SNIA CDMI™ (Cloud Data Management Interface)

Graph Relationships CDMI Extension

Version 0.0.1

ABSTRACT: This CDMI Extension is intended for CDMI 2.0 developers who are considering a standardized way to add structured relationship metadata to CDMI objects. When multiple compatible implementations are demonstrated and approved by the Technical Working Group, this extension will be incorporated into the CDMI standard.

Working Draft

January 26, 2026

USAGE

DISCLAIMER

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

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

Clause 1

Graph relationships

1.1 Overview

Support for graph relationships allows CDMI clients to store structured graph relationship metadata as part of each object's CDMI representation. Graph relationship metadata provides an interoperable way to store and exchange graph relationships, and provides a standardized way to specify graph relationships within CDMI objects, between CDMI objects, and between CDMI objects and external resources.

A CDMI-enabled storage system may optionally implement support for graph relationships. Support for graph relationships is indicated to CDMI-enabled clients by the presence of the `"cdmi_graph_rels"` system-wide capability.

Supporting graph relationships enables following client value:

- Graph relationship metadata consistency: By providing a standard way to represent graph relationships, clients no longer need to develop their own custom formats, which reduces design time.
- Graph relationship metadata richness: By using an open and time-tested standard for representing graph relationships, clients are able to obtain the expressiveness that does not limit their required use cases.
- Graph relationship metadata interoperability: By using a standard way to represent graph relationships, applications can discover, access and manipulate graph relationships in a consistent and constrained way, reducing implementation complexity and improving cross-application compatibility.

1.2 Graph relationship `rel` field format

1.2.1 Top-level `rel` field format

Graph relationship metadata is stored in a `rel` JSON object at the top level of each CDMI object.

```
{
    "rel": { }
}
```

The `rel` field contains zero or one context objects, zero or more RDF triples and zero or more named graphs.

1.2.2 Context object

To reduce the length of predicate strings in RDF triples, an optional `@context` object can specify short name prefixes that are substituted when the corresponding long name is found at the beginning of predicate string values.

```
"@context": {
  "short name": "long name"
}
```

1.2.3 RDF triples

Each RDF triple JSON object shall contain a W3C RDF 1.2 Triple (See https://www.w3.org/TR/rdf12-primer/) specifying a metadata relationships as specified in the W3C RDF 1.1 JSON Alternate Serialization (see https://www.w3.org/TR/rdf-json/).

RDF triples are defined as follows:

```
"S": {"P": [{"value": "O"}, {"type": "T"}, {"datatype": "D"}, {"lang": "L"}]}
```

In each RDF triple, S (subject), P (predicate) and O (object) are mandatory, and T (type), D (datatype) and L (language) are optional.

Subjects, predicates and objects shall support absolute and relative URIs. When using a relative URI with a fragment identifier that refers to the contents of a JSON resource, RFC 6901 JSON pointer URI fragment identifier representations shall be used. For example, `"#/value"` refers to the value of a CDMI object, `"#/metadata/cdmi_size"` refers to the `"cdmi_size"` storage system metadata, and `"#/rel/graph_name"` refers to a named graph with the name `"graph_name"`.

Example RDF triples

An example of a `rel` field with two RDF triples is shown below:

```
{
  "rel": {
    "S": {"P": [{"value": "O"}]},
    "S": {"P": [{"value": "O"}]}
  }
}
```

1.2.4 Named graphs

Each named graph JSON array shall contain zero or more JSON objects, each of which contains zero or one context objects, zero or more RDF triples, and zero or more named graphs.

Named graphs are defined as follows:

```
"graph_name": [{ }]
```

Context objects in named graphs override context objects in the `rel` field.

Example named graph

An example of a `rel` field with two named graphs is shown below:

```
{
  "rel": {
    "graph_name_1": [
      {
        "S": {"P": [{"value": "O"}]}
      }, {
        "S": {"P": [{"value": "O"}]}
      }
    ],
    "graph_name_2": [
      {
        "S": {"P": [{"value": "O"}]}
      }, {
        "sub_graph": [
          {
            "S": {"P": [{"value": "O"}]}
          }
        ]
      }
    ]
  }
}
```

1.3 Examples

EXAMPLE 1: Create a data object that contains a “publisher” graph relationship between the object’s value and SNIA.

HTTP request:

```
PUT /myContainer/document HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
Content-Type: application/cdmi-object

{
  "valuetransferencoding" : "json",
  "mimetype" : "application/json",
  "rel" : {
    "#/value": { "http://purl.org/dc/elements/1.1/publisher": [ {
      "value": "https://www.snia.org/",
      "type": "uri" } ]
    }
  },
  "value" : {
  }
}
```

HTTP response:

```
HTTP/1.1 201 Created
Content-Type: application/cdmi-object

{
  "objectType" : "application/cdmi-object",
  "objectID" : "00007ED90010D891022876A8DE0BC0FD",
  "objectName" : "document",
  "parentURI" : "/myContainer/",
  "parentID" : "00007E7F00102E230ED82694DAA975D2",
  "domainURI" : "/cdmi_domains/MyDomain/",
  "capabilitiesURI" : "/cdmi_capabilities/dataobject/",
  "completionStatus" : "Complete",
  "mimetype" : "application/json",
  "metadata" : {
    "cdmi_size" : "314",
    ...
  },
  "rel" : {
    "#/value": { "http://purl.org/dc/elements/1.1/publisher": [ {
      "value": "https://www.snia.org/",
      "type": "uri" } ]
    }
  }
}
```

EXAMPLE 2: Read a container object that has descriptions of subdirectories.

HTTP request:

```
GET /myContainer/?rel&children HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-container
```

HTTP response:

```
HTTP/1.1 200 OK
Content-Type: application/cdmi-container

{
  "rel" : {
    "@context": {
      "dc": "http://purl.org/dc/terms/"
    },
    "#/children/red": { "dc:description": [ {
      "value": "Images that are red",
      "type": "literal" } ]
      }
    },
    "#/children/green": { "dc:description": [ {
      "value": "Images that are green",
      "type": "literal" } ]
      }
    }
  },
  "children" : [
    "red",
    "green",
    "yellow"
  ]
}
```

1.4 Graph relationship `rel` field validation

A CDMI server that supports graph relationships shall enforce the above-described format, and return a `400 Bad Request` HTTP status code error on object creation or update if non-conforming `rel` field contents are provided.

The following schema shall be used to validate the contents of the `rel` field:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "http://snia.org/cdmi/3.0/rel-schema.json#",
  "definitions": {
    "context": {
        "type": "object",
        "patternProperties": { ".*": { "type": "string" } },
        "additionalProperties": false
    },
    "rdf_triple": {
      "type": "object",
      "patternProperties": {
        ".*": {
          "type": "array",
          "items": {
            "type": "object",
            "properties": {
              "value":    { "type": "string" },
              "type":     { "type": "string" },
              "datatype": { "type": "string" },
              "lang":     { "type": "string" }
            },
            "required": ["value"],
            "additionalProperties": false
          }
        }
      },
      "minProperties": 1,
      "additionalProperties": false
    },
    "named_graph": {
      "type": "array",
      "items": {
        "anyOf": [ { "$ref": "#/definitions/rdf_triple" },
                   { "$ref": "#/definitions/rel",
                     "minProperties": 1 }
                 ]
      },
      "additionalProperties": false
    },
    "rel": {
      "type": "object",
      "properties": { "@context": { "$ref": "#/definitions/context" } },
      "patternProperties": {
        "^((?!@context).)*$": {
          "anyOf": [ { "$ref": "#/definitions/rdf_triple" },
                     { "$ref": "#/definitions/named_graph" }
                   ]
        }
      },
      "additionalProperties": false
    }
  },
  "type": "object",
  "properties": { "rel": { "$ref": "#/definitions/rel" } }
}
```

CDMI implementations that do not support graph relationships shall preserve unsupported fields.

1.5 Backwards compatiblity

Support for graph relationships adds a new top-level `rel` field. This field is ignored by CDMI clients that do not support graph relationships. Only CDMI clients that support graph relationships shall generate and process `rel` fields.

CDMI servers that do not support graph relationships shall preserve `rel` fields without validation.

Table 1.1: Graph Relationships Backwards Compatibility

Client Version	CDMI 1.X Server Mode	CDMI 2.0 Server Mode	CDMI 3.0 Server Mode
CDMI 1.X Client Mode	Default Behavour: • Client support not permitted • Server support not permitted • Server shall preserve `rel` fields when encountered • Client shall ignore `rel` fields when encountered	Not Supported	Not Supported
CDMI 2.0 Client Mode	N/A	If CDMI Server supports graph relationships extension: • Client may attach `rel` fields to CDMI objects • Client may process `rel` fields attached to CDMI objects • Server shall process `rel` fields attached to CDMI objects If CDMI Server does not support graph relationships extension: • Client shall not attach `rel` fields to CDMI objects • Client may process `rel` fields attached to CDMI objects • Server shall preserve `rel` fields when encountered	If CDMI Server supports graph relationships: • Client may attach `rel` fields to CDMI objects • Client may process `rel` fields attached to CDMI objects • Server shall process `rel` fields attached to CDMI objects If CDMI Server does not support graph relationships: • Client shall not attach `rel` fields to CDMI objects • Client may process `rel` fields attached to CDMI objects • Server shall preserve `rel` fields when encountered
CDMI 3.0 Client Mode	N/A	N/A	If CDMI Server supports graph relationships: • Client may attach `rel` fields to CDMI objects. • Client may process `rel` fields attached to CDMI objects. If CDMI Server does not support graph relationships: • Client shall not attach `rel` fields to CDMI objects. • Client may process `rel` fields attached to CDMI objects.

Clause 2

Instructions to the editor

2.1 Overview

In addition to adding the preceding clause, incorporating this extension into the CDMI 2.0 specification requires the following changes to be made to the specification document.

2.2 Normative references

Insert into preamble/normative_refences.txt, as follows:

RFC 6901, *JavaScript Object Notation (JSON) Pointer* - see **:cite:`rfc6901`**

W3C RDF/JSON, *RDF 1.1 JSON Alternate Serialization (RDF/JSON)* - see **:cite:`rdfjson`**

2.3 Terms, acronyms, and definitions

Insert into preamble/terms.txt, as follows:

x.1
graph relationship
|br| metadata that specifies a typed graph arc (where the type is known as the graph predicate) between a graph subject and a graph object **|br|**

x.2
graph subject
|br| the entity being described in a graph relationship **|br|**

x.3
graph object
|br| the entity being associated by the graph predicate to the graph object **|br|**

x.4
graph predicate
|br| the type of relationship between a graph subject and a graph object **|br|**

x.5
named graph
|br| a graph that has an associated name **|br|**

x.6
RDF
|br| Resource Description Framework **|br|**

x.7
RDF triple
|br| three components: a graph subject, a graph predicate, and a graph object **|br|**

2.4 Cloud storage system-wide capabilities

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

Table 2.2: System-wide capabilities

Capability name	Type	Definition
`cdmi_graph_rels`	JSON string	If present and "`true`", indicates that the CDMI server supports graph relationships.