE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/UPDATE/003**		
Objective:	Modify the value of an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.6 CDMI - ISO/IEC 17826 [4], clause 8.7		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_modify_value		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update value of an existing CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request If HTTP Header includes X-CDMI-Specification-Version • CDMI Client updates CDMI Data Object using CDMI Content Type • Request URI is <root URI>/<ContainerName>/<DataObjectName>?value according to clause 8.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP X-CDMI-Partial is present it is set to true and the update request continues in another HTTP message • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • HTTP Content-Type header is application/cdmi-object • HTTP Body consists of a JSON object containing only the value field as defined in clause 8.6.4 If HTTP Header does not include X-CDMI-Specification-Version: • CDMI Client updates CDMI Data Object using non-CDMI Content Type • HTTP Content-Type header includes the MIME type of the data object to be updated. It may include the charset of the data object (e.g. ;charset=utf-8 or ;charset=base64) as specified in RFC 2046 [i.1] • If HTTP X-CDMI-Partial is present it is set to true and the update request continues in another HTTP message • HTTP Body contains the contents of the CDMI Data Object to be created
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.2.3.4 TD/CDMI/DATA/UPDATE/004

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/UPDATE/004**		
Objective:	Modify the first 10 bytes of the value of an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.6 CDMI - ISO/IEC 17826 [4], clause 8.7		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_modify_value_range		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update first 10 bytes of the value of an existing CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request If HTTP Header includes X-CDMI-Specification-Version: • CDMI Client updates CDMI Data Object using CDMI Content Type • Request URI is <root URI>/<ContainerName>/<DataObjectName>?value:0-9 according to clause 8.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP X-CDMI-Partial is present it is set to true and the update request continues in another HTTP message • If HTTP Accept header is present it is containing the MIME type application/cdmi-object • HTTP Content-Type header is application/cdmi-object • HTTP Body consists of a JSON object as defined in clause 8.6.4 and contains value for the first 10 bytes of the CDMI Data Object which is to be updated If HTTP Header does not include X-CDMI-Specification-Version: • CDMI Client updates CDMI Data Object using non-CDMI Content Type • HTTP Content-Type header includes the MIME type of the data object to be updated. It may include the charset of the data object (e.g. ;charset=utf-8 or ;charset=base64) as specified in RFC 2046 [i.1] • If HTTP X-CDMI-Partial is present it is set to true and the update request continues in another HTTP message • HTTP Body contains the a value for the first 10 bytes of the CDMI Data Object which is to be updated
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.2.4 Delete

8.2.4.1 TD/CDMI/DATA/DELETE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DATA/DELETE/001**		
Objective:	Delete an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 8.8 CDMI - ISO/IEC 17826 [4], clause 8.9		
Pre-test conditions:	Existing CDMI Data Object with capability cdmi_delete_dataobject		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to delete the CDMI Data Object**
	2	check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/<ContainerName>/<DataObjectName> according to clause 8.8.1
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of delete operation**

8.3 Container Objects

8.3.1 Create

8.3.1.1 TD/CDMI/CONTAINER/CREATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/CREATE/001**		
Objective:	Create a new CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.2 CDMI - ISO/IEC 17826 [4], clause 9.3		
Pre-test conditions:	Capability cdmi_create_container is present in the root CDMI Container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a new CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • If HTTP Header includes X-CDMI-Specification-Version: o CDMI Client creates CDMI Container using the CDMI Content Type o Request URI is <root URI>/<ContainerName>/<NewContainerName> according to clause 9.2.1 o HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) o HTTP Content-Type header is application/cdmi-container o If HTTP Accept header is present it is containing the MIME type application/cdmi-container o HTTP Body consists of a JSON object containing the fields defined in clause 9.2.5
	3	check	If Request HTTP Header includes X-CDMI-Specification-Version CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of create operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of create operation**
	5	**verify**	**CDMI Server has successfully created CDMI Container**

8.3.1.2 TD/CDMI/CONTAINER/CREATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/CREATE/002**		
Objective:	Create a reference to an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_create_reference		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a reference to an existing CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<NewContainerName> according to clause 9.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-container • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.5 . The reference field contains the URI of a CDMI Container
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of create reference operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of create reference operation**
	5	**verify**	**CDMI Server has successfully created reference to CDMI Container**

8.3.1.3 TD/CDMI/CONTAINER/CREATE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/CREATE/003**		
Objective:	Copy a CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_copy_container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to copy an existing CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<NewContainerName> according to clause 9.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-container • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.5 . The copy field contains the URI of an existing CDMI Container
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of copy operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of copy operation**
	5	**verify**	**CDMI Server has successfully copied CDMI Container**

8.3.1.4 TD/CDMI/CONTAINER/CREATE/004

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/CREATE/004**		
Objective:	Move an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_move_container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to move an existing CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<NewContainerName> according to clause 9.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-container • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.5 . The move field contains the URI of an existing CDMI Container
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.7 • If HTTP status code is 201 o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202 o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of move operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of move operation**
	5	**verify**	**CDMI Server has successfully moved CDMI Container**

8.3.1.5 TD/CDMI/CONTAINER/CREATE/005

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/CREATE/005**		
Objective:	Create a new CDMI Container by deserializing an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_deserialize_container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to deserialize an existing CDMI Data Object into a new CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/<NewContainerName> according to clause 9.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-container • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.5 . The deserialize field contains the URI of an existing CDMI Data Object
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of deserialize operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of deserialize operation**
	5	**verify**	**CDMI Server has successfully deserialized CDMI Data Object into CDMI Container**

8.3.2 Read

8.3.2.1 TD/CDMI/CONTAINER/READ/001

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/READ/001**		
Objective:	Read all fields from existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.4		
Pre-test conditions:	Existing CDMI Container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to describe CDMI Container**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<TheContainerName> according to clause 9.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 9.4.6 • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**CDMI Client displays all fields of the CDMI Container**

8.3.2.2 TD/CDMI/CONTAINER/READ/002

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/READ/002**		
Objective:	Read metadata from existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.4		
Pre-test conditions:	Existing CDMI Container with capability cdmi_read_metadata		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests metadata of the CDMI Container from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<TheContainerName>?metadata according to clause 9.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the metadata field as defined in clause 9.4.6 • Field metadata of JSON object in HTTP Body contains entries according to clause 16
	4	**verify**	**CDMI Client displays metadata of the CDMI Container**

8.3.2.3 TD/CDMI/CONTAINER/READ/003

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/READ/003**		
Objective:	List children of an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.4		
Pre-test conditions:	Existing CDMI Container with capability cdmi_list_children		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests list of children of the CDMI Container from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<TheContainerName>?children according to clause 9.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the children field as defined in clause 9.4.6 • Field children of JSON object in HTTP Body contains a JSON Array with the names of all children of the CDMI Container
	4	**verify**	**CDMI Client displays list of children of the CDMI Container**

8.3.2.4 TD/CDMI/CONTAINER/READ/004

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/READ/004**		
Objective:	List first 2 children of an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.4		
Pre-test conditions:	Existing CDMI Container with capability cdmi_list_children_range		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests first two children of the CDMI Container from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<TheContainerName>?children:0-2 according to clause 9.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-container • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the children field as defined in clause 9.4.6 • Field children of JSON object in HTTP Body contains a JSON Array with the first two children of the CDMI Container
	4	**verify**	**CDMI Client displays first two children of the CDMI Container**

8.3.3 Update

8.3.3.1 TD/CDMI/CONTAINER/UPDATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/UPDATE/001**		
Objective:	Modify an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.5		
Pre-test conditions:	Existing CDMI Container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Content-Type header is application/cdmi-container • HTTP Body consists of a JSON object containing the fields defined in clause 9.5.5
	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.3.3.2 TD/CDMI/CONTAINER/UPDATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/UPDATE/002**		
Objective:	Modify the metadata of an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.5		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_metadata		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update metadata of CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< TheContainerName >?metadata according to clause 9.5.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Content-Type header is application/cdmi-container • HTTP Body consists of a JSON object containing only the metadata field as defined in clause 9.5.5
	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.3.3.3 TD/CDMI/CONTAINER/UPDATE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/UPDATE/003**		
Objective:	Create a snapshot of the contents of an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.5		
Pre-test conditions:	Existing CDMI Container with capability cdmi_snapshot		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to snapshot the contents of the CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Content-Type header is application/cdmi-container • HTTP Body consists of a JSON object contains the snapshot field with the name of the snapshot as defined in clause 9.5.5
	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of snapshot operation**

8.3.3.4 TD/CDMI/CONTAINER/UPDATE/004

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/UPDATE/004**		
Objective:	Add an export protocol to an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.5 CDMI - ISO/IEC 17826 [4], clause 13		
Pre-test conditions:	Existing CDMI Container with capability cdmi_export_nfs, cdmi_export_cifs, cdmi_export_occi, cdmi_export_iscsi or cdmi_export_webdav		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to add NFS, CIFS, OCCI, iSCSI or WebDAV as export protocol to the CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Content-Type header is application/cdmi-container • HTTP Body consists of a JSON object which contains the exports field with information on each enabled export protocol as defined in clause 13
	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of adding export protocol**

8.3.4 Delete

8.3.4.1 TD/CDMI/CONTAINER/DELETE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/CONTAINER/DELETE/001**		
Objective:	Delete an existing CDMI Container		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 9.6 CDMI - ISO/IEC 17826 [4], clause 9.7		
Pre-test conditions:	Existing CDMI Container with capability cdmi_delete_container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to delete the CDMI Container**
	2	check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.6.1
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of delete operation**

8.4 Domain Objects

8.4.1 Create

8.4.1.1 TD/CDMI/DOMAIN/CREATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/CREATE/001**		
Objective:	Create a new CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.2		
Pre-test conditions:	Existing CDMI Domain with capability cdmi_create_domain		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a new CDMI Domain**
	2	check	CDMI Client sends a HTTP PUT request If HTTP Header includes X-CDMI-Specification-Version • CDMI Client creates CDMI Container using CDMI Content Type • Request URI is <root URI/cdmi_domains/<DomainName>/<NewDomainName>/ according to clause 10.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-domain • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain • HTTP Body consists of a JSON object containing the fields defined in clause 10.2.4
	3	check	CDMI Server sends HTTP 201 (CREATED) • HTTP Content-Type header is application/cdmi-domain • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 10.2.6
	4	**verify**	**CDMI Client reports success of create operation**
	5	**verify**	**CDMI Server has successfully created CDMI Domain**

8.4.1.2 TD/CDMI/DOMAIN/CREATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/CREATE/002**		
Objective:	Copy an existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.2		
Pre-test conditions:	Existing CDMI Domain with capability cdmi_copy_domain		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to copy a CDMI Domain**
	2	check	CDMI Client sends a HTTP PUT request If HTTP Header includes X-CDMI-Specification-Version • CDMI Client creates CDMI Container using CDMI Content Type • Request URI is <root URI/cdmi_domains/<DomainName>/<NewDomainName>/ according to clause 10.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-domain • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain • HTTP Body consists of a JSON object containing the copy field with the URI of the CDMI Domain to be copied as defined in clause 10.2.4
	3	check	CDMI Server sends HTTP 201 (CREATED) • HTTP Content-Type header is application/cdmi-domain • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 10.2.6
	4	**verify**	**CDMI Client reports success of copy operation**
	5	**verify**	**CDMI Server has successfully copied CDMI Domain**

8.4.1.3 TD/CDMI/DOMAIN/CREATE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/CREATE/003**		
Objective:	Create a new CDMI Domain by deserializing an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.2		
Pre-test conditions:	Existing CDMI Data Object containing the serialization of a CDMI Domain Existing CDMI Domain with capability cdmi_deserialize_domain		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to deserialize a CDMI Data Object into a new CDMI Domain**
	2	check	CDMI Client sends a HTTP PUT request If HTTP Header includes X-CDMI-Specification-Version • CDMI Client creates CDMI Container using CDMI Content Type • Request URI is <root URI/cdmi_domains/<DomainName>/<NewDomainName>/ according to clause 10.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-domain • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain • HTTP Body consists of a JSON object containing the deserialize field with the URI of the CDMI Data Object to be deserialized as defined in clause 10.2.4
	3	check	CDMI Server sends HTTP 201 (CREATED) • HTTP Content-Type header is application/cdmi-domain • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 10.2.6
	4	**verify**	**CDMI Client reports success of deserialize operation**
	5	**verify**	**CDMI Server has successfully deserialized CDMI Data Object into CDMI Domain**

8.4.2 Read

8.4.2.1 TD/CDMI/DOMAIN/READ/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/READ/001**		
Objective:	Read all fields from existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.3		
Pre-test conditions:	Existing CDMI Domain		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to describe CDMI Domain**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/cdmi_domains/<DomainName>/<TheDomainName>/ according to clause 10.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-domain • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 10.3.6
	4	**verify**	**CDMI Client displays all fields of the CDMI Domain**

8.4.2.2 TD/CDMI/DOMAIN/READ/002

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/READ/002**		
Objective:	Read metadata from existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.3		
Pre-test conditions:	Existing CDMI Domain with capability cdmi_read_metadata		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests metadata of CDMI Domain from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/ cdmi_domains/<DomainName>/<TheDomainName>/?metadata according to clause 10.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-domain • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the field metadata as defined in clause 10.3.6 with entries as defined in clause 16
	4	**verify**	**CDMI Client displays metadata of the CDMI Domain**

8.4.2.3 TD/CDMI/DOMAIN/READ/003

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/READ/003**		
Objective:	List children of existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.3		
Pre-test conditions:	Existing CDMI Domain with capability cdmi_list_children		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests childrens of CDMI Domain from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/ cdmi_domains/<DomainName>/<TheDomainName>/?children according to clause 10.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain
	3	check	CDMI Server sends a HTTP 200 (OK) • HTTP Content-Type header is application/cdmi-domain • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the field children as defined in clause 10.3.6
	4	**verify**	**CDMI Client displays children of the CDMI Domain**

8.4.3 Update

8.4.3.1 TD/CDMI/DOMAIN/UPDATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/UPDATE/001**		
Objective:	Modify an existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.4		
Pre-test conditions:	Existing CDMI Domain		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update CDMI Domain**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI/cdmi_domains/<DomainName>/<NewDomainName>/ according to clause 10.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain • HTTP Content-Type header is application/cdmi-domain • HTTP Body consists of a JSON object containing the fields defined in clause 10.4.4
	3	check	CDMI Server sends HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.4.3.2 TD/CDMI/DOMAIN/UPDATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/UPDATE/002**		
Objective:	Modify the metadata of an existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.4		
Pre-test conditions:	Existing CDMI Domain with capability cdmi_modify_metadata		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update CDMI Domain**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI/ cdmi_domains/<DomainName>/<NewDomainName>/?metadata according to clause 10.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-domain • HTTP Content-Type header is application/cdmi-domain • HTTP Body consists of a JSON object containing only the metadata field as defined in clause 10.4.4 and the content defined in clause 16
	3	check	CDMI Server sends HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.4.4 Delete

8.4.4.1 TD/CDMI/DOMAIN/DELETE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/DOMAIN/DELETE/001**		
Objective:	Delete an existing CDMI Domain		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 10.5		
Pre-test conditions:	Existing CDMI Domain with capability cdmi_delete_domain		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to delete the CDMI Domain**
	2	check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/cdmi_domains/<DomainName>/<TheDomainName>/ according to clause 10.5.1 • HTTP Header includes X-CDMI-Specification-Version
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of delete operation**

8.5 Queue Objects

8.5.1 Create

8.5.1.1 TD/CDMI/QUEUE/CREATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/CREATE/001**		
Objective:	Create a new CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_create_queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a new CDMI Queue**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of create operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of create operation**
	5	**verify**	**CDMI Server has successfully created CDMI Queue**

8.5.1.2 TD/CDMI/QUEUE/CREATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/CREATE/002**		
Objective:	Create a reference to an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_create_reference Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a reference to a CDMI Queue**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5. The reference field contains the URI of a CDMI Queue
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of create operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of create operation**
	5	**verify**	**CDMI Server has successfully created reference to CDMI Queue**

8.5.1.3 TD/CDMI/QUEUE/CREATE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/CREATE/003**		
Objective:	Copy an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_copy_queue Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to copy an existing CDMI Queue**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5. The copy field contains the URI of a CDMI Queue
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of copy operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of copy operation**
	5	**verify**	**CDMI Server has successfully copied CDMI Queue**

8.5.1.4 TD/CDMI/QUEUE/CREATE/004

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/CREATE/004**		
Objective:	Move an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_move_queue Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to move an existing CDMI Queue**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5. The move field contains the URI of a CDMI Queue
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** • **CDMI Client reports success of move operation** **If CDMI Server sends HTTP 202 status code** • **CDMI Client reports delayed completion of move operation**
	5	**verify**	**CDMI Server has successfully moved CDMI Queue**

8.5.1.5 TD/CDMI/QUEUE/CREATE/005

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/CREATE/005**		
Objective:	Create a new CDMI Queue by deserializing an existing CDMI Data Object		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.2		
Pre-test conditions:	Existing CDMI Container with capability cdmi_deserialize_queue Existing CDMI Data Object containing serialization of CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to deserialize a CDMI Queue from a CDMI Data Object**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.2.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5. The deserialize field contains the URI of a CDMI Data Object
	3	check	CDMI Server sends a HTTP 201 (CREATED) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.7 • If HTTP status code is 201: o Field completionStatus of JSON object in HTTP Body is Complete • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**If CDMI Server sends HTTP 201 status code** **• CDMI Client reports success of deserialize operation** **If CDMI Server sends HTTP 202 status code** **• CDMI Client reports delayed completion of deserialize operation**
	5	**verify**	**CDMI Server has successfully deserialized CDMI Queue**

8.5.2 Read

8.5.2.1 TD/CDMI/QUEUE/READ/001

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/READ/001**		
Objective:	Read all fields from existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.3		
Pre-test conditions:	Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to describe CDMI Queue**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing the fields defined in clause 11.3.6 • Field value of JSON object in HTTP Body contains value of the oldest item in the queue, unless the queueValues range is empty • If HTTP status code is 202: o Field completionStatus of JSON object in HTTP Body is Processing o Field percentComplete of JSON object in HTTP Body is present and indicates the percentage of completion as a numeric integer value from 0 through 100
	4	**verify**	**CDMI Client displays all fields of the CDMI Queue**

8.5.2.2 TD/CDMI/QUEUE/READ/002

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/READ/002**		
Objective:	Read metadata from existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.3		
Pre-test conditions:	Existing CDMI Container with capability cdmi_read_metadata Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests capabilities of CDMI Container from CDMI Server according to TD/CDMI/CAPABILITIES/READ/004**
	2	**verify**	**Capability cdmi_read_metadata is present in CDMI Container**
	3	**stimulus**	**CDMI Client requests metadata of the CDMI Queue from CDMI Server**
	4	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/< QueueName>?metadata according to clause 11.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue
	5	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the metadata field as defined in clause 11.3.6 • Field metadata of JSON object in HTTP Body contains entries according to clause 16
	6	**verify**	**CDMI Client displays metadata of the CDMI Queue**

8.5.2.3 TD/CDMI/QUEUE/READ/003

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/READ/003**		
Objective:	Read value of oldest enqueued object of existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.3		
Pre-test conditions:	Existing CDMI Container with capability cdmi_read_value Existing CDMI Queue with at least one enqueued value		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests value of oldest enqueued object of the CDMI Queue from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request Request URI is <root URI>/<ContainerName>/< QueueName>?value according to clause 11.3.1 HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) If HTTP Accept header is present it is containing the MIME type `application/cdmi-queue`
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) HTTP Content-Type header is `application/cdmi-queue` HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) HTTP Body consists of a JSON object containing only the value field as defined in clause 11.3.6 Field value of JSON object in HTTP Body contains only the value of the oldest enqueued object
	4	**verify**	**CDMI Client displays value of oldest enqueued object of the CDMI Queu**

8.5.2.4 TD/CDMI/QUEUE/READ/004

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/READ/004**		
Objective:	Read first 10 bytes of oldest enqueued object value of existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.3		
Pre-test conditions:	Existing CDMI Container with capability cdmi_read_value Existing CDMI Queue with at least one enqueued value		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests first 10 bytes of oldest enqueued object value of the CDMI Queue from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/< QueueName>?value:0-9 according to clause 11.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the value field as defined in clause 11.3.6 • Field value of JSON object in HTTP Body contains only the first 10 bytes of the oldest enqueued object value
	4	**verify**	**CDMI Client displays first 10 bytes of oldest enqueued object value of the CDMI Queue**

8.5.2.5 TD/CDMI/QUEUE/READ/005

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/READ/005**		
Objective:	Read queue values from existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.3		
Pre-test conditions:	Existing CDMI Queue with at least one enqueued value and capability cdmi_read_value		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests value of oldest enqueued object of the CDMI Queue from CDMI Server**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/< QueueName>?values according to clause 11.3.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue
	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED) • HTTP Content-Type header is application/cdmi-queue • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0) • HTTP Body consists of a JSON object containing only the value field as defined in clause 11.3.6 • Field value of JSON object in HTTP Body contains all values of the enqueued objects
	4	**verify**	**CDMI Client displays all values enqueued objects of the CDMI Queue**

8.5.3 Update

8.5.3.1 TD/CDMI/QUEUE/UPDATE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/UPDATE/001**		
Objective:	Modify an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.4		
Pre-test conditions:	Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update CDMI Queue**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Content-Type header is application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.4.4
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT) • HTTP Body is empty
	4	**verify**	**CDMI Client displays success of update operation**

8.5.3.2 TD/CDMI/QUEUE/UPDATE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/UPDATE/002**		
Objective:	Modify the metadata of an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.4		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_metadata Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to update CDMI Queue**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< QueueName>?metadata according to clause 11.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Content-Type header is application/cdmi-queue • HTTP Body consists of a JSON object containing only the metadata field as defined in clause 11.4.4
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of update operation**

8.5.4 Delete

8.5.4.1 TD/CDMI/QUEUE/DELETE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/DELETE/001**		
Objective:	Delete an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.5		
Pre-test conditions:	Existing CDMI Container with capability cdmi_delete_queue Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to delete the CDMI Queue**
	2	check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/<ContainerName>/< QueueName > according to clause 11.5.1
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of delete operation**

8.5.5 Enqueue

8.5.5.1 TD/CDMI/QUEUE/ENQUEUE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/ENQUEUE/001**		
Objective:	Enqueue a data value to an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.6		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_value Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to enqueue a data value to an existing CDMI Queue**
	2	check	CDMI Client sends a HTTP POST request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5 • Field value of JSON object in HTTP Body contains data value to be enqueued
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client reports success of enqueue operation**
	5	**verify**	**CDMI Server has successfully enqueued data object to CDMI Queue**

8.5.5.2 TD/CDMI/QUEUE/ENQUEUE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/ENQUEUE/002**		
Objective:	Copy an existing CDMI Data Object or CDMI Queue to an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.6		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_value Existing CDMI Queue Existing CDMI Data Object or CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to copy an existing CDMI Data Object or the values of an existing CDMI Queue to a CDMI Queue**
	2	check	CDMI Client sends a HTTP POST request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5 • Field copy of JSON object in HTTP Body contains URI of a CDMI Data Object or CDMI Queue
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client reports success of enqueue operation**
	5	**verify**	**CDMI Server has successfully enqueued CDMI Data Object or values of CDMI Queue to a CDMI Queue**

8.5.5.3 TD/CDMI/QUEUE/ENQUEUE/003

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/ENQUEUE/003**		
Objective:	Move an existing CDMI Data Object or CDMI Queue to an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.6		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_value Existing CDMI Queue Existing CDMI Data Object or CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to move an existing CDMI Data Object or the values of an existing CDMI Queue to a CDMI Queue**
	2	check	CDMI Client sends a HTTP POST request • Request URI is <root URI>/<ContainerName>/< QueueName> according to clause 11.6.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue • HTTP Body consists of a JSON object containing the fields defined in clause 11.2.5 • Field move of JSON object in HTTP Body contains URI of a CDMI Data Object or CDMI Queue
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client reports success of enqueue operation**

8.5.6 Dequeue

8.5.6.1 TD/CDMI/QUEUE/DEQUEUE/001

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/DEQUEUE/001**		
Objective:	Dequeue oldest data value from an existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.7		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_value Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to dequeue oldest data value from existing CDMI Queue**
	2	check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/<ContainerName>/< QueueName>?value according to clause 11.7.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client reports success of dequeue operation**
	5	**verify**	**CDMI Server has successfully dequeued oldest data value from CDMI Queue**

8.5.6.2 TD/CDMI/QUEUE/DEQUEUE/002

Interoperability Test Description			
Identifier:	**TD/CDMI/QUEUE/DEQUEUE/002**		
Objective:	Dequeue the two oldest values from existing CDMI Queue		
Configuration:	CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 11.7		
Pre-test conditions:	Existing CDMI Container with capability cdmi_modify_value Existing CDMI Queue		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to dequeue the two oldest data values from existing CDMI Queue**
	2	check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/<ContainerName>/< QueueName>?values:2 according to clause 11.7.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • HTTP Content-Type header is application/cdmi-queue • If HTTP Accept header is present it is containing the MIME type application/cdmi-queue
	3	check	CDMI Server sends a HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client reports success of dequeue operation**
	5	**verify**	**CDMI Server has successfully dequeued the two oldest data values from CDMI Queue**

9 Interworking

This section provides the test descriptions for the features addressed jointly with several Cloud specifications.

9.1 OCCI and CDMI

This section provides the test descriptions for the different features addressed jointly by OCCI and CDMI specifications.

9.1.1 Create

9.1.1.1 TD/INTER/OCCI+CDMI/CREATE/001

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/CREATE/001**		
Objective:	Create an OCCI Storagelink between an existing OCCI Compute Resource and existing CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/OCCI/INFRA/CREATE/006		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing OCCI Compute Resource Existing CDMI Container with permission for OCCI Compute Resource to access it		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create an OCCI Storagelink between the OCCI Compute Resource and the CDMI Container**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Storagelink to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Storagelink description • The OCCI Storagelink description is compliant with the requested MIME type and the OCCI format restrictions. The target of the OCCI Storagelink is the URI of the CDMI Container • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports that OCCI Compute Resource has been successfully linked to CDMI Container**
	5	**verify**	**OCCI Compute Resource can access the CDMI Container**

9.1.1.2 TD/INTER/OCCI+CDMI/CREATE/002

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/CREATE/002**		
Objective:	Create an OCCI Compute Resource with an OCCI Storagelink to an existing CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/OCCI/INFRA/CREATE/005		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing CDMI Container with permission for OCCI Compute Resource to access it		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to create an OCCI Compute Resource with an OCCI Storagelink to the CDMI Container**
	2	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Compute Resource to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Compute Resource description including information on the OCCI Storagelink . The target of the OCCI Storagelink is the URI of the CDMI Container • The description is compliant with the requested MIME type and the OCCI format restrictions. The target of the OCCI Storagelink is the URI of the CDMI Container • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	3	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	4	**verify**	**OCCI Client reports that OCCI Compute Resource has been successfully created**
	5	**stimulus**	**CDMI Client adds permission for the OCCI Compute Resource to access the CDMI Container**

	6	check	CDMI Client sends a HTTP PUT request
			• Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1
			• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
			• If HTTP Accept header is present it is containing the MIME type application/cdmi-container
			• HTTP Content-Type header is application/cdmi-container
			• HTTP Body consists of a JSON object which contains the exports field with information on each enabled export protocol as defined in clause 13. It contains the following information on the OCCI Export Protocol using the iSCSI or NFSv4 as $PROTOCOL
			"OCCI/$PROTOCOL": {
			  "identifier": "$CDMI_CONTAINER_OBJECT_ID",
			  "permissions": [
			    "$OCCI_COMPUTE_URL"
			  ]
			}

	7	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	8	**verify**	**CDMI Client displays success of adding access permission to CDMI Container**
	9	**verify**	**OCCI Compute Resource can access the CDMI Container**

9.1.1.3 TD/INTER/OCCI+CDMI/CREATE/003

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/CREATE/003**		
Objective:	Create a CDMI Container and connect it to an existing OCCI Compute Resource using an OCCI Storagelink		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/CDMI/CONTAINER/CREATE/001		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing OCCI Compute Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to create a new CDMI Container**

	2	check	CDMI Client sends a HTTP PUT request
			• Request URI is <root URI>/<ContainerName>/<NewContainerName> according to clause 9.2.1
			• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
			• HTTP Content-Type header is application/cdmi-container
			• If HTTP Accept header is present it is containing the MIME type application/cdmi-container
			• HTTP Body consists of a JSON object containing the fields defined in clause 9.2.5
			• The exports field contains information on each enabled export protocol as defined in clause 13. It contains the following information on the OCCI Export Protocol using the iSCSI or NFSv4 as $PROTOCOL
			"OCCI/$PROTOCOL": {
			"identifier": "$CDMI_CONTAINER_OBJECT_ID",
			"permissions": [
			"$OCCI_COMPUTE_URL"
			]
			}

	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of creating CDMI Container and granting access permission to OCCI Compute Resource**
	5	**stimulus**	**OCCI Client requests OCCI Server to create an OCCI Storagelink between the OCCI Compute Resource and the CDMI Container**
	6	check	OCCI Client sends a HTTP POST request • Request-URI is the location of the OCCI Kind corresponding to the OCCI Storagelink to be created • HTTP Content-Type header is one of the following MIME types: • text/occi • text/plain • application/occi+json • HTTP Body contains the OCCI Storagelink description • The OCCI Storagelink description is compliant with the requested MIME type and the OCCI format restrictions. The target of the OCCI Storagelink is the URI of the CDMI Container • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/occi • text/plain • text/uri-list • application/occi+json
	7	check	OCCI Server sends a HTTP 201 (CREATED) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP Location header contains URL of the created OCCI Resource
	8	**verify**	**OCCI Client reports that OCCI Compute Resource has been successfully linked to CDMI Container**
	9	**verify**	**OCCI Compute Resource can access the CDMI Container**

9.1.2 Read

9.1.2.1 TD/INTER/OCCI+CDMI/READ/001

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/READ/001**		
Objective:	Retrieve the description of an OCCI Compute Resource with an OCCI Storagelink to a CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	OCCI - GFD.184 [2], clause 3.4.3 TD/OCCI/CORE/READ/007		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing OCCI Compute Resource with an OCCI Storagelink to a CDMI Container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to send the description of the OCCI Compute Resource**
	2	check	OCCI Client sends a HTTP GET request • Request-URI is the location of the OCCI Resource • If HTTP Accept header is present it is containing at least one of the following MIME types: • text/plain • text/occi • application/occi+json
	3	check	OCCI Server sends a HTTP 200 (OK) response • HTTP Content-Type header corresponds to request's HTTP Accept header if present (see GDF.185 [3], clause 3.6.6) • HTTP body message contains the rendering of the OCCI Resource according to the MIME type specified in the HTTP Content-type header
	4	**verify**	**OCCI Client displays the description of the OCCI Compute Resource which includes information on the OCCI Storagelink targeting a CDMI Container**

9.1.2.2 TD/INTER/OCCI+CDMI/READ/002

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/READ/002**		
Objective:	Read OCCI export protocol field from existing CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 TD/CDMI/CONTAINER/READ/001		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing CDMI Container with OCCI export protocol		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client requests CDMI Server to describe CDMI Container**
	2	check	CDMI Client sends a HTTP GET request • Request URI is <root URI>/<ContainerName>/<TheContainerName>?exports according to clause 9.4.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container

	3	check	CDMI Server sends a HTTP 200 (OK) or HTTP 202 (ACCEPTED)
			• HTTP Content-Type header is application/cdmi-container
			• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Server (e.g. 1.0.2, 1.5, 2.0)
			• HTTP Body consists of a JSON object which contains the exports field with the following information on the OCCI Export Protocol using the iSCSI or NFSv4 as $PROTOCOL:
			"OCCI/$PROTOCOL": {
			"identifier": "$CDMI_CONTAINER_OBJECT_ID",
			"permissions": [
			"$OCCI_COMPUTE_URL"
			]
			}

	4	**verify**	**CDMI Client displays all fields of the CDMI Container and contains information on the OCCI export protocol**

9.1.3 Update

9.1.3.1 TD/INTER/OCCI+CDMI/UPDATE/001

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/UPDATE/001**		
Objective:	Add permission for an existing OCCI Compute Resource to access an existing CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/CDMI/CONTAINER/UPDATE/004		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage An existing CDMI Container An existing OCCI Compute Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client adds permission for the OCCI Compute Resource to access the CDMI Container**

	2	check	CDMI Client sends a HTTP PUT request
			• Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1
			• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
			• If HTTP Accept header is present it is containing the MIME type application/cdmi-container
			• HTTP Content-Type header is application/cdmi-container
			• HTTP Body consists of a JSON object which contains the exports field with information on each enabled export protocol as defined in clause 13. It contains the following information on the OCCI Export Protocol using the iSCSI or NFSv4 as $PROTOCOL
			"OCCI/$PROTOCOL": {
			"identifier": "$CDMI_CONTAINER_OBJECT_ID",
			"permissions": [
			"$OCCI_COMPUTE_URL"
			]
			}

	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of adding access permission to CDMI Container**

9.1.3.2 TD/INTER/OCCI+CDMI/UPDATE/002

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/UPDATE/002**		
Objective:	Remove permission for an existing OCCI Compute Resource to access an existing CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/CDMI/CONTAINER/UPDATE/004		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage An existing OCCI Compute Resource with an OCCI Storagelink to a CDMI Container An existing CDMI Container with access permission for the OCCI Compute Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**CDMI Client removes permission for the OCCI Compute Resource to access the CDMI Container**
	2	check	CDMI Client sends a HTTP PUT request • Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1 • HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0) • If HTTP Accept header is present it is containing the MIME type application/cdmi-container • HTTP Content-Type header is application/cdmi-container • HTTP Body consists of a JSON object which contains the exports field with information on each enabled export protocol as defined in clause 13. It contains information on the OCCI Export Protocol without the URL of the OCCI Compute Resource to be removed
	3	check	CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)
	4	**verify**	**CDMI Client displays success of removing access permission from CDMI Container**
	5	**verify**	**OCCI Compute Resource cannot access CDMI Container anymore**

9.1.4 Delete

9.1.4.1 TD/INTER/OCCI+CDMI/DELETE/001

<table>
<tr><th colspan="4">Interoperability Test Description</th></tr>
<tr><td><b>Identifier:</b></td><td colspan="3"><b>TD/INTER/OCCI+CDMI/DELETE/001</b></td></tr>
<tr><td><b>Objective:</b></td><td colspan="3">Delete an OCCI Compute Resource with an OCCI Storagelink to a CDMI Container</td></tr>
<tr><td><b>Configuration:</b></td><td colspan="3">OCCI_CDMI_CFG_01</td></tr>
<tr><td><b>References:</b></td><td colspan="3">CDMI - ISO/IEC 17826 [4], clause 13.6
OCCI - GFD.184 [2], clause 3.4.3
TD/OCCI/CORE/DELETE/001
TD/INTER/OCCI+CDMI/UPDATE/002</td></tr>
<tr><td><b>Pre-test conditions:</b></td><td colspan="3">CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols
OCCI Server supports linking to CDMI storage
Existing OCCI Compute Resource with an OCCI Storagelink to a CDMI Container
Existing CDMI Container with access permission for the OCCI Compute Resource</td></tr>
<tr><td><b>Test Sequence:</b></td><td><b>Step</b></td><td><b>Type</b></td><td><b>Description</b></td></tr>
<tr><td></td><td><b>1</b></td><td><b>stimulus</b></td><td><b>OCCI Client deletes an OCCI Compute Resource</b></td></tr>
<tr><td></td><td>2</td><td>Check</td><td>OCCI Client sends a HTTP DELETE request
• Request-URI is the location of the OCCI Resource</td></tr>
<tr><td></td><td>3</td><td>Check</td><td>OCCI Server sends a HTTP 200 (OK) response</td></tr>
<tr><td></td><td><b>4</b></td><td><b>verify</b></td><td><b>OCCI Client displays success message</b></td></tr>
<tr><td></td><td><b>5</b></td><td><b>verify</b></td><td><b>OCCI Server has deleted OCCI Compute Resource</b></td></tr>
<tr><td></td><td><b>6</b></td><td><b>stimulus</b></td><td><b>CDMI Client removes access permission for the OCCI Compute Resource from the CDMI Container</b></td></tr>
<tr><td></td><td>7</td><td>Check</td><td>CDMI Client sends a HTTP PUT request
• Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.5.1
• HTTP X-CDMI-Specification-Version contains the CDMI version supported by the CDMI Client (e.g. 1.0.2, 1.5, 2.0)
• If HTTP Accept header is present it is containing the MIME type application/cdmi-container
• HTTP Content-Type header is application/cdmi-container
• HTTP Body consists of a JSON object which contains the exports field with information on each enabled export protocol as defined in clause 13. It contains information on the OCCI Export Protocol without the URL of the OCCI Compute Resource to be removed</td></tr>
<tr><td></td><td>8</td><td>Check</td><td>CDMI Server sends HTTP 202 (ACCEPTED) or HTTP 204 (NO CONTENT)</td></tr>
<tr><td></td><td><b>9</b></td><td><b>verify</b></td><td><b>CDMI Client displays success of removing access permission from CDMI Container</b></td></tr>
</table>

9.1.4.2 TD/INTER/OCCI+CDMI/DELETE/002

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/DELETE/002**		
Objective:	Delete an existing CDMI Container with access permission for an OCCI Compute Resource		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/CDMI/CONTAINER/DELETE/001		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing OCCI Compute Resource with an OCCI Storagelink to a CDMI Container Existing CDMI Container with access permission for the OCCI Compute Resource		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client deletes OCCI Storagelink**
	2	Check	OCCI Client sends a HTTP DELETE request • Request-URI is the location of the OCCI Link
	3	Check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Compute Resources can't access CDMI Container anymore**
	5	**stimulus**	**CDMI Client requests CDMI Server to delete the CDMI Container**
	6	Check	CDMI Client sends a HTTP DELETE request • Request URI is <root URI>/<ContainerName>/< TheContainerName > according to clause 9.6.1
	7	Check	CDMI Server sends a HTTP 204 (NO CONTENT)
	8	**verify**	**CDMI Client displays success of delete operation**
	9	**verify**	**CDMI Server has deleted CDMI Container**

9.1.4.3 TD/INTER/OCCI+CDMI/DELETE/003

Interoperability Test Description			
Identifier:	**TD/INTER/OCCI+CDMI/DELETE/003**		
Objective:	Delete the OCCI Storagelink between an OCCI Compute Resource and a CDMI Container		
Configuration:	OCCI_CDMI_CFG_01		
References:	CDMI - ISO/IEC 17826 [4], clause 13.6 OCCI - GFD.184 [2], clause 3.4.3 TD/OCCI/CORE/DELETE/001		
Pre-test conditions:	CDMI Server supports the OCCI/iSCSI or OCCI/NFSv4 Export Protocols OCCI Server supports linking to CDMI storage Existing OCCI Compute Resource with OCCI Storagelink to a CDMI Container		
Test Sequence:	**Step**	**Type**	**Description**
	1	**stimulus**	**OCCI Client requests OCCI Server to delete the OCCI Storagelink**
	2	Check	OCCI Client sends a HTTP DELETE request • Request-URI is the location of the OCCI Storagelink
	3	Check	OCCI Server sends a HTTP 200 (OK) response
	4	**verify**	**OCCI Client displays success message**
	5	**verify**	**OCCI Compute Resource can't access CDMI Container anymore**

History

Document history		
V1.1.1	April 2013	Publication