

1



2

CORS CDMI Extension

Version 2.0

3

4

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

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

11

SNIA Working Draft

12

November 4, 2020

13 USAGE

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

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

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

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

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

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

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

30 BSD 3-Clause Software License

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

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

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

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

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

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

48 **DISCLAIMER**

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

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

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

Contents

56

57	Clause 1: CORS CDMI Extension	1
58	1.1 Overview	1
59	1.2 Instructions to the Editor	1
60	Clause 2: Cross-Origin Request Sharing (CORS)	4
61	2.1 Overview	4
62	2.2 Non-Preflight Operations	4
63	2.3 Preflight Operations	4
64	2.4 Cross-Origin Rules	5

Clause 1

CORS CDMI Extension

1.1 Overview

HTTP user agents, including HTTP proxies, web browsers, and secure web applications, commonly apply same-origin restrictions to network requests. These restrictions prevent a client-side web application loaded from one origin from obtaining or modifying data retrieved from another origin. These restrictions also limit unsafe HTTP requests that can be performed against destinations that differ from the running application's origin. Many web browsers also treat requests with custom headers as unsafe HTTP requests. To operate in a multi-origin environment, CDMI servers must comply with the W3C Cross-Origin Resource Sharing (CORS) Specification.

This extension specifies how CDMI works with the CORS Specification.

1.2 Instructions to the Editor

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

1. Insert into preamble/normative_references.txt, as follows:

RFC 6454, *The Web Origin Concept* - see [rfc6454]

REC-cors-20140116, *Cross-Origin Resource Sharing* - see [REC-cors-20140116]

2. Insert into preamble/terms.txt, as follows:

x.x

Cross-Origin Request Sharing (CORS) |br| A method by which resources hosted on one domain can be permitted to be accessed by resources hosted on a second, different domain |br|

3. Insert into references/normative.bib and references/refs.bib, as follows:

```
@Misc{rfc6454, author = {Adam Barth}, title = {{The Web Origin Concept}}, howpublished = {RFC 6454}, month = dec,
year = {2011}, abstract = {This document defines the concept of an "origin", which is often used as the scope
of authority or privilege by user agents. Typically, user agents isolate content retrieved from different origins
to prevent malicious web site operators from interfering with the operation of benign web sites. In addition to
outlining the principles that underlie the concept of origin, this document details how to determine the origin of
a URI and how to serialize an origin into a string. It also defines an HTTP header field, named "Origin", that
indicates which origins are associated with an HTTP request.}, doi = {10.17487/RFC6454}, number = {6454},
owner = {Peter van Liesdonk}, pagetotal = {20}, publisher = {RFC Editor}, series = {Request for Comments}, url
= {https://rfc-editor.org/rfc/rfc6454.txt},
}
```

4. Insert into references/refs.bib, as follows:

```
@Misc{REC-cors-20140116, author = {Anne van Kesteren}, title = {Cross-Origin Resource Sharing}, howpub-
lished = {TR/2014/REC-cors-20140116/}, month = jan, year = {2014}, url = {https://www.w3.org/TR/2014/
REC-cors-20140116/}, abstract = {This document defines a mechanism to enable client-side cross-origin re-
quests. Specifications that enable an API to make cross-origin requests to resources can use the algo-
rithms defined by this specification. If such an API is used on http://example.org resources, a resource on
```

101 <http://hello-world.example> can opt in using the mechanism described by this specification (e.g., specifying Access-
 102 Control-Allow-Origin: <http://example.org> as response header), which would allow that resource to be fetched
 103 cross-origin from <http://example.org>}, publisher = {World Wide Web Consortium}
 104 }

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

Table 1: System-wide capabilities

Capability name	Type	Definition
<code>cdmi_cors</code>	JSON string	If present and "true", indicates that the cloud storage system supports CORS.

106 6. Add an entry to the end of the table starting on line 451 of `cdmi_advanced/cdmi_capability_object.txt`, as follows:

Table 2: Capabilities for data system metadata

Capability name	Type	Definition
<code>cdmi_cors_methods</code>	JSON array of JSON strings	When the cloud storage system supports the <code>cdmi_cors_methods</code> data system metadata as defined in <code>ref_support_for_data_system_metadata</code> , the <code>cdmi_cors_methods</code> capability shall be present and contain a list of HTTP methods supported. When this capability is absent, or present and is an empty JSON array, <code>cdmi_cors_methods</code> data system metadata shall not be used. When a cloud storage system supports CORS, the system-wide capability of <code>cdmi_cors</code> specified in <code>ref_cloud_storage_system-wide_capabilities</code> shall be present and set to "true".
<code>cdmi_cors_origins</code>	JSON String	When the cloud storage system supports the <code>cdmi_cors_origins</code> data system metadata as defined in <code>ref_support_for_data_system_metadata</code> , the <code>cdmi_cors_origins</code> capability shall be present and set to the string value "true". When this capability is absent, or present and set to the string value "false", <code>cdmi_cors_origins</code> data system metadata shall not be used.
<code>cdmi_cors_headers</code>	JSON String	When the cloud storage system supports the <code>cdmi_cors_headers</code> data system metadata as defined in <code>ref_support_for_data_system_metadata</code> , the <code>cdmi_cors_headers</code> capability shall be present and set to the string value "true". When this capability is absent, or present and set to the string value "false", <code>cdmi_cors_headers</code> data system metadata shall not be used.

107 7. Add an entry to the end of the table starting on line 216 of `cdmi_advanced/cdmi_metadata.txt`, as follows:

Table 3: Data system metadata

Metadata name	Type	Description	Requirement
cdmi_cors_methods	JSON array of JSON strings	<p>If this data system metadata item is present, it indicates that contained CORS request methods are permitted, and shall be present in the “Access-Control-Allow-Methods” header.</p> <p>When this data system metadata item is absent, the “Access-Control-Allow-Methods” header shall also be absent.</p> <p>Supported request methods are expressed as JSON strings, as defined in section 9 of [rfc2616].</p>	Optional
cdmi_cors_origins	JSON array of JSON strings	<p>If this data system metadata item is present, it indicates that the contained CORS source origins are permitted, and shall be present in the “Access-Control-Allow-Origin” header.</p> <p>When this data system metadata item is absent, the “Access-Control-Allow-Origin” header shall also be absent.</p> <p>Supported source origins are expressed as JSON strings, as defined in section 4 of :cite:`rfc6454`, with “*” indicating that all source origins are permitted.</p>	Optional
cdmi_cors_headers	JSON array of JSON strings	<p>If this data system metadata item is present, it indicates that the contained headers are permitted, and shall be present in the “Access-Control-Allow-Headers” header.</p> <p>When this data system metadata item is absent, the “Access-Control-Allow-Headers” header shall also be absent.</p> <p>Supported request headers are expressed as JSON strings, as defined in section 5.3 of :cite:`rfc2616`.</p>	Optional

108

8. Create new clause, “cdmi_cors.txt” after existing clause 25 “Data Object Versions”, as follows.

109 Clause 2

110 Cross-Origin Request Sharing (CORs)

111 2.1 Overview

112 HTTP clients that conform to the W3C Cross-Origin Resource Sharing (CORs) Specification, such as web browsers and
113 web applications, apply restrictions on network requests sent to a different domain (origin) than the initiating resource
114 is hosted from. These network requests require additional headers (and sometimes additional “preflight” requests) to
115 ensure that the server will permit a cross-domain operation to be performed.

116 As CDMI is based on HTTP, in order to operate in a multi-origin environment, CDMI servers shall provide these headers,
117 as defined in the W3C Cross-Origin Resource Sharing Specification, in order to permit cross-origin HTTP operations.

118 2.2 Non-Preflight Operations

119 Non-preflight operations (GET, HEAD and POST) require the HTTP client to include a new “Origin” header, indicating
120 the domain where the initiating resource is hosted, and requires the server to return an “Access-Control-Allow-
121 Origin” header to indicate if the operation is permitted.

122 EXAMPLE 1: Non-preflight request

```
--> GET /cdmi/2.0.0/data.txt HTTP/1.1
--> Host: cloud.example.com
--> Accept: */*
--> Origin: http://app.example.com

<-- HTTP/1.1 200 OK
<-- Access-Control-Allow-Origin: *
<-- Content-Type: text/plain
<-- Content-Length: 6
<--
<-- [text]
```

123 2.3 Preflight Operations

124 Preflight operations (PUT, DELETE and OPTIONS, as well as when custom headers are used) require the HTTP client
125 to perform a “preflight” operation. This preflight operation is a OPTIONS operation with “Access-Control-Request-
126 Method” and “Access-Control-Request-Headers” headers, and is responded to with “Access-Control-
127 Allow-Origin”, “Access-Control-Allow-Methods”, and “Access-Control-Allow-Headers” headers.

128 If the preflight operation is successful, and the returned headers indicate that the CORs request is permitted, the HTTP
129 client will then perform the original operation, and shall include an “Origin” header, and requires the server to return
130 an “Access-Control-Allow-Origin” header, as with non-preflight operations.

131 EXAMPLE 2: Preflight request


```

--> OPTIONS /cdmi/2.0.0/data.txt HTTP/1.1
--> Host: cloud.example.com
--> Accept: */*
--> Origin: http://app.example.com
--> Access-Control-Request-Method: PUT
--> Access-Control-Request-Headers: Content-Type

<-- HTTP/1.1 200 OK
<-- Access-Control-Allow-Origin: http://app.example.com
<-- Access-Control-Allow-Methods: GET, PUT, DELETE, OPTIONS
<-- Access-Control-Allow-Headers: Content-Type
<-- Access-Control-Max-Age: 3600
    
```

132 This instructs the client that the PUT operation is permitted.

133 **EXAMPLE 3: CORS-enabled PUT**

```

--> PUT /cdmi/2.0.0/data.txt HTTP/1.1
--> Host: cloud.example.com
--> Content-Type: text/plain;charset=utf-8
--> Content-Length: 37
--> Origin: http://app.example.com
-->
--> This is the Value of this Data Object

<-- HTTP/1.1 201 Created
<-- Access-Control-Allow-Origin: *
    
```

134 **2.4 Cross-Origin Rules**

- 135 1. A request shall be considered a CORS request if the “Origin” header is present
- 136 2. A request shall be considered a CORS preflight request if the request type is “OPTIONS” and the “Origin” header
137 is present
- 138 3. For all CORS requests, if the “cdmi_cors_methods” metadata contains the request method, and the
139 “cdmi_cors_origins” metadata matches against the “Origin” request header value (or the metadata item
140 contains “*”), permit the operation and include an “Access-Control-Allow-Origin” response header with
141 the contents of the request “Origin” request header value (or “*” if matched against “*” in the metadata item).
- 142 4. For all CORS preflight requests, if the “cdmi_cors_methods” metadata matches against the “Access-
143 Control-Request-Method” request header value, and the “cdmi_cors_origins” metadata matches
144 against the “Origin” request header value (or the metadata item contains “*”), respond to the request with:
 - 145 • An “Access-Control-Allow-Origin” response header with the contents of the request “Origin” re-
146 quest header value (or “*” if matched against “*” in the metadata item)
 - 147 • An “Access-Control-Allow-Methods” response header with the intersection of the methods
148 specified in the “Access-Control-Request-Method” request header and the methods in the
149 “cdmi_cors_methods” metadata item
 - 150 • An “Access-Control-Allow-Headers” response header with the intersection of the headers
151 specified in the “Access-Control-Request-Headers” request header and the headers in the
152 “cdmi_cors_headers” metadata item
 - 153 • An “Access-Control-Max-Age” response header with a value sufficiently high enough to permit the HTTP
154 Client to perform the corresponding HTTP operation.