



# SNIA CDMI™ (Cloud Data Management Interface) Exports Provided CDMI Extension

*Version 0.0.1*

**ABSTRACT:** This CDMI Extension is intended for CDMI 2.0 developers who are considering a standardized way to indicate which exports are provided for a given CDMI object. When multiple compatible implementations are demonstrated and approved by the Technical Working Group, this extension will be incorporated into the CDMI standard.

Publication of this Working Draft for review and comment has been approved by the Cloud Storage Technical Working Group. This draft represents a “best effort” attempt by the Cloud Storage Technical Working Group to reach preliminary consensus, and it may be updated, replaced, or made obsolete at any time. This document should not be used as reference material or cited as other than a “work in progress.” Suggestions for revisions should be directed to <http://www.snia.org/feedback/>.

Working Draft

April 10, 2026

## USAGE

Copyright © 2026 SNIA. All rights reserved. All other trademarks or registered trademarks are the property of their respective owners.

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

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

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

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

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

BSD 3-Clause Software License

Copyright (c) 2026, The Storage Networking Industry Association.

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

- \* Redistributions of source code must retain the above copyright notice, this list of conditions and the following disclaimer.
- \* Redistributions in binary form must reproduce the above copyright notice, this list of conditions and the following disclaimer in the documentation and/or other materials provided with the distribution.
- \* Neither the name of The Storage Networking Industry Association (SNIA) nor the names of its contributors may be used to endorse or promote products derived from this software without specific prior written permission.

THIS SOFTWARE IS PROVIDED BY THE COPYRIGHT HOLDERS AND CONTRIBUTORS "AS IS" AND ANY EXPRESS OR IMPLIED WARRANTIES, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE ARE DISCLAIMED. IN NO EVENT SHALL THE COPYRIGHT OWNER OR CONTRIBUTORS BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS SOFTWARE, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE.

## 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

## Exports Provided

### 1.1 Overview

If supported by the CDMI server, exports may be provided for container objects, data objects and queue objects. When an export is provided for a container, the container and all of the container's children can be accessed using the specified export protocol. When an export is provided for a data object or queue object, the data object or queue object can be accessed using the specified export protocol.

In order to identify which data access protocols can be used to access a given CDMI object, a CDMI client must inspect the "exports" specified for each parent container, until the root container is reached. This requires multiple CDMI operations to obtain a complete list of provided exports. The `exports provided` field collects all exports that provide access to the specified CDMI object, eliminating the need for multiple CDMI operations.

A CDMI-enabled storage system may optionally implement support for the `exports provided` field. Support for the `exports provided` field is indicated to CDMI-enabled clients by the presence of the `"cdmi_exports_provided"` system-wide capability.

Supporting the `exports provided` field enables following client value:

- Clients can discover which data access protocols are available to access a given CDMI object with a single CDMI operation.
- Clients can discover additional server-provided information required to access a given CDMI object over a given data access protocol, such as security settings and configuration options.
- Clients can select from an ordered list of multiple data access protocols, where the server can order the list based on the highest performance for the requesting client.
- Clients can discover multiple request locations when data is replicated across multiple locations.

### 1.2 Exports provided field format

#### 1.2.1 Top-level `exports-provided` field format

When a CDMI server supports the `exports provided` field, each request for a CDMI object shall return an `exports-provided` JSON array at the top level of each CDMI object.

```
{
  "exports-provided": [
  ]
}
```

The `exports-provided` field contains zero or more export objects.

## 1.2.2 Export object

Each export object is a JSON object within the `exports-provided` field.

```
{
  "exports-provided": [
    {
    },
    {
    }
  ]
}
```

Each export object contains the following contents:

Table 1.1: Exports Provided - Export Contents

Field Name	Type	Description	Requirement
<code>client_protocol</code>	JSON string	Indicates the type of the data access protocol associated with the export. For example, "NFSv4.1". For details, see clause 13: Exported protocols.	Mandatory
<code>client_protocol</code> ↔ <code>_transport</code>	JSON string	Indicates the underlying network transport used by the data access protocol. This can be used to decide between data access protocols when multiple transports are supported, such as TCP/IP and RDMA. For example, "TCP/IP". For details, see clause 13: Exported protocols.	Optional
<code>client_protocol</code> ↔ <code>_flags</code>	JSON array of JSON strings	Indicates data access protocol-specific flags that are required by the client to successfully connect. For example, "port=2049".	Mandatory
<code>access_uri</code>	JSON string	Indicates the URI that can be used by a client to initiate access using the exported data access protocol. For example, "nfs://example/docs".	Mandatory
<code>access_location</code>	JSON string	Provides a text description of the location of the access URI. For example, "eu-west-3".	Optional
<code>export_</code> ↔ <code>definition</code>	JSON string	Indicates the URI of the CDMI object where the export is defined. For example, "/shares/docs".	Optional

## 1.3 Examples

**EXAMPLE 1:** Retrieve the exports provided for a data object:

HTTP request:

```
GET /shares/docs/mydocument.txt?exports-provided HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-object
```

HTTP response:

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

{
  "exports-provided": [
    {
      "client_protocol": "SMB 3.1",
      "client_protocol_transport": "TCP/IP",
      "client_flags": [ "vers=3.1.1", "multiuser", "sec=krb5", "mfsymlinks" ],
      "access_uri": "smb://example/shares/docs/mydocument.txt",
      "export_definition_uri": "/shares/"
    },
    {
      "client_protocol": "NFSv4.1",
      "client_protocol_transport": "TCP/IP",
      "client_flags": [ "hard", "port=2049", "sec=krb5p" ],
      "access_uri": "nfs://example/docs/mydocument.txt",
      "export_definition_uri": "/shares/docs/"
    },
    {
      "client_protocol": "S3",
      "client_protocol_transport": "TCP/IP",
      "client_flags": [ "rdma-put", "rdma-get" ],
      "access_uri": "https://docs.example.com/mydocument.txt",
      "access_location": "eu-west-3",
      "export_definition_uri": "/shares/docs/"
    },
    {
      "client_protocol": "S3",
      "client_protocol_transport": "TCP/IP",
      "client_flags": [ ],
      "access_uri": "https://docs.s3.us-west-2.amazonaws.com/mydocument.txt",
      "access_location": "us-west-2",
      "export_definition_uri": "/shares/docs/"
    }
  ]
}
```

**EXAMPLE 2:** Retrieve the exports provided for a container object:

HTTP request:

```
GET /shares/?exports-provided HTTP/1.1
Host: cloud.example.com
Accept: application/cdmi-container
```

HTTP response:

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

```
{
  "exports-provided": [
    {
      "client_protocol": "SMB 3.1",
      "client_protocol_transport": "TCP/IP",
      "client_flags": [ "vers=3.1.1", "multiuser", "sec=krb5", "mfsymlinks" ],
      "access_uri": "smb://example/shares/",
      "export_definition_uri": "/shares/"
    }
  ]
}
```

DRAFT

## 1.4 Exports provided metadata validation

A CDMI server that supports exports provided shall validate the server-generated "exports-provided" field against the following schema:

```
{
  "$schema": "http://json-schema.org/draft-04/schema#",
  "id": "http://snia.org/cdmi/3.0/cdmi-exports-provided-schema.json#",
  "definitions": {
    "field_exports_provided": {
      "type": "array",
      "items": {
        "type": "object",
        "properties": {
          "client_protocol": {
            "type": "string"
          },
          "client_protocol_transport": {
            "type": "string"
          },
          "client_flags": {
            "type": "array",
            "items": {
              "type": "string"
            }
          },
          "access_uri": {
            "type": "string"
          },
          "access_location": {
            "type": "string"
          },
          "export_definition_uri": {
            "type": "string"
          }
        },
        "required": [ "client_protocol", "client_flags", "access_uri" ],
        "additionalProperties": false
      }
    }
  },
  "type": "object",
  "properties": { "exports-provided": { "$ref": "#/definitions/field_exports_provided" } }
}
```

## Clause 2

# Instructions to the editor

### 2.1 Overview

When incorporated into the CDMI standard, the clause numbers in table 1.1 shall be changed to reflect updated clause numbering.

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 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
<code>cdmi_exports_provided</code>	JSON string	If present and “true”, indicates that the CDMI server supports exports provided.