

Content Management Interoperability Services (CMIS) Version 1.0 + Errata 1

OASIS Standard

~~1 May 2010~~ 31 March 2011 ~~November 2010~~ 2011

Specification URIs:

This Version:

<http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-v1.0.html>
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-v1.0.pdf>

Previous Version:

<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cs01/cmisis-spec-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cs01/cmisis-spec-v1.0.html>
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cs01/cmisis-spec-v1.0.pdf>

Latest Version:

<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cmisis-spec-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cmisis-spec-v1.0.html>
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cmisis-spec-v1.0.pdf>

Technical Committee:

OASIS Content Management Interoperability Services (CMIS) TC

Chair(s):

David Choy, EMC

Editor(s):

~~Al Brown, IBM~~
~~Ethan Gur-Esh, Microsoft~~
Ryan McVeigh, [OracleZia Consulting](#)
Florian Müller, [OpenTextAlfresco](#)

Related work:

N/A

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/cmisis/core/200908/>
<http://docs.oasis-open.org/ns/cmisis/restatom/200908/>
<http://docs.oasis-open.org/ns/cmisis/messaging/200908/>
<http://docs.oasis-open.org/ns/cmisis/ws/200908/>
<http://docs.oasis-open.org/ns/cmisis/link/200908/>

Abstract:

The Content Management Interoperability Services (CMIS) standard defines a domain model and Web Services and Restful AtomPub bindings that can be used by applications to work with one or more Content Management repositories/systems.

The CMIS interface is designed to be layered on top of existing Content Management systems and their existing programmatic interfaces. It is not intended to prescribe how specific features should be implemented within those CM systems, not to exhaustively expose all of the CM system's capabilities through the CMIS interfaces. Rather, it is intended to define a generic/universal set of capabilities provided by a CM system and a set of services for working with those capabilities.

Status:

This document was last revised or approved by the CMIS TC on the above date. The level of approval is also listed above. Check the "[Latest Version](#)" or "[Latest Approved Version](#)" location noted above for possible later revisions of this document.

Technical Committee members should send comments on this specification to the Technical Committee's email list. Others should send comments to the Technical Committee by using the "[Send A Comment](#)" button on the Technical Committee's web page at <http://www.oasis-open.org/committees/cmis/>.

For information on whether any patents have been disclosed that may be essential to implementing this specification, and any offers of patent licensing terms, please refer to the Intellectual Property Rights section of the Technical Committee web page (<http://www.oasis-open.org/committees/cmis/ipr.php>).

The non-normative errata page for this specification is located at <http://www.oasis-open.org/committees/cmis/>.

Notices

Copyright © OASIS® 2009, 2010. All Rights Reserved.

All capitalized terms in the following text have the meanings assigned to them in the OASIS Intellectual Property Rights Policy (the "[OASIS IPR Policy](#)"). The full Policy may be found at the OASIS website.

This document and translations of it may be copied and furnished to others, and derivative works that comment on or otherwise explain it or assist in its implementation may be prepared, copied, published, and distributed, in whole or in part, without restriction of any kind, provided that the above copyright notice and this section are included on all such copies and derivative works. However, this document itself may not be modified in any way, including by removing the copyright notice or references to OASIS, except as needed for the purpose of developing any document or deliverable produced by an OASIS Technical Committee (in which case the rules applicable to copyrights, as set forth in the OASIS IPR Policy, must be followed) or as required to translate it into languages other than English.

The limited permissions granted above are perpetual and will not be revoked by OASIS or its successors or assigns.

This document and the information contained herein is provided on an "[AS IS](#)" basis and OASIS DISCLAIMS ALL WARRANTIES, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO ANY WARRANTY THAT THE USE OF THE INFORMATION HEREIN WILL NOT INFRINGE ANY OWNERSHIP RIGHTS OR ANY IMPLIED WARRANTIES OF MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE.

OASIS requests that any OASIS Party or any other party that believes it has patent claims that would necessarily be infringed by implementations of this OASIS Committee Specification or OASIS Standard, to notify OASIS TC Administrator and provide an indication of its willingness to grant patent licenses to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification.

OASIS invites any party to contact the OASIS TC Administrator if it is aware of a claim of ownership of any patent claims that would necessarily be infringed by implementations of this specification by a patent holder that is not willing to provide a license to such patent claims in a manner consistent with the IPR Mode of the OASIS Technical Committee that produced this specification. OASIS may include such claims on its website, but disclaims any obligation to do so.

OASIS takes no position regarding the validity or scope of any intellectual property or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; neither does it represent that it has made any effort to identify any such rights. Information on [OASIS](#) procedures with respect to rights in any document or deliverable produced by an OASIS Technical Committee can be found on the OASIS website. Copies of claims of rights made available for publication and any assurances of licenses to be made available, or the result of an attempt made to obtain a general license or permission for the use of such proprietary rights by implementers or users of this OASIS Committee Specification or OASIS Standard, can be obtained from the OASIS TC Administrator. OASIS makes no representation that any information or list of intellectual property rights will at any time be complete, or that any claims in such list are, in fact, Essential Claims.

The names "[OASIS](#)", "[CMIS](#)" are trademarks of OASIS, the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

Table of Contents

1	Introduction	10
1.1	Terminology	10
1.2	Normative References	10
1.3	Non-Normative References	10
2	Domain Model	11
2.1	Data Model.....	11
2.1.1	Repository	11
2.1.1.1	Optional Capabilities	11
2.1.1.2	Implementation Information	14
2.1.2	Object	14
2.1.2.1	Property	15
2.1.3	Object-Type	<u>1746</u>
2.1.3.1	Object-Type Hierarchy and Inheritance	<u>1746</u>
2.1.3.2	Object-Type Attributes	<u>1847</u>
2.1.3.3	Object-Type Property Definitions	19
2.1.4	Document Object.....	<u>2423</u>
2.1.4.1	Content Stream.....	24
2.1.4.2	Renditions.....	<u>2524</u>
2.1.4.3	Document Object-Type Definition	<u>2625</u>
2.1.5	Folder Object	<u>3433</u>
2.1.5.1	File-able Objects	<u>3433</u>
2.1.5.2	Folder Hierarchy	<u>3534</u>
2.1.5.3	Paths.....	<u>3635</u>
2.1.5.4	Folder Object-Type Definition	<u>3736</u>
2.1.6	Relationship Object	<u>4140</u>
2.1.6.1	Relationship Object-Type Definition.....	<u>4241</u>
2.1.7	Policy Object.....	<u>4746</u>
2.1.7.1	Policy Object-Type Definition	<u>4847</u>
2.1.8	Access Control	<u>5150</u>
2.1.8.1	ACL, ACE, Principal, and Permission	<u>5150</u>
2.1.8.2	CMIS Permissions	<u>5150</u>
2.1.8.3	ACL Capabilities	<u>5251</u>
2.1.9	Versioning.....	<u>6160</u>
2.1.9.1	Version Series.....	<u>6160</u>
2.1.9.2	Latest Version.....	<u>6160</u>
2.1.9.3	Major Versions	<u>6160</u>
2.1.9.4	Services that modify Version Series	<u>6261</u>
2.1.9.5	Versioning Properties on Document Objects	<u>6362</u>
2.1.9.6	Document Creation and Initial Versioning State	<u>6463</u>
2.1.9.7	Version Specific/Independent membership in Folders	<u>6463</u>
2.1.9.8	Version Specific/Independent membership in Relationships	<u>6463</u>
2.1.9.9	Versioning visibility in Query Services	<u>6564</u>
2.1.10	Query.....	<u>6564</u>
2.1.10.1	Relational View Projection of the CMIS Data Model	<u>6665</u>
2.1.10.2	Query Language Definition	<u>6766</u>

2.1.10.3 Escaping	<u>7675</u>
2.1.11 Change Log	<u>7875</u>
2.1.11.1 Completeness of the Change Log.....	<u>7876</u>
2.1.11.2 Change Log Token	<u>7876</u>
2.1.11.3 Change Event	<u>7876</u>
2.2 Services	<u>7976</u>
2.2.1 Common Service Elements.....	<u>7977</u>
2.2.1.1 Paging.....	<u>7977</u>
2.2.1.2 Retrieving additional information on objects in CMIS service calls	<u>7977</u>
2.2.1.3 Change Tokens.....	<u>8279</u>
2.2.1.4 Exceptions	<u>8280</u>
2.2.1.5 ACLs	<u>8683</u>
2.2.2 Repository Services	<u>8683</u>
2.2.2.1 getRepositories	<u>8683</u>
2.2.2.2 getRepositoryInfo	<u>8684</u>
2.2.2.3 getTypeChildren.....	<u>8885</u>
2.2.2.4 getTypeDescendants	<u>8886</u>
2.2.2.5 getTypeDefinition	<u>8987</u>
2.2.3 Navigation Services.....	<u>9087</u>
2.2.3.1 getChildren	<u>9087</u>
2.2.3.2 getDescendants	<u>9188</u>
2.2.3.3 getFolderTree	<u>9289</u>
2.2.3.4 getFolderParent	<u>9390</u>
2.2.3.5 getObjectParents	<u>9391</u>
2.2.3.6 getCheckedOutDocs.....	<u>9492</u>
2.2.4 Object Services	<u>9592</u>
2.2.4.1 createDocument.....	<u>9593</u>
2.2.4.2 createDocumentFromSource.....	<u>9694</u>
2.2.4.3 createFolder.....	<u>9895</u>
2.2.4.4 createRelationship	<u>9996</u>
2.2.4.5 createPolicy	<u>10097</u>
2.2.4.6 getAllowableActions.....	<u>10098</u>
2.2.4.7 getObject	<u>10198</u>
2.2.4.8 getProperties.....	<u>10199</u>
2.2.4.9 getObjectByPath	<u>10299</u>
2.2.4.10 getContentStream.....	<u>102400</u>
2.2.4.11 getRenditions	<u>103401</u>
2.2.4.12 updateProperties.....	<u>103401</u>
2.2.4.13 moveObject.....	<u>104402</u>
2.2.4.14 deleteObject.....	<u>105403</u>
2.2.4.15 deleteTree.....	<u>105403</u>
2.2.4.16 setContentStream	<u>106404</u>
2.2.4.17 deleteContentStream	<u>107405</u>
2.2.5 Multi-filing Services	<u>108405</u>
2.2.5.1 addObjectToFolder	<u>108405</u>
2.2.5.2 removeObjectFromFolder	<u>108406</u>
2.2.6 Discovery Services	<u>108406</u>
2.2.6.1 query	<u>108406</u>
2.2.6.2 getContentChanges	<u>110407</u>

2.2.7	Versioning Services	111408
2.2.7.1	checkOut	111408
2.2.7.2	cancelCheckOut	111409
2.2.7.3	checkIn	112409
2.2.7.4	getObjectOfLatestVersion	113440
2.2.7.5	getPropertiesOfLatestVersion	113444
2.2.7.6	getAllVersions	114442
2.2.8	Relationship Services	115442
2.2.8.1	getObjectRelationships	115442
2.2.9	Policy Services	116443
2.2.9.1	applyPolicy	116444
2.2.9.2	removePolicy	116444
2.2.9.3	getAppliedPolicies	116444
2.2.10	ACL Services	117445
2.2.10.1	getACL	117445
2.2.10.2	applyACL	117445
3	Restful AtomPub Binding	119447
3.1	Overview	119447
3.1.1	Namespaces	119447
3.1.2	Authentication	119447
3.1.3	Response Formats	119447
3.1.4	Optional Arguments	120448
3.1.5	Errors and Exceptions	120448
3.1.6	Renditions	120448
3.1.7	Content Streams	120448
3.1.8	Paging of Feeds	120448
3.1.9	Services not exposed	120448
3.2	HTTP	121449
3.2.1	Entity Tag	121449
3.2.2	HTTP Range	121449
3.2.3	HTTP OPTIONS Method	122449
3.2.4	HTTP Status Codes	122449
3.2.4.1	General CMIS Exceptions	122449
3.2.4.2	Specific Exceptions	Error! Bookmark not defined.
3.2.4.3	Notable HTTP Status Codes	122420
3.3	Media Types	122420
3.3.1	CMIS Atom	123420
3.3.2	CMIS Query	124424
3.3.3	CMIS Allowable Actions	124422
3.3.4	CMIS Tree	125423
3.3.5	CMIS ACL	129427
3.4	Atom Extensions for CMIS	130428
3.4.1	Atom Element Extensions	130428
3.4.1.1	AtomPub Workspace	130428
3.4.1.2	Atom Feed	130428
3.4.1.3	Atom Entry	130428
3.4.2	Attributes	131429

3.4.2.1 cmisra:id	131429
3.4.2.2 cmisra:renditionKind	132429
3.4.3 CMIS Link Relations	132430
3.4.3.1 Existing Link Relations	132430
3.4.3.2 Hierarchy Navigation Internet Draft Link Relations	134432
3.4.3.3 Versioning Internet Draft Link Relations	134432
3.4.3.4 CMIS Specific Link Relations	135432
3.5 Atom Resources	136434
3.5.1 Feeds	136434
3.5.2 Entries	137435
3.5.2.1 Hierarchical Atom Entries	138436
3.6 AtomPub Service Document (Repository)	140437
3.6.1 URI Templates	141439
3.6.1.1 Object By Id	142440
3.6.1.2 Object By Path	143441
3.6.1.3 Query	144442
3.6.1.4 Type By Id	145442
3.6.2 HTTP Methods	145443
3.6.2.1 GET	145443
3.7 Service Collections	146443
3.7.1 Root Folder Collection	146443
3.7.2 Query Collection	146443
3.7.2.1 POST	146444
3.7.3 Checked Out Collection	148446
3.7.3.1 GET	149447
3.7.3.2 POST	149447
3.7.4 Unfiled Collection	153450
3.7.4.1 POST	153451
3.7.5 Types Children Collection	157454
3.7.5.1 GET	157455
3.8 Collections	157455
3.8.1 Relationships Collection	158455
3.8.1.1 GET	158456
3.8.1.2 POST	158456
3.8.2 Folder Children Collection	161458
3.8.2.1 GET	162459
3.8.2.2 POST	162460
3.8.3 Policies Collection	170467
3.8.3.1 GET	171468
3.8.3.2 POST	171468
3.8.3.3 DELETE	171468
3.9 Feeds	173470
3.9.1 Object Parents Feed	173470
3.9.1.1 GET	176473
3.9.2 Changes	176473
3.9.2.1 GET	181478
3.9.3 Folder Descendants	181479

3.9.3.1 GET	187184
3.9.3.2 DELETE	187485
3.9.4 Folder Tree	188185
3.9.4.1 GET	191488
3.9.4.2 DELETE	191488
3.9.5 AllVersions Feed	191488
3.9.5.1 GET	193490
3.9.5.2 DELETE	193490
3.9.6 Type Descendants Feed	193490
3.9.6.1 GET	202499
3.10 Resources	202499
3.10.1 Type Entry	202499
3.10.1.1 GET	203499
3.10.2 Document Entry	204201
3.10.2.1 GET	205202
3.10.2.2 PUT	207203
3.10.2.3 DELETE	207204
3.10.3 Document Private Working Copy (PWC) Entry	207204
3.10.3.1 GET	208204
3.10.3.2 PUT	209206
3.10.3.3 DELETE	210206
3.10.4 Folder Entry	210207
3.10.4.1 GET	211207
3.10.4.2 PUT	212209
3.10.4.3 DELETE	212209
3.10.5 Relationship Entry	212209
3.10.5.1 GET	213210
3.10.5.2 PUT	215211
3.10.5.3 DELETE	215211
3.10.6 Policy Entry	215211
3.10.6.1 GET	215212
3.10.6.2 PUT	217213
3.10.6.3 DELETE	217214
3.10.7 Content Stream	217214
3.10.7.1 GET	217214
3.10.7.2 PUT	217214
3.10.7.3 DELETE	218214
3.10.8 ACL Resource	218214
3.10.8.1 GET	218215
4 Web Services Binding	220216
4.1 Overview	220216
4.1.1 WS-I	220216
4.1.2 Authentication	220216
4.1.3 Content Transfer	220216
4.1.4 Reporting Errors	220216
4.2 Web Services Binding Mapping	220216
4.3 Additions to the Services section	220216

4.3.1	updateProperties and checkIn Semantics.....	<u>220216</u>
4.3.2	Content Ranges	<u>220216</u>
4.3.3	Extensions	<u>221217</u>
4.3.4	Web Services Specific Structures	<u>221217</u>
4.3.4.1	cmisFaultType and cmisFault	<u>221217</u>
4.3.4.2	cmisRepositoryEntryType	<u>221217</u>
4.3.4.3	cmisTypeContainer	<u>221217</u>
4.3.4.4	cmisTypeDefinitionListType	<u>221217</u>
4.3.4.5	cmisObjectInFolderType, cmisObjectParentsType and cmisObjectInFolderContainerType	<u>221217</u>
4.3.4.6	cmisObjectListType and cmisObjectInFolderListType	<u>222217</u>
4.3.4.7	cmisContentStreamType	<u>222218</u>
4.3.4.8	cmisACLType.....	<u>222218</u>
4.3.4.9	cmisExtensionType.....	<u>222218</u>
5	IANA Considerations	<u>223219</u>
5.1	Content-Type Registration.....	<u>223219</u>
5.1.1	CMIS Query.....	<u>223219</u>
5.1.2	CMIS AllowableActions	<u>223219</u>
5.1.3	CMIS Tree	<u>224220</u>
5.1.4	CMIS Atom	<u>225221</u>
5.1.5	CMIS ACL.....	<u>226222</u>
6	Conformance.....	<u>228224</u>
A.	Acknowledgements	<u>230226</u>
B.	Non-Normative Text	<u>232228</u>
C.	Revision History.....	<u>233229</u>

1 Introduction

The Content Management Interoperability Services (CMIS) standard defines a domain model and set of bindings that include Web Services and ReSTful AtomPub that can be used by applications to work with one or more Content Management repositories/systems.

The CMIS interface is designed to be layered on top of existing Content Management systems and their existing programmatic interfaces. It is not intended to prescribe how specific features should be implemented within those CM systems, nor to exhaustively expose all of the CM system's capabilities through the CMIS interfaces. Rather, it is intended to define a generic/universal set of capabilities provided by a CM system and a set of services for working with those capabilities.

1.1 Terminology

The key words "MUST", "MUST NOT", "REQUIRED", "SHALL", "SHALL NOT", "SHOULD", "SHOULD NOT", "RECOMMENDED", "MAY", and "OPTIONAL" in this document are to be interpreted as described in RFC2119.

1.2 Normative References

- [RFC4287] M. Nottingham, R. Sayre, *Atom Syndication Format*, <http://www.ietf.org/rfc/rfc4287.txt>, December 2005
- [RFC5023] J. Gregorio, B. de hOra, *Atom Publishing Protocol*, <http://www.ietf.org/rfc/rfc5023.txt>, October 2007
- [RFC2616] R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, *Hypertext Transfer Protocol --HTTP/1.1*, <http://www.ietf.org/rfc/rfc2616.txt>, June 1999
- [RFC2119] S. Bradner, *Key words for use in RFCs to Indicate Requirement Levels*, <http://www.ietf.org/rfc/rfc2119.txt>, March 1997
- [RFC4918] L. Dusseault, *HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)*, June 2007
- [RFC3986] T. Berners-Lee, R. Fielding, L. Masinter, *Unified Resource Identifier*, January 2005
- [ID-Brown] J. Reschke Editor, A. Brown, G. Clemm, *Link Relation Types for Simple Version Navigation between Web Resources*, <http://www.ietf.org/id/draft-brown-versioning-link-relations-07.txt>, 2010
- [ID-WebLinking] M. Nottingham, *Web Linking*, <http://tools.ietf.org/id/draft-nottingham-http-link-header-07.txt>, 2010

1.3 Non-Normative References

2 Domain Model

2.1 Data Model

CMIS provides an interface for an application to access a **Repository**. To do so, CMIS specifies a core data model that defines the *persistent* information entities that are managed by the repository, and specifies a set of basic services that an application can use to access and manipulate these entities. In accordance with the CMIS objectives, this data model does not cover *all* the concepts that a full-function ECM repository typically supports. Specifically, transient entities (such as programming interface objects), administrative entities (such as user profiles), and extended concepts (such as compound or virtual document, work flow and business process, event and subscription) are not included.

However, when an application connects to a CMIS service endpoint, the same endpoint MAY provide access to more than one CMIS repository. (How an application obtains a CMIS service endpoint is outside the scope of CMIS. How the application connects to the endpoint is a part of the protocol that the application uses.) An application MUST use the CMIS “_Get Repositories_” service (*getRepositories*) to obtain a list of repositories that are available at that endpoint. The Repository Identity MUST uniquely identify an available repository at this service endpoint. Both the repository name and the repository identity are opaque to CMIS. Aside from the “_Get Repositories_” service, all other CMIS services are single-repository-scoped, and require a Repository Identity as an input parameter. In other words, except for the “_Get Repositories_” service, multi-repository and inter-repository operations are not supported by CMIS.

2.1.1 Repository

The repository itself is described by the CMIS “_Get Repository Information_” service. The service output is fully described in section 2.2.2.2 *getRepositoryInfo*.

2.1.1.1 Optional Capabilities

Commercial ECM repositories vary in their designs. Moreover, some repositories are designed for a specific application domain and may not provide certain capabilities that are not needed for their targeted domain. Thus, a repository implementation may not necessarily be able to support all CMIS capabilities. A few CMIS capabilities are therefore “_optional_” for a repository to be compliant. A repository’s support for each of these optional capabilities is discoverable using the *getRepositoryInfo* service. The following is the list of these optional capabilities. All capabilities are “_Boolean_” (i.e. the Repository either supports the capability entirely or not at all) unless otherwise noted.

Navigation Capabilities:

capabilityGetDescendants

Ability for an application to enumerate the descendants of a folder via the *getDescendants* service.

[See section: 2.2.3.2 getDescendants](#)

capabilityGetFolderTree

Ability for an application to retrieve the folder tree via the *getFolderTree* service.

[See section: 2.2.3.3 getFolderTree](#)

77 **Object Capabilities:**

78 **capabilityContentStreamUpdatability** (**enumCapabilityContentStreamUpdates**)

79 Indicates the support a repository has for updating a document's content stream. Valid values
80 are:

- 81 • **none**: The content stream may never be updated.
- 82 • **anytime**: The content stream may be updated any time.

83 **pwonly**: The content stream may be updated only when checked out. The abbreviation PWC is
84 described in section 0

85 Versioning

- 86 • Versioning.

87 See Section: 2.1.4.1 Content Stream~~Content Stream~~

88

89 **capabilityChanges** (**enumCapabilityChanges**)

90 Indicates what level of changes (if any) the repository exposes via the "change log" service.
91 Valid values are:

- 92 • **none**: The repository does not support the change log feature.
- 93 • **objectidonly**: The change log can return only the ObjectIDs for changed objects in
94 the repository and an indication of the type of change, not details of the actual change.
- 95 • **properties**: The change log can return properties and the ObjectID for the changed
96 objects
- 97 • **all**: The change log can return the ObjectIDs for changed objects in the repository and
98 more information about the actual change

99 See Section: 2.1.11 Change Log

100

101 **capabilityRenditions** (**enumCapabilityRendition**)

102 Indicates whether or not the repository exposes renditions of document or folder objects.

- 103 • **none**: The repository does not expose renditions at all.
- 104 • **read**: Renditions are provided by the repository and readable by the client.

105

106 **Filing Capabilities:**

107 **capabilityMultifiling**

108 Ability for an application to file a document or other file-able object in more than one folder

109 See Section: 2.1.5 Folder Object

110

111 **capabilityUnfiling**

112 Ability for an application to leave a document or other file-able object not filed in any folder

113 See Section: 2.1.5 Folder Object

114

115 **capabilityVersionSpecificFiling**

116 Ability for an application to file individual versions (i.e., not all versions) of a document in a folder

117 See Section: 0

118 Versioning

119 Versioning

120

121 Versioning Capabilities:

122 `capabilityPWCUpdatable`

123 Ability for an application to update the `"Private Working Copy"` of a checked-out document

124 See Section: 0

125 [Versioning](#)

126 [Versioning](#)

127

128 `capabilityPWCSearchable`

129 Ability of the Repository to include the `"Private Working Copy"` of checked-out documents in
130 query search scope; otherwise PWC:s are not searchable

131 See Section: 0

132 [Versioning](#)

133 [Versioning](#)

134

135 `capabilityAllVersionsSearchable`

136 Ability of the Repository to include all versions of document.If False, typically either the latest or
137 the latest major version will be searchable.

138 See Section: 0

139 [Versioning](#)

140 [Versioning](#)

141

142 Query Capabilities:

143 `capabilityQuery (enumCapabilityQuery)`

144 Indicates the types of queries that the Repository has the ability to fulfill. Query support levels are:

- 145 • **none:** No queries of any kind can be fulfilled.
- 146 • **metadataonly:** Only queries that filter based on object properties can be fulfilled.
147 Specifically, the CONTAINS() predicate function is not supported.
- 148 • **fulltextonly:** Only queries that filter based on the full-text content of documents can be
149 fulfilled. Specifically, only the CONTAINS() predicate function can be included in the
150 WHERE clause.
- 151 • **bothseparate:** The repository can fulfill queries that filter EITHER on the full-text content
152 of documents OR on their properties, but NOT if both types of filters are included in the
153 same query.
- 154 • **bothcombined:** The repository can fulfill queries that filter on both the full-text content of
155 documents and their properties in the same query.

156 See Section: 2.1.10 Query

157

158 `capabilityJoin (enumCapabilityJoin)`

159 Indicates the types of JOIN keywords that the Repository can fulfill in queries. Support levels are:

- 160 • **none:** The repository cannot fulfill any queries that include any JOIN clauses.
- 161 • **inneronly:** The repository can fulfill queries that include an INNER JOIN clause, but
162 cannot fulfill queries that include other types of JOIN clauses.

- **innerandouter**: The repository can fulfill queries that include any type of JOIN clause defined by the CMIS query grammar.

See Section: 2.1.10 Query

ACL Capabilities:

capabilityACL (enumCapabilityACL)

Indicates the level of support for ACLs by the repository

- **none**: The repository does not support ACL services
- **discover**: The repository supports discovery of ACLs (getACL and other services)
- **manage**: The repository supports discovery of ACLs AND applying ACLs (getACL and applyACL services)

See Section: 2.8 Access Control

2.1.1.2 Implementation Information

The `__Get Repository Information__` service MUST also return implementation information including vendor name, product name, product version, version of CMIS that it supports, the root folder ID (see section 2.1.5.2 Folder Hierarchy), and MAY include other implementation-specific information. The version of CMIS that the repository supports MUST be expressed as a Decimal that matches the specification version.

2.1.2 Object

The entities managed by CMIS are modeled as typed **Objects**. There are four base types of objects: **Document Objects**, **Folder Objects**, **Relationship Objects**, and **Policy Objects**.

- A *document object* represents a standalone information asset. Document objects are the elementary entities managed by a CMIS repository.
- A *folder object* represents a logical container for a collection of `__file-able__` objects, which include folder objects and document objects. Folder objects are used to organize file-able objects. Whether or not an object is file-able is specified in its [object-type definition](#).
- A *relationship object* represents an instance of directional relationship between two objects. The support for relationship objects is optional, and may be discovered via the `__Get Type Children__` service.
- A *policy object* represents an administrative policy, which may be `__applied__` to one or more `__controllablePolicy__` objects. Whether or not an object is controllable is specified in its object-type definition. The support for policy objects is optional, and may be discovered via the `__Get Type Children__` service.

Additional object-types MAY be defined in a repository as subtypes of these base types. CMIS services are provided for the discovery of object-types that are defined in a repository. However, object-type management services, such as the creation, modification, and deletion of an object-type, are outside the scope of CMIS.

Every CMIS object has an opaque and immutable **Object Identity** (ID), which is assigned by the repository when the object is created. An ID uniquely identifies an object within a repository regardless of the type of the object. Repositories SHOULD assign IDs that are `__permanent__` – that is, they remain unchanged during the lifespan of the identified objects, and they are never reused or reassigned after the objects are deleted from the repository.

205 Every CMIS object has a set of named, but not explicitly ordered, **Properties**. (However, a Repository
206 SHOULD always return object properties in a consistent order.) Within an object, each property is
207 uniquely identified by its property definition id.

208 In addition, a document object MAY have a **Content-Stream**, which may be used to hold a raw digital
209 asset such as an image or a word-processing document. A repository MUST specify, in each object-type
210 definition, whether document objects of that type MAY, MUST, or MUST NOT have a content-stream. A
211 document MAY also have one or more **Renditions** associated with it. A rendition can be a thumbnail or
212 an alternate representation of the content stream.

213 Document or folder objects MAY have one **Access Control List** (ACL), which controls access to the
214 document or folder. A policy object may also control access to the document or folder. An ACL
215 represents a list of **Access Control Entries** (ACEs). An ACE in turn represents one or more permissions
216 being granted to a **principal** (a user, group, role, or something similar).

217 The notion of localization of the objects in the data model is entirely repository specific.

218 [CMIS objects MAY expose additional information, such as vendor-specific workflow data, beyond the](#)
219 [attributes described above. In this respect, the data model can be extended as desired. This specification](#)
220 [does not standardize such extensions.](#)

221 2.1.2.1 Property

222 A property MAY hold zero, one, or more typed data value(s). Each property MAY be *single-valued* or
223 *multi-valued*. A single-valued property contains a single data value, whereas a multi-valued property
224 contains an ordered list of data values of the same type. The ordering of values in a multi-valued property
225 MAY be preserved by the repository.

226 ~~If a value is not provided for a property, the property is in a "value not set" state. There is no "null" value~~
227 ~~for a property. Through protocol binding, a property is either not set, or is set to a particular value or a list~~
228 ~~of values. A property, either single-valued or multi-valued, MAY be in a "not set" state. CMIS does not~~
229 ~~support "null" property value.~~

230 ~~If a multi-valued property is not in a "not set" state, its property value MUST be a non-empty list of~~
231 ~~individual values. Each individual value in the list MUST NOT be in a "not set" state and MUST conform~~
232 ~~to the property's property-type.~~

233

234 A multi-valued property is either set or not set in its entirety. An individual value of a multi-valued property
235 MUST NOT be in an individual "value not set" state and hold a position in the list of values. An empty list
236 of values MUST NOT be allowed.

237 Every property is typed. The Property-type defines the data type of the data value(s) held by the property.
238 CMIS specifies the following Property-types. They include the following data types defined by "XML
239 Schema Part 2: Datatypes Second Edition" (W3C Recommendation, 28 October 2004,
240 <http://www.w3.org/TR/xmlschema-2/>):

- 241 • string (xsd:string)
- 242 • boolean (xsd:boolean)
- 243 • decimal (see section 2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions)
- 244 • integer (xsd:integer)
- 245 • datetime (xsd:dateTime and see section 2.1.3.3.5 Attributes specific to Decimal Object-Type
- 246 Property Definitions)
- 247 • uri (xsd:anyURI)

248

249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269

In addition, the following Property-Types are also specified by CMIS:

- id
- html

Individual protocol bindings MAY override or re-specify these property types.

All properties MUST supply a String **queryName** attribute which is used for query and filter operations on object-types. This is an opaque String with limitations. This string SHOULD NOT contain any characters that negatively interact with the BNF grammar.

The string MUST NOT contain:

- whitespace `" "`,
- comma `" , "`
- double quotes `"\""`
- single quotes `"'"`
- backslash `"\""`
- the period `" . "` character or,
- the open `" ("` or close `") "` parenthesis characters.

2.1.2.1.1 ID Property

An ID property holds a system-generated, read-only identifier, such as an Object ID, an Object-Type ID, etc. (The ID Property-Type is NOT defined by xsd:id.) The lexical representation of an ID is an opaque string. As such, an ID cannot be assumed to be interpretable syntactically or assumed to be collatable with other IDs, and can only be used in its entirety as a single atomic value. When used in a query predicate, an ID can only participate in an `" = "` or a `" ≠ "` comparison with a string literal or with another ID.

While all CMIS identities share the same Property-Type, they do not necessarily share the same address space. Unless explicitly specified, ID properties NEED NOT maintain a referential integrity constraint. Therefore, storing the ID of one object in another object NEED NOT constrain the behavior of either object. A repository MAY, however, support referential constraint underneath CMIS if the effect on CMIS services remains consistent with an allowable behavior of the CMIS model. For example, a repository MAY return an exception when a CMIS service call violates an underlying referential constraint maintained by the repository. In that case, an error message SHOULD be returned to the application to describe the cause of exception and suggest a remedial action. The content of such messages is outside the scope of CMIS.

2.1.2.1.2 HTML Property

An HTML property holds a document or fragment of Hypertext Markup Language (HTML) content. HTML properties are not guaranteed to be validated in any way. The validation behavior is entirely repository specific.

290 2.1.3 Object-Type

291 An **Object-Type** defines a fixed and non-hierarchical set of properties ("schema") that all objects of that
292 type have. This schema is used by a repository to validate objects and enforce constraints, and is also
293 used by a user to compose object-type-based (structured) queries.

294 All CMIS objects are strongly typed. If a property not specified in an object's object-type definition is
295 supplied by an application, an exception SHOULD be thrown.

296 Each object-type is uniquely identified within a repository by a system-assigned and immutable **Object-**
297 **Type Identifier**, which is of type ID.

298 A CMIS repository MUST expose exactly one collection of Object-Types via the "Repository" services
299 (*getTypeChildren*, *getTypeDescendants*, *getTypeDefinition*).

300 While a repository MAY define additional object-types beyond the [CMIS Base Object-Types](#), these
301 Object-Types MUST NOT extend or alter the behavior or semantics of a CMIS service (for example, by
302 adding new services). A repository MAY attach additional constraints to an object-type underneath CMIS,
303 provided that the effect visible through the CMIS interface is consistent with the allowable behavior of
304 CMIS.

305 2.1.3.1 Object-Type Hierarchy and Inheritance

306 **Hierarchy** and **Inheritance** for Object-Types are supported by CMIS in the following manner:

- 307 • A CMIS repository MUST have these base types:
 - 308 ○ *cmis:document* object-type
 - 309 ○ *cmis:folder* object-type
- 310 • A CMIS repository MAY have these base types:
 - 311 ○ *cmis:relationship* object-type
 - 312 ○ *cmis:policy* object-type
- 313 • Additional base types MUST NOT exist. Additional object-types MAY be defined as sub-types or
314 descendant types of these four base types.
- 315 • A **Base Type** does not have a parent type.
- 316 • A non-base type has one and only one parent type. An object-type's **Parent Type** is a part of the
317 object-type definition.
- 318 • An object-type definition includes a set of [object-type attributes](#) (e.g. Fileable, Queryable, etc.)
319 and a property schema that will apply to Objects of that type.
 - 320 ○ There is no inheritance of object-type attributes from a parent object-type to its sub-types.
- 321 • The properties of a CMIS base type MUST be inherited by its descendant types.
- 322 • A **Child Type** whose immediate parent is NOT its base type SHOULD inherit all the property
323 definitions that are specified for its parent type. In addition, it MAY have its own property
324 definitions.
 - 325 ○ If a property is NOT inherited by a subtype, the exhibited behavior for query MUST be as if
326 the value of this property is "not set" for all objects of this sub-type.
- 327 • The scope of a query on a given object-type is automatically expanded to include all the
328 **Descendant Types** of the given object-type with the attribute `includedInSuperTypeQuery`
329 equals TRUE. This was added for synthetic types as well as to support different type hierarchies
330 that are not necessarily the same as CMIS. Only the properties of the given object-type,

331 including inherited ones, MUST be used in the query. Properties defined for its descendant types
332 MAY NOT be used in the query, and CAN NOT be returned by the query.

333 ○ If a property of its parent type is not inherited by this type, the property MUST still appear as
334 a column in the corresponding virtual table in the relational view, but this column MUST
335 contain a NULL value for all objects of this type. (See section 2.1.10 Query.)

336 2.1.3.2 Object-Type Attributes

337 2.1.3.2.1 Attributes common to ALL Object-Type Definitions

338 All **Object-Type Definitions** MUST contain the following **attributes**:

339 **id** ID

340 This opaque attribute identifies this object-type in the repository.

341

342 **localName** String ~~(optional)~~

343 This attribute represents the underlying repository's name for the object-type. This field is
344 opaque and has no uniqueness constraint imposed by this specification.

345 Two properties with the same localName and localNamespace MUST have the same semantic
346 equality.

347

348 **localNamespace** String (optional)

349 This attribute allows repositories to represent the internal namespace of the underlying
350 repository's name for the object-type.

351

352 **queryName** String

353 Used for query and filter operations on object-types. This is an opaque String with limitations.

354 This string SHOULD NOT contain any characters that negatively interact with the BNF grammar.

355

356 The string MUST NOT contain:

- 357 • whitespace `" "`,
- 358 • comma `" , "`
- 359 • double quotes `" "" "`
- 360 • single quotes `" '' "`
- 361 • backslash `" \ "`
- 362 • the period `" . "` character or,
- 363 • the open `" ("` or close `")" "` parenthesis characters.

364

365 **displayName** String (optional)

366 Used for presentation by application.

367

368 **baseId** Enum

369 A value that indicates whether the base type for this Object-Type is the Document, Folder,
370 Relationship, or Policy base type.

371

372 **parentId** ID

373 | The ID of the Object-Type's immediate parent type.
374 | It MUST be `"not set"` for a base type.
375 |

376 | **description** String (optional)
377 | Description of this object-type, such as the nature of content, or its intended use. Used for
378 | presentation by application.
379 |

380 | **creatable** Boolean
381 | Indicates whether new objects of this type MAY be created. If the value of this attribute is FALSE,
382 | the repository MAY contain objects of this type already, but MUST NOT allow new objects of this
383 | type to be created.
384 |

385 | **fileable** Boolean
386 | Indicates whether or not objects of this type are `file-able`.
387 |

388 | **queryable** Boolean
389 | Indicates whether or not this object-type can appear in the FROM clause of a query statement. A
390 | non-queryable object-type is not visible through the relational view that is used for query, and
391 | CAN NOT appear in the FROM clause of a query statement.
392 |

393 | **controllablePolicy** Boolean
394 | Indicates whether or not objects of this type are controllable via policies. Policy objects can only
395 | be applied to controllablePolicy objects.
396 |

397 | **controllableACL** Boolean
398 | This attribute indicates whether or not objects of this type are controllable by ACL's. Only objects
399 | that are controllableACL can have an ACL.
400 |

401 | **fulltextIndexed** Boolean
402 | Indicates whether objects of this type are indexed for full-text search for querying via the
403 | CONTAINS() query predicate.
404 |

405 | **includedInSupertypeQuery** Boolean
406 | Indicates whether this type and its subtypes appear in a query of this type's ancestor types.
407 | For example: if Invoice is a sub-type of cmis:document, if this is TRUE on Invoice then for a query
408 | on cmis:document, instances of Invoice will be returned if they match.
409 | If this attribute is FALSE, no instances of Invoice will be returned even if they match the query.

410 | 2.1.3.3 Object-Type Property Definitions

411 | Besides these object-type attributes, an object-type definition SHOULD contain inherited property
412 | definitions and zero or more additional property definitions. All the properties of an object, including
413 | inherited properties, MUST be retrievable through the `"get"` services, and MAY appear in the SELECT
414 | clause of a query.

415 2.1.3.3.1 Property Types

416 Property types are defined in section 2.1.2.1 Property.

417 2.1.3.3.2 Attributes common to ALL Object-Type Property Definitions

418 All **Object-Type Property Definitions** MUST contain the following **attributes**:

419 **id** ID

420 This opaque attribute uniquely identifies the property in the repository. If two Object-Types each
421 contain property definitions with the same ID, those property definitions are the same.

422

423 **localName** String (optional)

424 This attribute represents the underlying repository's name for the property. This field is opaque
425 and has no uniqueness constraint imposed by this specification.

426

427 **localNamespace** String (optional)

428 This attribute allows repositories to represent the internal namespace of the underlying
429 repository's name for the property.

430

431 **queryName** String

432 Used for query operations on properties. This is an opaque String with limitations. Please see
433 `queryName` in Object-Type Attributes for the limitations on what characters are not allowed.

434

435 **displayName** String (optional)

436 Used for presentation by application.

437

438 **description** String (optional)

439 This is an optional attribute containing a description of the property

440

441 **propertyType** Enum

442 This attribute indicates the type of this property. It MUST be one of the allowed property types.
443 (See section 2.1.2.1 Property.)

444

445 **cardinality** Enum

446 Indicates whether the property can have `"zero or one"` or `"zero or more"` values.

447 Values:

- 448 • **single**: Property can have zero or one values (if property is not required), or exactly one
449 value (if property is required)
- 450 • **multi**: Property can have zero or more values (if property is not required), or one or more
451 values (if property is required).

452 Repositories SHOULD preserve the ordering of values in a multi-valued property. That is, the
453 order in which the values of a multi-valued property are returned in get operations SHOULD be
454 the same as the order in which they were supplied during previous create/update operation.

455

456 **updatability** Enum

457 Indicates under what circumstances the value of this property MAY be updated.

- 458 Values:
- 459 • **readonly:** The value of this property MUST NOT ever be set directly by an application. It
 - 460 is a system property that is either maintained or computed by the repository.
 - 461 ○ The value of a readOnly property MAY be indirectly modified by other repository
 - 462 interactions (for example, calling `updateProperties()` on an object will change the
 - 463 object's last modified date, even though that property cannot be directly set via an
 - 464 `updateProperties()` service call.)
 - 465 • **readwrite:** The property value can be modified using the `updateProperties` service.
 - 466 4. **whencheckedout:** The property value MUST only be update-able using a `private`
 - 467 `working copy` Document.
 - 468 a. i.e. the update is either made on a `private working copy` object or made using a
 - 469 `check in` service.
 - 470 2. **oncreate:** The property value MUST only be update-able during the Create operation on
 - 471 that Object.

472

473 **inherited** Boolean

474 Indicates whether the property definition is inherited from the parent-type when TRUE or it is

475 explicitly defined for this object-type when FALSE.

476

477 **required** Boolean

478

479 This attribute is only applicable to non-sytem properties, i.e. properties whose value is provided

480 by the application.

481 If TRUE, then the value of this property MUST never be set to the `not set` state when an object

482 of this type is created/updated. If not provided during a create or update operation, the repository

483 MUST provide a value for this property.

484 If a value is not provided, then the default value defined for the property MUST be set. If no

485 default value is provided and no default value is defined, the repository MUST throw an

486 exception.

487 This attribute is not applicable when the `updatability` attribute is `readonly`. In that case,

488 `required` SHOULD be set to FALSE.

489 Note: For CMIS-defined object types, the value of a system property (such as `cmis:objectId`,

490 `cmis:createdBy`) MUST be set by the repository. However, the property's `required` attribute

491 SHOULD be FALSE because it is read-only to applications.

492

493 **queryable** Boolean

494 Indicates whether or not the property MAY appear in the WHERE clause of a CMIS query

495 statement.

496 This attribute MUST have a value of FALSE if the Object-type's attribute for `Queryable` is set

497 to FALSE.

498

499 **orderable** Boolean

500 Indicates whether the property can appear in the ORDER BY clause of a CMIS query statement

501 or an ORDERBY parameter.

502 This property MUST be FALSE for any property whose cardinality is `multi`.

503

504 **choices** <PropertyChoiceType list> (multi-valued)

505 Indicates an explicit ordered set of single values allowed for this property.

506 | If the cardinality of the property definition is `"single"` and the `"openChoice"` attribute is FALSE,
507 | then the property value MUST be at most one of the values listed in this attribute.

508 | If the cardinality of the property definition is `"single"` and the `"openChoice"` attribute is TRUE,
509 | then the property value MAY be one of the values listed in this attribute.

510 | If the cardinality of the property definition is `"multi"` and the `"openChoice"` attribute is FALSE,
511 | then the property value MUST be zero, one or more than one of the values listed in this attribute.

512 | If the cardinality of the property definition is `"multi"` and the `"openChoice"` attribute is TRUE,
513 | then the property value MAY be zero, one, or more than one of the values listed in this attribute. If
514 | this attribute is `"not set"`, then any valid value for this property based on its type may be used.

515 Each choice includes a `displayName` and a value. The `displayName` is used for presentation
516 | purpose. The value will be stored in the property when selected.

517 | Choices MAY be hierarchically presented. For example: a value of `"choices"` for a geographic
518 | location would be represented as follows:

519 | o Europe:

520 | ▪ England

521 | ▪ France

522 | ▪ Germany

523 | o North America

524 | ▪ Canada

525 | ▪ USA

526 | ▪ Mexico

527 **openChoice** Boolean

528 | This attribute is only applicable to properties that provide a value for the `"Choices"` attribute.
529 | If FALSE, then the data value for the property MUST only be one of the values specified in the
530 | `"Choices"` attribute. If TRUE, then values other than those included in the `"Choices"` attribute
531 | may be set for the property.

532

533 **defaultValue** <PropertyType>

534 The value that the repository MUST set for the property if a value is not provided by an
535 | application when the object is created.

536 If no default value is specified and an application creates an object of this type without setting a
537 | value for the property, the repository MUST attempt to store a `"value not set"` state for the
538 | property value. If this occurs for a property that is defined to be required, then the creation
539 | attempt MUST throw an exception.

540 The attributes on the default value element are the same as the attributes on the property
541 | definition.

542 2.1.3.3.3 Attributes specific to Integer Object-Type Property Definitions

543 The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is
544 | `"Integer"`, in addition to the common attributes specified above. A repository MAY provide additional
545 | guidance on what values can be accepted. If the following attributes are not present the repository
546 | behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

547 **minValue** Integer

548 The minimum value allowed for this property.

549 If an application tries to set the value of this property to a value lower than `minValue`, the
550 repository MUST throw a **constraint** exception.

551

552 **maxValue** Integer

553 The maximum value allowed for this property.

554 If an application tries to set the value of this property to a value higher than `maxValue`, the
555 repository MUST throw a **constraint** exception.

556

557 2.1.3.3.4 Attributes specific to DateTime Object-Type Property Definitions

558 The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is
559 | `"DateTime"`, in addition to the common attributes specified above. A repository MAY provide additional
560 guidance on what values can be accepted. If the following attributes are not present the repository
561 behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

562 **resolution** String Enumeration

563 This is the precision in bits supported for values of this property. Valid values for this attribute are:

- 564 • Year: Year resolution is persisted
- 565 • Date: Date resolution is persisted
- 566 • Time: Time resolution is persisted

567

568 2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions

569 The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is
570 | `"Decimal"`, in addition to the common attributes specified above. A repository MAY provide additional
571 guidance on what values can be accepted. If the following attributes are not present the repository
572 behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

573 **precision** Integer Enumeration

574 This is the precision in bits supported for values of this property. Valid values for this attribute are:

- 575 • 32: 32-bit precision (`"single"` as specified in IEEE-754-1985).
- 576 • 64: 64-bit precision (`"double"` as specified in IEEE-754-1985).

577

578 **minValue** Decimal

579 The minimum value allowed for this property.

580 If an application tries to set the value of this property to a value lower than `minValue`, the
581 repository MUST throw a **constraint** exception.

582

583 **maxValue** Decimal

584 The maximum value allowed for this property.

585 If an application tries to set the value of this property to a value higher than `maxValue`, the
586 repository MUST throw a **constraint** exception.

587 2.1.3.3.6 Attributes specific to String Object-Type Property Definitions

588 The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is
589 | `"String"`, in addition to the common attributes specified above. A repository MAY provide additional
590 guidance on what values can be accepted. If the following attributes are not present the repository
591 behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

592 **maxLength** Integer
593 The maximum length (in characters) allowed for a value of this property.
594 If an application attempts to set the value of this property to a string larger than the specified
595 maximum length, the repository MUST throw a **constraint** exception.

596 **2.1.4 Document Object**

597 Document objects are the elementary information entities managed by the repository.

598 Depending on its Object-type definition, a Document Object may be:

- 599 • **Version-able:** Can be acted upon via the Versioning Services (for example: *checkOut*, *checkIn*).
- 600 • **File-able:** Can be filed in zero, one, or more than one folder via the Multi-filing services.
- 601 • **Query-able:** Can be located via the Discovery Services (*query*).
- 602 • **Controllable-Policy:** Can have Policies applied to it (see section 2.1.7 Policy Object.)
- 603 • **Controllable-ACL:** Can have an ACL applied to it (see section 2.8 Access Control)

604 Additionally, whether a Document object MUST, MAY or MUST NOT have a content-stream is specified
605 in its object-type definition. A Document Object MAY be associated with zero or more renditions.

606 Note: When a document is versioned, each version of the document is a separate document object. Thus,
607 for document objects, an object ID actually identifies a specific version of a document.

608 **2.1.4.1 Content Stream**

609 A content-stream is a binary stream. Its maximum length is repository-specific. Each content-stream has
610 a **MIME Media Type**, as defined by RFC2045 and RFC2046. A content-stream's attributes are
611 represented as properties of the content-stream containing document object. There is no MIME-type-
612 specific attribute or name directly associated with the content-stream outside of the document object.

613 CMIS provides basic CRUD services for content-stream, using the ID of a content-stream containing
614 document object for identification. A content stream also has a `streamId` which is used for access to the
615 stream. The `Set Content-Stream` service (*setContentStream*) either creates a new content-stream for a
616 document object or replaces an existing content-stream. The `Get Content-Stream` service
617 (*getContentStream*) retrieves a content-stream. The `Delete Content-Stream` service
618 (*deleteContentStream*) deletes a content-stream from a document object. In addition, the
619 `Create Document` and `Check-in` services MAY also take a content-stream as an optional input. A
620 content stream MUST be specified if required by the type definition. These are the only services that
621 operate on content-stream. The `Get Properties` and `Query` services, for example, do not return a
622 content-stream.

623 `Set Content-Stream` and `Delete Content-Stream` services are considered modifications to a
624 content-stream containing document object, and SHOULD therefore change the object's
625 `LastModificationDate` property upon successful completion.

626 The ability to set or delete a content stream is controlled by the
627 `capabilityContentStreamUpdatability` capability.

628 2.1.4.2 Renditions

629 Some ECM repositories provide a facility to retrieve alternative representations of a document. These
630 alternative representations are known as renditions. This could apply to a preview case which would
631 enable the client to preview the content of a document without needing to download the full content.
632 Previews are generally reduced fidelity representations such as thumbnails. Renditions can take on any
633 general form, such as a PDF version of a word document.

634 A CMIS repository MAY expose zero or more renditions for a document or folder in addition to a
635 document's content stream. CMIS provides no capability to create or update renditions accessed
636 through the rendition services. Renditions are specific to the version of the document or folder and may
637 differ between document versions. Each rendition consists of a set of rendition attributes and a rendition
638 stream. Rendition attributes are not object properties, and are not queryable. They can be retrieved using
639 the getRenditions service. A rendition stream can be retrieved using the getContentStream service with
640 the rendition's streamId parameter.

641 2.1.4.2.1 Rendition Attributes

642 A rendition has the following attributes:

643	streamId	ID
644		Identifies the rendition stream.
645		
646	mimeType	String
647		The MIME type of the rendition stream.
648		
649	length	Integer (optional)
650		The length of the rendition stream in bytes.
651		
652	title	String (optional)
653		Human readable information about the rendition.
654		
655	kind	String
656		A categorization String associated with the rendition.
657		
658	height	Integer (optional)
659		Typically used for "image" renditions (expressed as pixels). SHOULD be present if kind =
660		cmis:thumbnail.
661		
662	width	Integer (optional)
663		Typically used for "image" renditions (expressed as pixels). SHOULD be present if kind =
664		cmis:thumbnail.
665		
666	renditionDocumentId	ID (optional)
667		If specified, then the rendition can also be accessed as a document object in the CMIS services.
668		If not set, then the rendition can only be accessed via the rendition services. Referential integrity
669		of this ID is repository-specific.

670 2.1.4.2.2 Rendition Kind

671 A Rendition may be categorized via its `kind`. The repository is responsible for assigning kinds to
672 Renditions, including custom kinds. A repository kind does not necessarily identify a single Rendition for
673 a given Object.

674 CMIS defines the following kind:

- 675 • `cmis:thumbnail`: A rendition whose purpose is to provide an image preview of the document
676 without requiring the client to download the full document content stream. Thumbnails are
677 generally reduced fidelity representations.

678 2.1.4.3 Document Object-Type Definition

679 This section describes the definition of the Document Object-Type's attribute values and property
680 definitions which must be present on Document instance objects. All attributes and property definitions
681 are listed by their ID.

682 2.1.4.3.1 Attributes specific to Document Object-Types

683 The following Object **attributes** MUST only apply to Object-Type definitions whose baseId is the
684 `cmis:document` Object-Type, in addition to the common attributes specified above:

685 **versionable** Boolean

686 Indicates whether or not objects of this type are version-able. (See section 0

687 [Versioning](#)

688 [Versioning](#).)

689

690 **contentStreamAllowed** Enum

691 A value that indicates whether a content-stream MAY, MUST, or MUST NOT be included in
692 objects of this type. Values:

- 693 • **notallowed**: A content-stream MUST NOT be included
- 694 • **allowed**: A content-stream MAY be included
- 695 • **required**: A content-stream MUST be included (i.e. MUST be included when the object is
696 created, and MUST NOT be deleted.)

697 2.1.4.3.2 Attribute Values

698 The Document Object-Type MUST have the following attribute values.

699 Notes:

- 700 • A value of `<repository-specific>` indicates that the value of the property MAY be set to any valid
701 value for the attribute type.
- 702 • Unless explicitly stated otherwise, all values specified in the list MUST be followed for the Object-
703 Type definition.

704

705 **id**

706 Value: `cmis:document`

707

708 **localName**

709 Value: `<repository-specific>`

710

711 **localNamespace**
712 Value: <repository-specific>
713
714 **queryName**
715 Value: cmis:document
716
717 **displayName**
718 Value: <repository-specific>
719
720 **baseId**
721 Value: cmis:document
722
723 **parentId**
724 Value: Not set
725
726 **description**
727 Value: <repository-specific>
728
729 **creatable**
730 Value: <repository-specific>
731
732 **fileable**
733 Value: TRUE
734
735 **queryable**
736 Value: SHOULD be TRUE
737
738 **controllablePolicy**
739 Value: <repository-specific>
740
741 **includedInSupertypeQuery**
742 Value: <repository-specific>
743
744 **versionable**
745 Value: <repository-specific>
746
747 **contentStreamAllowed**
748 Value: <repository-specific>
749
750 **controllableACL**
751 Value: <repository-specific>
752
753 **fulltextIndexed**

754 Value: <repository-specific>

755 2.1.4.3.3 Property Definitions

756 The Document base Object-Type MUST have the following property definitions, and MAY include
757 additional property definitions. Any attributes not specified for the property definition are repository
758 specific. For all property definitions on base types, the query name MUST be the same as the property
759 ID. The repository MUST have the following property definitions on the Document Type:

760

761 **cmis:name** Name of the object

762 Inherited: False

763 Property Type: String

764 Cardinality: Single

765

766 **cmis:objectId** Id of the object

767 Required: False

768 Inherited: False

769 Property Type: ID

770 Cardinality: Single

771 Updatability: Read Only

772 Choices: Not Applicable

773 Open Choice: Not Applicable

774 Repository MUST return this property with non-empty values when an object is requested and the
775 property filter does not exclude them

776

777

778 **cmis:baseTypeid** Id of the base object-type for the object

779 Required: False

780 Inherited: False

781 Property Type: ID

782 Cardinality: Single

783 Updatability: Read Only

784 Choices: Not Applicable

785 Open Choice: Not Applicable

786 Repository MUST return this property with non-empty values when an object is requested and the
787 property filter does not exclude them

788

789 **cmis:objectTypeid** Id of the object's type

790 Required: True

791 Inherited: False

792 Property Type: ID

793 Cardinality: Single

794 Updatability: oncreate

795 Choices: Not Applicable

796 Open Choice: Not Applicable

797 Repository MUST return this property with non-empty values when an object is requested and the
798 property filter does not exclude them

799

800 **cmis:createdBy** User who created the object.

801 Required: False

802 Inherited: False

803 Property Type: String

804 Cardinality: Single

805 Updatability: Read Only

806 Choices: Not Applicable

807 Open Choice: Not Applicable

808 Queryable: True

809 Orderable: True

810 Repository MUST return this property with non-empty values when an object is requested and the
811 property filter does not exclude them

812

813 **cmis:creationDate** DateTime when the object was created.

814 Required: False

815 Inherited: False

816 Property Type: DateTime

817 Cardinality: Single

818 Updatability: Read Only

819 Choices: Not Applicable

820 Open Choice: Not Applicable

821 Queryable: True

822 Orderable: True

823 Repository MUST return this property with non-empty values when an object is requested and the
824 property filter does not exclude them

825

826 **cmis:lastModifiedBy** User who last modified the object.

827 Required: False

828 Inherited: False

829 Property Type: String

830 Cardinality: Single

831 Updatability: Read Only

832 Choices: Not Applicable

833 Open Choice: Not Applicable

834 Queryable: True

835 Orderable: True

836 Repository MUST return this property with non-empty values when an object is requested and the
837 property filter does not exclude them

838

839 **cmis:lastModificationDate** DateTime when the object was last modified.

840 Required: False
 841 Inherited: False
 842 Property Type: DateTime
 843 Cardinality: Single
 844 Updatability: Read Only
 845 Choices: Not Applicable
 846 Open Choice: Not Applicable
 847 Queryable: True
 848 Orderable: True
 849 Repository MUST return this property with non-empty values when an object is requested and the
 850 property filter does not exclude them

851
 852 **cmis:changeToken** Opaque token used for optimistic locking & concurrency
 853 checking. (see section 2.2.1.3 Change Tokens)

854 Required: False
 855 Inherited: False
 856 Property Type: String
 857 Cardinality: Single
 858 Updatability: Read Only
 859 Choices: Not Applicable
 860 Open Choice: Not Applicable

861 ~~Repository MUST return this property with non-empty values when an object is requested and the~~
 862 ~~property filter does not exclude them~~ Repository MUST return this property with non-empty values
 863 when an object is requested and the property filter does not exclude them. If the repository does
 864 not support change tokens, this property SHOULD not be set.

865
 866 **cmis:isImmutable** TRUE if the repository MUST throw an error at any attempt to
 867 update or delete the object.

868 Required: False
 869 Inherited: False
 870 Property Type: Boolean
 871 Cardinality: Single
 872 Updatability: Read Only
 873 Choices: Not Applicable
 874 Open Choice: Not Applicable

875 Repository MUST return this property with non-empty values when an object is requested and the
 876 property filter does not exclude them

877
 878 **cmis:isLatestVersion** See section 0

879 Versioning
 880 ~~Versioning.~~

881 Required: False
 882 Inherited: False
 883 Property Type: Boolean

884 Cardinality: Single
 885 Updatability: Read Only
 886 Choices: Not Applicable
 887 Open Choice: Not Applicable
 888 Repository MUST return this property with non-empty values when an object is requested and the
 889 property filter does not exclude them. Version Property Values are repository-specific when a
 890 document is defined as non-versionable.

891

892 **cmis:isMajorVersion** See section 0

893 Versioning

894 ~~Versioning.~~

895 Required: False
 896 Inherited: False
 897 Property Type: Boolean
 898 Cardinality: Single
 899 Updatability: Read Only
 900 Choices: Not Applicable
 901 Open Choice: Not Applicable

902 Repository MUST return this property with non-empty values when an object is requested and the
 903 property filter does not exclude them. Version Property Values are repository-specific when a
 904 document is defined as non-versionable.

905

906 **cmis:isLatestMajorVersion** See section 0

907 Versioning

908 ~~Versioning.~~

909 Required: False
 910 Inherited: False
 911 Property Type: Boolean
 912 Cardinality: Single
 913 Updatability: Read Only
 914 Choices: Not Applicable
 915 Open Choice: Not Applicable

916 Repository MUST return this property with non-empty values when an object is requested and the
 917 property filter does not exclude them. Version Property Values are repository-specific when a
 918 document is defined as non-versionable.

919

920 **cmis:versionLabel** See section 0

921 Versioning

922 ~~Versioning.~~

923 Required: False
 924 Inherited: False
 925 Property Type: String
 926 Cardinality: Single
 927 Updatability: Read Only

928 Choices: Not Applicable
 929 Open Choice: Not Applicable
 930 Repository MUST return this property with non-empty values when an object is requested and the
 931 property filter does not exclude them. Version Property Values are repository-specific when a
 932 document is defined as non-versionable.

933
 934 **cmis:versionSeriesId** See section 0

935 [Versioning](#)
 936 ~~Versioning.~~

937 Required: False
 938 Inherited: False
 939 Property Type: ID
 940 Cardinality: Single
 941 Updatability: Read Only
 942 Choices: Not Applicable
 943 Open Choice: Not Applicable

944 Repository MUST return this property with non-empty values when an object is requested and the
 945 property filter does not exclude them. Version Property Values are repository-specific when a
 946 document is defined as non-versionable.

947
 948 **cmis:isVersionSeriesCheckedOut** See section 0

949 [Versioning](#)
 950 ~~Versioning.~~

951 Required: False
 952 Inherited: False
 953 Property Type: Boolean
 954 Cardinality: Single
 955 Updatability: Read Only
 956 Choices: Not Applicable
 957 Open Choice: Not Applicable

958 Repository MUST return this property with non-empty values when an object is requested and the
 959 property filter does not exclude them. Version Property Values are repository-specific when a
 960 document is defined as non-versionable.

961
 962 **cmis:versionSeriesCheckedOutBy** See section 0

963 [Versioning](#)
 964 ~~Versioning.~~

965 Required: False
 966 Inherited: False
 967 Property Type: String
 968 Cardinality: Single
 969 Updatability: Read Only
 970 Choices: Not Applicable
 971 Open Choice: Not Applicable

972 Version Property Values are repository-specific when a document is defined as non-versionable.

973

974 **cmis:versionSeriesCheckedOutId** See section 0

975 [Versioning](#)

976 ~~Versioning.~~

977 Required: False

978 Inherited: False

979 Property Type: ID

980 Cardinality: Single

981 Updatability: Read Only

982 Choices: Not Applicable

983 Open Choice: Not Applicable

984 Version Property Values are repository-specific when a document is defined as non-versionable.

985

986 **cmis:checkinComment** See section 0

987 [Versioning](#)

988 ~~Versioning.~~

989 Required: False

990 Inherited: False

991 Property Type: String

992 Cardinality: Single

993 Updatability: Read Only

994 Choices: Not Applicable

995 Open Choice: Not Applicable

996 Version Property Values are repository-specific when a document is defined as non-versionable.

997

998 **cmis:contentStreamLength** Length of the content stream (in bytes).

999 Required: False

1000 Inherited: False

1001 Property Type: Integer

1002 Cardinality: Single

1003 Updatability: Read Only

1004 Choices: Not Applicable

1005 Open Choice: Not Applicable

1006 Repository MUST return this property with non-empty values when an object is requested and the
1007 property filter does not exclude them and if the document has a content stream

1008

1009 **cmis:contentStreamMimeType** MIME type of the Content Stream

1010 Required: False

1011 Inherited: False

1012 Property Type: String

1013 Cardinality: Single

1014 Updatability: Read Only

1015	Choices:	Not Applicable
1016	Open Choice:	Not Applicable
1017	Repository MUST return this property with non-empty values when an object is requested and the	
1018	property filter does not exclude them and if the document has a content stream	
1019		
1020	cmis:contentStreamFileName	File name of the Content Stream
1021	Required:	False
1022	Inherited:	False
1023	Property Type:	String
1024	Cardinality:	Single
1025	Repository MUST return this property with non-empty values when an object is requested and the	
1026	property filter does not exclude them and if the document has a content stream	
1027		
1028	cmis:contentStreamId	Id of the stream
1029	Required:	False
1030	Inherited:	False
1031	Property Type:	ID
1032	Cardinality:	Single
1033	Updatability:	Read Only
1034	Choices:	Not Applicable
1035	Open Choice:	Not Applicable

1036 2.1.5 Folder Object

1037 A folder object serves as the anchor for a collection of *file-able* objects. The folder object has an *implicit*
1038 hierarchical relationship with each object in its collection, with the anchor folder object being the **Parent**
1039 object and each object in the collection being a **Child** object. This implicit relationship has specific
1040 containment semantics which MUST be maintained by the repository with implicit referential integrity.
1041 (That is, there will never be a dangling parent-relationship or a dangling child-relationship. Furthermore,
1042 object A is a parent of object B if and only if object B is a child of object A.) This system-maintained
1043 implicit relationship is distinct from an *explicit* relationship which is instantiated by an application-
1044 maintained Relationship Object. (See section 2.1.6 Relationship Object.)

1045 A folder object does not have a content-stream and is not version-able. A folder object MAY be
1046 associated with zero or more renditions (see section 2.1.4.2 Renditions).

1047 2.1.5.1 File-able Objects

1048 A *file-able* object is one that MAY be "filed" into a folder. That is, it MAY be a child object of a folder
1049 object. The following list defines whether the base CMIS Object-types are file-able:

1050	cmis:folder
1051	MUST be file-able
1052	
1053	cmis:document
1054	MUST be file-able
1055	
1056	cmis:relationship

1057 MUST NOT be file-able

1058

1059 **cmis:policy**

1060 MAY be file-able

1061 2.1.5.1.1 Document Version Series and Filing

1062 | Since document objects are versionable, a document object's membership in a folder MAY be version-
1063 specific or version-independent. That is, the folder membership MAY be restricted to that particular
1064 version of the document or MAY apply to all versions of the document. Whether or not a repository
1065 supports version-specific filing is discoverable via the `Get Repository Information` service
1066 (`getRepositoryInfo`).

1067 When the child objects of a folder are retrieved, a specific version of a document MAY be returned. If the
1068 repository supports version-specific filing, the specific version filed in that folder is returned. If the
1069 repository does not support version-specific filing, the latest version of the document is returned.

1070 Likewise, this version sensitivity in child-binding also affects the behavior of parent retrieval for a
1071 document object, as well as the scope of the `IN_FOLDER()` and `IN_TREE()` function calls in a query. For
1072 non-versionable fileable objects, their membership in a folder does not have version sensitivity.

1073 2.1.5.1.2 Filing Restrictions by Object-Type

1074 | A folder collection's membership MAY be restricted by object-type. Each folder object has a multi-valued
1075 `AllowedChildObjectTypes` property, which specifies that only objects of these types are allowed to be
1076 its children. If this property is `not set`, then objects of any file-able type MAY be filed in the Folder. It is
1077 repository-specific if subtypes of the types listed in the `AllowedChildObjectTypes` property MAY be filed
1078 in the folder.

1079 Because of these filing constraints, when a new folder object is created, an existing folder object MUST
1080 be specified as its parent.

1081 When a non-file-able object is created, a parent folder MUST NOT be specified.

1082 When a file-able object is deleted, it is removed from any folder collection in which the object is a
1083 member. In other words, when an object is deleted, all implicit parent-child relationships with the deleted
1084 object as a child cease to exist.

1085 2.1.5.2 Folder Hierarchy

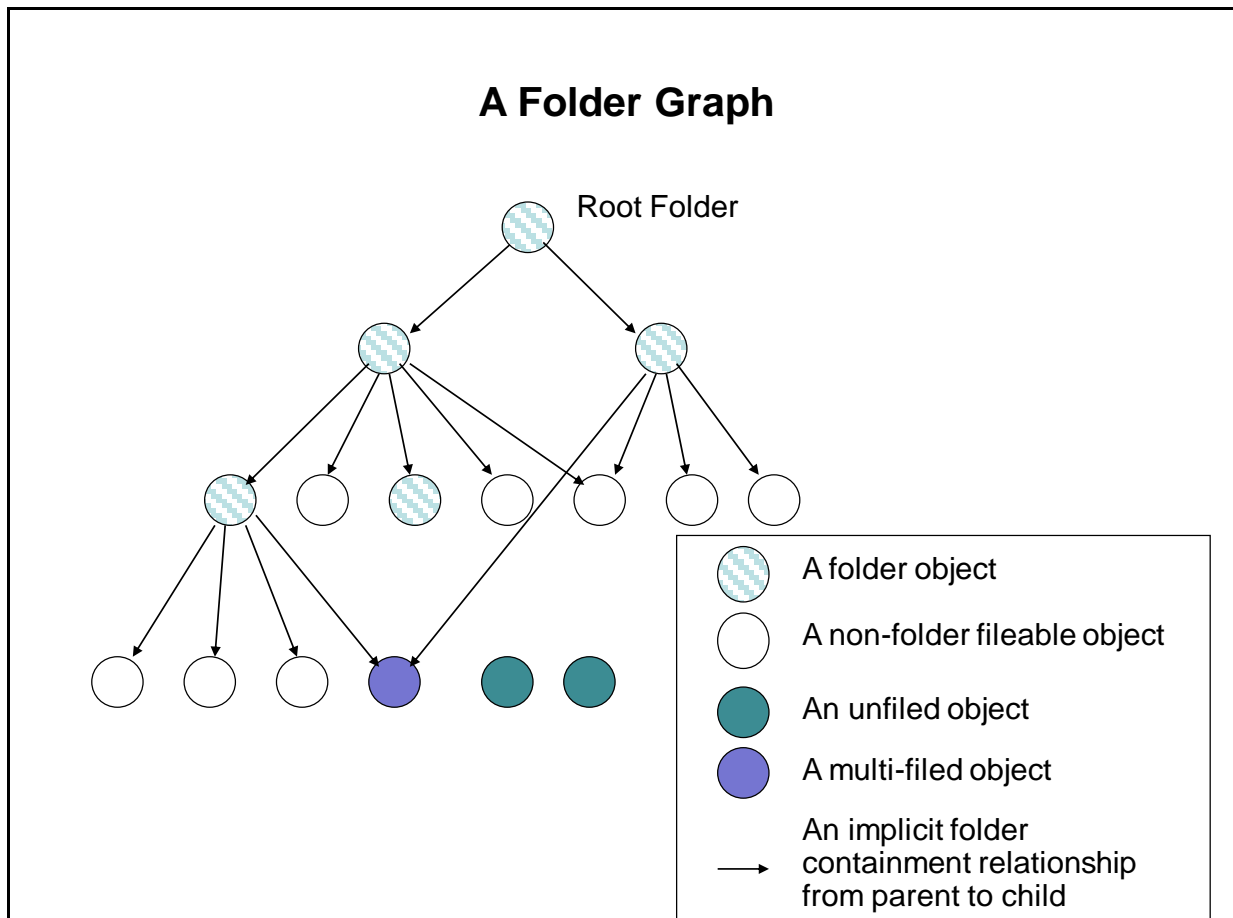
1086 CMIS imposes the following constraints on folder objects:

- 1087 • Every folder object, except for one which is called the **Root Folder**, MUST have one and only
1088 one parent folder. The Root Folder does not have a parent.
- 1089 • A cycle in folder containment relationships is not allowed. That is, a folder object cannot have
1090 itself as one of its descendant objects.
- 1091 • A child object that is a folder object can itself be the parent object of other file-able objects.

1092 With these constraints, the folder objects in a CMIS repository necessarily form a strict hierarchy, with the
1093 Root Folder being the root of the hierarchy.

1094 The child objects of a given folder object, their child objects, and grandchild objects, etc., are called
1095 **Descendant** objects of the given folder object. A folder object together with all its descendant objects are
1096 collectively called a **Tree** rooted at that folder object.

1097 A non-folder object does not have any descendant object. Thus, a **Folder Graph** that consists of all
 1098 fileable objects as nodes, and all the implicit folder containment relationships as directed edges from
 1099 parent to child, is a directed acyclic graph, possibly with some disconnected (orphan) nodes. It follows
 1100 that the tree rooted at any given folder object is also a directed acyclic graph, although a non-folder object
 1101 in the tree MAY have ancestors that are not ancestors of the rooted folder.



1102

1103 Folder objects are handled using [the basic CRUD services for objects](#), and the folder graph is traversed
 1104 using the [Navigation Services](#).

1105 The **Root Folder** is a special folder such that it cannot be created, deleted, or moved using CMIS
 1106 services. Otherwise, it behaves like any other folder object.

1107 2.1.5.3 Paths

1108 A folder hierarchy MAY be represented in a canonical notation such as path. For CMIS, a path is
 1109 represented by:

- 1110 • ["/](#) for the root folder
- 1111 • All paths start with the root folder.
- 1112 • A set of the folder and object path segments separated by ["/](#) in order of closest to the root.
- 1113 • Folder and object path segments are specified by pathSegment tokens which can be retrieved by
 1114 all services that take an `includePathSegments` parameter.
- 1115 • A pathSegment token MUST not include a ["/](#) character.

- 1116 ○ It is repository specific how a repository chooses the value for pathSegment.
1117 Repositories might choose to use cmis:name or content stream filename for
1118 pathSegment token.
- 1119 • The pathSegment token for each item MUST uniquely identify the item in the folder.

1120

1121 E.g., if folder A is under the root, and folder B is under A, then the path would be /A/B.

1122 A path for an object may be calculated by taking the item's parent folder cmis:path property and
1123 appending the "/" character and the object's pathSegment. This constructed path may be given as
1124 input to the *getObjectByPath* service for object by path retrieval.

1125 The *getObjectParents* service returns *relativePathSegment* tokens. These tokens are the
1126 pathSegment of the input object relative to the parent folders.

1127 **2.1.5.4 Folder Object-Type Definition**

1128 This section describes the definition of the Folder Object-Type's attribute values and property definitions
1129 which must be present on Folder instance objects. All attributes and property definitions are listed by
1130 their ID.

1131 **2.1.5.4.1 Attribute Values**

1132 The Folder Object-Type MUST have the following attribute values.

1133 Notes:

- 1134 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid
1135 value for the attribute type.
- 1136 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1137 Object-Type definition.

1138

1139 **id**

Value: cmis:folder

1141

1142 **localName**

Value: <repository-specific>

1144

1145 **localNamespace**

Value: <repository-specific>

1147

1148 **queryName**

Value: cmis:folder

1150

1151 **displayName**

Value: <repository-specific>

1153

1154 **baseId**

Value: cmis:folder

1156

1157 **parentId**

Value: Not set

1158

1159
 1160 **description**
 1161 Value: <repository-specific>
 1162
 1163 **creatable**
 1164 Value: <repository-specific>
 1165
 1166 **fileable**
 1167 Value: TRUE
 1168
 1169 **queryable**
 1170 Value: SHOULD be TRUE
 1171
 1172 **controllablePolicy**
 1173 Value: <repository-specific>
 1174
 1175 **includedInSupertypeQuery**
 1176 Value: <repository-specific>
 1177
 1178 **controllableACL**
 1179 Value: <repository-specific>
 1180
 1181 **fulltextIndexed**
 1182 Value: <repository-specific>
 1183

1184 **2.1.5.4.2 Property Definitions**

1185 The Folder base Object-Type MUST have the following property definitions, and MAY include additional
 1186 property definitions. Any attributes not specified for the Property Definition are repository specific. For all
 1187 property definitions on base types, the query name MUST be the same as the property ID. The
 1188 repository MUST have the following property definitions on the Folder Type:

1189	cmis:name	Name of the object
1190	Inherited:	False
1191	Property Type:	String
1192	Cardinality:	Single
1193	Required:	True
1194		
1195	cmis:objectId	Id of the object
1196	Required:	False
1197	Inherited:	False
1198	Property Type:	ID
1199	Cardinality:	Single
1200	Updatability:	Read Only
1201	Choices:	Not Applicable

1202	Open Choice:	Not Applicable
1203	Repository MUST return this property with non-empty values when an object is requested and the	
1204	property filter does not exclude them	
1205		
1206	cmis:baseTypeid	Id of the base object-type for the object
1207	Required:	False
1208	Inherited:	False
1209	Property Type:	ID
1210	Cardinality:	Single
1211	Updatability:	Read Only
1212	Choices:	Not Applicable
1213	Open Choice:	Not Applicable
1214	Repository MUST return this property with non-empty values when an object is requested and the	
1215	property filter does not exclude them	
1216		
1217	cmis:objectTypeid	Id of the object's type
1218	Required:	False <u>True</u>
1219	Inherited:	False
1220	Property Type:	ID
1221	Cardinality:	Single
1222	Updatability:	oncreate
1223	Choices:	Not Applicable
1224	Open Choice:	Not Applicable
1225	Repository MUST return this property with non-empty values when an object is requested and the	
1226	property filter does not exclude them	
1227		
1228	cmis:createdBy	User who created the object.
1229	Required:	False
1230	Inherited:	False
1231	Property Type:	String
1232	Cardinality:	Single
1233	Updatability:	Read Only
1234	Choices:	Not Applicable
1235	Open Choice:	Not Applicable
1236	Queryable:	True
1237	Orderable:	True
1238	Repository MUST return this property with non-empty values when an object is requested and the	
1239	property filter does not exclude them	
1240		
1241	cmis:creationDate	DateTime when the object was created.
1242	Required:	False
1243	Inherited:	False
1244	Property Type:	DateTime

1245	Cardinality:	Single
1246	Updatability:	Read Only
1247	Choices:	Not Applicable
1248	Open Choice:	Not Applicable
1249	Queryable:	True
1250	Orderable:	True
1251	Repository MUST return this property with non-empty values when an object is requested and the	
1252	property filter does not exclude them	
1253		
1254	cmis:lastModifiedBy	User who last modified the object.
1255	Required:	False
1256	Inherited:	False
1257	Property Type:	String
1258	Cardinality:	Single
1259	Updatability:	Read Only
1260	Choices:	Not Applicable
1261	Open Choice:	Not Applicable
1262	Queryable:	True
1263	Orderable:	True
1264	Repository MUST return this property with non-empty values when an object is requested and the	
1265	property filter does not exclude them	
1266		
1267	cmis:lastModificationDate	DateTime when the object was last modified.
1268	Required:	False
1269	Inherited:	False
1270	Property Type:	DateTime
1271	Cardinality:	Single
1272	Updatability:	Read Only
1273	Choices:	Not Applicable
1274	Open Choice:	Not Applicable
1275	Queryable:	True
1276	Orderable:	True
1277	MUST be set on the object	
1278		
1279	cmis:changeToken	Token used for optimistic locking & concurrency checking.
1280	(see section 2.2.1.3 Change Tokens)	
1281	Required:	False
1282	Inherited:	False
1283	Property Type:	String
1284	Cardinality:	Single
1285	Updatability:	Read Only
1286	Choices:	Not Applicable
1287	Open Choice:	Not Applicable

1288 ~~Repository MUST return this property with non-empty values when an object is requested and the~~
1289 ~~property filter does not exclude them~~ Repository MUST return this property with non-empty values
1290 when an object is requested and the property filter does not exclude them. If the repository does
1291 not support change tokens, this property SHOULD not be set.

1292
1293 **cmis:parentId** ID of the parent folder of the folder.
1294 Required: False
1295 Inherited: False
1296 Property Type: ID
1297 Cardinality: Single
1298 Updatability: Read Only
1299 Choices: Not Applicable
1300 Open Choice: Not Applicable
1301 Repository MUST return this property with non-empty values when an object is requested and the
1302 property filter does not exclude them

1303
1304 **cmis:path** The fully qualified path to this folder. See section 2.1.5.3
1305 Paths.
1306 Required: False
1307 Inherited: False
1308 Property Type: String
1309 Cardinality: Single
1310 Updatability: Read Only
1311 Choices: Not Applicable
1312 Open Choice: Not Applicable
1313 Repository MUST return this property with non-empty values when an object is requested and the
1314 property filter does not exclude them

1315
1316 **cmis:allowedChildObjectTypes** Id's of the set of Object-types that can be created, moved or
1317 filed into this folder.
1318 Required: False
1319 Inherited: False
1320 Property Type: ID
1321 Cardinality: Multi
1322 Updatability: Read Only
1323 Choices: Not Applicable
1324 Open Choice: Not Applicable

1325 2.1.6 Relationship Object

1326 A relationship object is semantically a *dependent* object. A relationship object MUST NOT have a
1327 content-stream, and MUST NOT be versionable, MAY be queryable, and MUST NOT be fileable,
1328 although it MAY be controllable.

1329 If a repository does not support relationship objects, the relationship base object-type SHOULD NOT be
1330 returned by a `GetTypes` service call.

1331 A **Relationship Object** instantiates an explicit, binary, directional, non-invasive, and typed relationship
1332 between a **Source Object** and a **Target Object**. The source object and the target object MUST both be
1333 independent objects, such as a document object, a folder object, or a policy object. Whether a policy
1334 object is allowed to be the source or target object of a relationship object is repository-specific.

1335 The relationship instantiated by a relationship object is *explicit* since it is explicitly represented by an
1336 object and is explicitly managed by application.

1337 This relationship is *non-invasive* in the sense that creating or removing this relationship SHOULD NOT
1338 modify either the source or the target object. That is, it SHOULD NOT require an update capability (or
1339 permission) on either object; SHOULD NOT affect the versioning state of either object; and SHOULD
1340 NOT change their `"Last Modification Date"`.

1341 Explicit relationships can be used to create an arbitrary relationship graph among independent objects.
1342 Such a relationship graph is only structural in nature. No inheritance or transitive properties are attached
1343 to a relationship graph.

1344 The notion of a source object and a target object of a relationship is used solely to indicate the direction of
1345 the relationship. No semantics or implementation bias is implied by this terminology.

1346 The binding of a relationship object to a source document object or to a target document object MAY be
1347 either version-specific or version-independent. This version sensitivity is repository-specific, and is largely
1348 transparent to CMIS. An independent object MAY participate in any number of explicit relationships, as
1349 the source object for some and as the target object for others. Multiple relationships MAY exist between
1350 the same pair of source and target objects.

1351 Referential integrity, either between the source object and the target object, or between the relationship
1352 object and the source or target object, is repository-specific. Therefore, creating an explicit relationship
1353 between two objects MAY impose a constraint on any of the three objects, and removing a relationship or
1354 deleting either the source or the target object MAY be restricted by such a constraint. If the source or the
1355 target object of a relationship is deleted, the repository MAY automatically delete the relationship object.

1356 Like all CMIS objects, relationship objects are typed. Typing relationship allows them to be grouped,
1357 identified, and traversed by type id, and for properties to be defined for individual relationship types.

1358 Additionally, a relationship object-type MAY specify that only Objects of a specific Object-Type can
1359 participate as the source object or target object for relationship objects of that type. If no such constraints
1360 are specified, then an independent object of any type MAY be the source or the target of a relationship
1361 object of that type.

1362 When a relationship object is created, the source object ID and the target object ID MUST reference valid
1363 non-relationship CMIS objects.

1364 When a relationship object is retrieved, its source object or target object MAY no longer exist, since
1365 referential integrity MAY not be maintained by a repository.

1366 In addition to object CRUD services, a `"Get Relationships"` service (*getObjectRelationships*) may be
1367 used to return a set of relationship objects in which a given independent object is identified as the source
1368 or the target object, according to the binding semantics maintained by the repository (i.e., either a
1369 version-specific or a version-independent binding as described above).

1370 **2.1.6.1 Relationship Object-Type Definition**

1371 This section describes the definition of the Relationship Object-Type's attribute values and property
1372 definitions which must be present on Relationship instance objects. All attributes and property definitions
1373 are listed by their ID.

1374 2.1.6.1.1 Attributes specific to Relationship Object-Types

1375 The following Object **attributes** MUST only apply to Object-Type definitions whose baseId is the
1376 cmis:relationship Object-Type, in addition to the common attributes specified above:

1377 **allowedSourceTypes** ID (multi-valued)

1378 A list of object-type IDs, indicating that the source object of a relationship object of this type
1379 MUST only be one of the types listed.

1380 If this attribute is `"not set"`, then the source object MAY be of any type.

1381

1382 **allowedTargetTypes** ID (multi-valued)

1383 A list of object-type IDs, indicating that the target object of a relationship object of this type MUST
1384 only be one of the types listed.

1385 If this attribute is `"not set"`, then the target object MAY be of any type.

1386 2.1.6.1.2 Attribute Values

1387 The Relationship Object-Type MUST have the following attribute values.

1388 Notes:

- 1389 • A value of `<repository-specific>` indicates that the value of the property MAY be set to any
1390 valid value for the attribute type.
- 1391 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1392 Object-Type definition.

1393

1394 **id**

1395 Value: cmis:relationship

1396

1397 **localName**

1398 Value: `<repository-specific>`

1399

1400 **localNamespace**

1401 Value: `<repository-specific>`

1402

1403 **queryName**

1404 Value: cmis:relationship

1405

1406 **displayName**

1407 Value: `<repository-specific>`

1408

1409 **baseId**

1410 Value: cmis:relationship

1411

1412 **parentId**

1413 Value: Not set

1414

1415 **description**

1416 Value: `<repository-specific>`

1417
1418 **creatable**
1419 Value: <repository-specific>
1420
1421 **fileable**
1422 Value: FALSE
1423
1424 **queryable**
1425 Value: <repository-specific>
1426
1427 **includedInSupertypeQuery**
1428 Value: <repository-specific>
1429
1430 **controllablePolicy**
1431 Value: <repository-specific>
1432
1433 **allowedSourceTypes**
1434 Value: <repository-specific>
1435
1436 **allowedTargetTypes**
1437 Value: <repository-specific>
1438
1439 **controllableACL**
1440 Value: <repository-specific>
1441
1442 **fulltextIndexed**
1443 Value: <repository-specific>
1444

1445 **2.1.6.1.3 Property Definitions**

1446 The Relationship base Object-Type MUST have the following property definitions, and MAY include
1447 additional property definitions. Any attributes not specified by the Property Definitions are repository
1448 specific. For all property definitions on base types, the query name MUST be the same as the property
1449 ID. The repository MUST have the following property definitions on the Relationship Type:

1450
1451 **cmis:name** Name of the object
1452 Inherited: False
1453 Property Type: String
1454 Cardinality: Single
1455
1456 **cmis:objectId** Id of the object
1457 Required: False
1458 Inherited: False
1459 Property Type: ID

1460	Cardinality:	Single
1461	Updatability:	Read Only
1462	Choices:	Not Applicable
1463	Open Choice:	Not Applicable
1464	Repository MUST return this property with non-empty values when an object is requested and the	
1465	property filter does not exclude them	
1466		
1467	cmis:baseTypeId	Id of the base object-type for the object
1468	Required:	False
1469	Inherited:	False
1470	Property Type:	ID
1471	Cardinality:	Single
1472	Updatability:	Read Only
1473	Choices:	Not Applicable
1474	Open Choice:	Not Applicable
1475	Repository MUST return this property with non-empty values when an object is requested and the	
1476	property filter does not exclude them	
1477		
1478	cmis:objectTypeid	Id of the object's type
1479	Required:	False True
1480	Inherited:	False
1481	Property Type:	ID
1482	Cardinality:	Single
1483	Updatability:	oncreate
1484	Choices:	Not Applicable
1485	Open Choice:	Not Applicable
1486	Repository MUST return this property with non-empty values when an object is requested and the	
1487	property filter does not exclude them	
1488		
1489	cmis:createdBy	User who created the object.
1490	Required:	False
1491	Inherited:	False
1492	Property Type:	String
1493	Cardinality:	Single
1494	Updatability:	Read Only
1495	Choices:	Not Applicable
1496	Open Choice:	Not Applicable
1497	Repository MUST return this property with non-empty values when an object is requested and the	
1498	property filter does not exclude them	
1499		
1500	cmis:creationDate	DateTime when the object was created.
1501	Required:	False
1502	Inherited:	False

1503 Property Type: DateTime
 1504 Cardinality: Single
 1505 Updatability: Read Only
 1506 Choices: Not Applicable
 1507 Open Choice: Not Applicable
 1508 Repository MUST return this property with non-empty values when an object is requested and the
 1509 property filter does not exclude them

1510
 1511 **cmis:lastModifiedBy** User who last modified the object.
 1512 Required: False
 1513 Inherited: False
 1514 Property Type: String
 1515 Cardinality: Single
 1516 Updatability: Read Only
 1517 Choices: Not Applicable
 1518 Open Choice: Not Applicable
 1519 Repository MUST return this property with non-empty values when an object is requested and the
 1520 property filter does not exclude them

1521
 1522 **cmis:lastModificationDate** DateTime when the object was last modified.
 1523 Required: False
 1524 Inherited: False
 1525 Property Type: DateTime
 1526 Cardinality: Single
 1527 Updatability: Read Only
 1528 Choices: Not Applicable
 1529 Open Choice: Not Applicable
 1530 Repository MUST return this property with non-empty values when an object is requested and the
 1531 property filter does not exclude them

1532
 1533 **cmis:changeToken** Opaque token used for optimistic locking & concurrency
 1534 checking. (see section 2.2.1.3 Change Tokens)
 1535 Required: False
 1536 Inherited: False
 1537 Property Type: String
 1538 Cardinality: Single
 1539 Updatability: Read Only
 1540 Choices: Not Applicable
 1541 Open Choice: Not Applicable
 1542 Repository MUST return this property with non-empty values when an object is requested and the
 1543 property filter does not exclude them. If the repository does not support change tokens, this
 1544 property SHOULD not be set.

1545
 1546 **cmis:sourced** ID of the source object of the relationship.

1547	Required:	True
1548	Inherited:	False
1549	Property Type:	ID
1550	Cardinality:	Single
1551	Choices:	Not Applicable
1552	Open Choice:	Not Applicable
1553		
1554	cmis:targetId	ID of the target object of the relationship.
1555	Required:	True
1556	Inherited:	False
1557	Property Type:	ID
1558	Cardinality:	Single
1559	Choices:	Not Applicable
1560	Open Choice:	Not Applicable

1561 2.1.7 Policy Object

1562 A policy object represents an administrative policy that can be enforced by a repository, such as a
 1563 retention management policy. CMIS 1.0 does not specify what kinds of administrative policies that are
 1564 specifically supported, nor attempts to model administrative policy of any particular kind. Only a base
 1565 object-type is specified for policy objects. Each policy object holds the text of an administrative policy as a
 1566 repository-specific string, which is opaque to CMIS and which may be used to support policies of various
 1567 kinds. A repository may create subtypes of this base type to support different kinds of administrative
 1568 policies more specifically. If a repository does not support policy objects, the policy base object-type
 1569 | SHOULD NOT be returned by a `“_Get Types”` service call. This is an extension point for repositories that
 1570 want to expose other capabilities via CMIS that are not supported directly in CMIS 1.0.

1571 Aside from allowing an application to create and maintain policy objects, CMIS allows an application to
 1572 | `“_apply”` a policy to an object, and to remove an applied policy from an object. An object to which a policy
 1573 may be applied is called a *controllable* object. A policy MAY be applied to multiple controllable objects.
 1574 Conversely, a repository MAY allow multiple policies applied to a controllable object. (A repository may,
 1575 for example, impose constraints such as only one policy of each kind can be applied to an object.)
 1576 | Whether or not an object is controllable is specified by the object’s type definition. Applying a policy to an
 1577 object is to place the object under the control of that policy (while the object may also be under the control
 1578 of other policies at the same time), and removing an applied policy from one of its controlled objects is to
 1579 remove the corresponding control from that object. This control may change the state of the object, may
 1580 impose certain constraints on service calls operating on this object, or may cause certain management
 1581 actions to take place. The effect of this control, when this effect takes place, and how this control interacts
 1582 with other controls, are repository-specific. Only directly/explicitly applied policies are covered by CMIS
 1583 1.0. Indirectly applying policy to an object, e.g. through inheritance, is outside the scope of CMIS 1.0.

1584 A policy object does not have a content-stream and is not versionable. It may be fileable, queryable or
 1585 controllable. Policy objects are handled using the basic CRUD services for objects. If a policy is updated,
 1586 the change may alter the corresponding control on objects that the policy is currently applied to. If a
 1587 controlled object is deleted, all the policies applied to that object, if there are any, are removed from that
 1588 object. A policy object that is currently applied to one or more controllable objects CAN NOT be deleted.
 1589 That is, there is an implicit referential constraint from a controlled object to its controlling policy object(s).
 1590 | Besides the basic CRUD services, the `“_Apply Policy”` (*applyPolicy*) and the `“_Remove Policy”`
 1591 (*removePolicy*) services may be used to apply a policy object to a controllable object and respectively to
 1592 | remove an applied policy from one of its controlled objects. In addition, the `“_Get Applied Policies”`
 1593 (*getAppliedPolicies*) service may be used to obtain the policy objects that are currently applied to a
 1594 controllable object.


1595 2.1.7.1 Policy Object-Type Definition


1596 | This section describes the definition of the Policy Object-Type's attribute values and property definitions
1597 | which must be present on Policy instance objects. All attributes and property definitions are listed by their
1598 | ID.

1599 2.1.7.1.1 Attribute Values

1600 The Policy Object-Type MUST have the following attribute values.

1601 Notes:

1602 |  A value of <repository-specific> indicates that the value of the property MAY be set to
1603 | any valid value for the attribute type.

1604 |  Unless explicitly stated otherwise, all values specified in the table MUST be followed for
1605 | the Object-Type definition.

1606

1607 **id**

1608 Value: cmis:policy

1609

1610 **localName**

1611 Value: <repository-specific>

1612

1613 **localNamespace**

1614 Value: <repository-specific>

1615

1616 **queryName**

1617 Value: cmis:policy

1618

1619 **displayName**

1620 Value: <repository-specific>

1621

1622 **baseId**

1623 Value: cmis:policy

1624

1625 **parentId**

1626 Value: Not set

1627

1628 **description**

1629 Value: <repository-specific>

1630

1631 **creatable**

1632 Value: <repository-specific>

1633

1634 **fileable**

1635 Value: <repository-specific>

1636

1637 **queryable**

1638 Value: <repository-specific>
 1639
 1640 **includedInSupertypeQuery**
 1641 Value: <repository-specific>
 1642
 1643 **controllablePolicy**
 1644 Value: <repository-specific>
 1645
 1646 **controllableACL**
 1647 Value: <repository-specific>
 1648
 1649 **fulltextIndexed**
 1650 Value: <repository-specific>
 1651

2.1.7.1.2 Property Definitions

1652 The Policy base Object-Type MUST have the following property definitions, and MAY include additional
 1653 property definitions. Any attributes not specified by the Property Definitions are repository specific. For
 1654 all property definitions on base types, the query name MUST be the same as the property ID. The
 1655 repository MUST have the following property definitions on the Policy Type:
 1656

1657
 1658 **cmis:name** Name of the object
 1659 Inherited: False
 1660 Property Type: String
 1661 Cardinality: Single
 1662

1663 **cmis:objectId** Id of the object
 1664 Required: False
 1665 Inherited: False
 1666 Property Type: ID
 1667 Cardinality: Single
 1668 Updatability: Read Only
 1669 Choices: Not Applicable
 1670 Open Choice: Not Applicable
 1671

1672 **cmis:baseTypeId** Id of the base object-type for the object
 1673 Required: False
 1674 Inherited: False
 1675 Property Type: ID
 1676 Cardinality: Single
 1677 Updatability: Read Only
 1678 Choices: Not Applicable
 1679 Open Choice: Not Applicable
 1680

1681	cmis:objectType	Id of the object's type
1682	Required:	False <u>True</u>
1683	Inherited:	False
1684	Property Type:	ID
1685	Cardinality:	Single
1686	Updatability:	oncreate
1687	Choices:	Not Applicable
1688	Open Choice:	Not Applicable
1689		
1690	cmis:createdBy	User who created the object.
1691	Required:	False
1692	Inherited:	False
1693	Property Type:	String
1694	Cardinality:	Single
1695	Updatability:	Read Only
1696	Choices:	Not Applicable
1697	Open Choice:	Not Applicable
1698		
1699	cmis:creationDate	DateTime when the object was created.
1700	Required:	False
1701	Inherited:	False
1702	Property Type:	DateTime
1703	Cardinality:	Single
1704	Updatability:	Read Only
1705	Choices:	Not Applicable
1706	Open Choice:	Not Applicable
1707		
1708	cmis:lastModifiedBy	User who last modified the object.
1709	Required:	False
1710	Inherited:	False
1711	Property Type:	String
1712	Cardinality:	Single
1713	Updatability:	Read Only
1714	Choices:	Not Applicable
1715	Open Choice:	Not Applicable
1716		
1717	cmis:lastModificationDate	DateTime when the object was last modified.
1718	Required:	False
1719	Inherited:	False
1720	Property Type:	DateTime
1721	Cardinality:	Single
1722	Updatability:	Read Only

1723	Choices:	Not Applicable
1724	Open Choice:	Not Applicable
1725		
1726	cmis:changeToken	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1727		
1728	Required:	False
1729	Inherited:	False
1730	Property Type:	String
1731	Cardinality:	Single
1732	Updatability:	Read Only
1733	Choices:	Not Applicable
1734	Open Choice:	Not Applicable
1735		<u>Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them. If the repository does not support change tokens, this property SHOULD not be set.</u>
1736		
1737		
1738		
1739	cmis:policyText	User-friendly description of the policy
1740	Required:	True
1741	Inherited:	False
1742	Property Type:	String
1743	Cardinality:	Single
1744	Choices:	Not Applicable
1745	Open Choice:	Not Applicable

1746 2.1.8 Access Control

1747 A repository can support either a base set of CMIS-defined permissions and/or its own set of repository
1748 specific permissions.

1749 The getACL service allows the requestor to specify that the result be expressed using only the CMIS
1750 defined permissions. Without this restriction, the response may include, or be solely expressed in
1751 repository specific permissions. The applyACL service permits either CMIS permissions or repository
1752 permissions, or a combination of both, to be used.

1753 2.1.8.1 ACL, ACE, Principal, and Permission

1754 An **ACL** is a list of **Access Control Entries** (ACEs) and MAY hold zero or more ACEs. If an ACL has no
1755 ACEs, the behavior is the same as if the ACL is not set.

1756 An **ACE** holds:

- 1757 • one **Principal**: A principal represents a user management object, e.g. a user, group, or role.
1758 It holds one **String** with the **principalid**.
- 1759 • One or more **Strings** with the names of the **permissions**.
- 1760 • a **Boolean** flag **direct**, which indicates if TRUE the ACE is directly assigned to the object. If
1761 FALSE, that the ACE is somehow derived.

1762 2.1.8.2 CMIS Permissions

1763 There are three basic permissions predefined by CMIS:

- 1764 | • **cmis:read**: to be used to express “permission to read”. A Repository SHOULD express
1765 | the permission for reading properties AND reading content with this permission.
- 1766 | • **cmis:write**: to be used to express “permission to write”. SHOULD be used to express
1767 | permission to write properties and content of an object. MAY include other basic CMIS
1768 | permissions.
- 1769 | • **cmis:all**: SHOULD be used to express all the permissions of a repository. SHOULD
1770 | include all other basic CMIS permissions.

1771 | How these basic permissions can be mapped to the allowable actions is repository specific. However, the
1772 | actual repository semantics for the basic permissions with regard to allowable actions can be discovered
1773 | by the *mappings* parameter returned by *getRepositoryInfo* (see below).

1774 | Repositories MAY extend this set with repository-specific permissions.

1775 | 2.1.8.3 ACL Capabilities

1776 | Whether a repository supports ACLs at all, may be discovered via *capabilityACL* returned by
1777 | *getRepositoryInfo* (see section 2.1.1.1 Optional Capabilities). If *capabilityACL* is *none*, ACLs are not
1778 | supported by the repository.

1779 | If *capabilityACL* is *discover* or *manage*, additional information about the repositories permission model
1780 | and how changes to ACL are handled, can be discovered via the *getRepositoryInfo* service:

- 1781 | • ~~<Array>~~ **Enum propagation**: specifies, how non-direct ACEs can be handled by the
1782 | repository using the following values (see section 2.2.10.2 applyACL):
 - 1783 | ○ **objectonly** indicates, that the repository is able to apply ACEs to a document or folder,
1784 | without changing the ACLs of other objects.
 - 1785 | ○ **propagate**: indicates that the ACEs is to be applied to the given object and all inheriting
1786 | objects. *Propagate incorporates the support for objectonly.*
 - 1787 | ○ **repositorydetermined** indicates, that the repository has its own mechanism of
1788 | computing how changing an ACL for an object influences the non-direct ACEs of other
1789 | objects.
- 1790 | • **<Array> PermissionDefinition repositoryPermissions**: is a list with names and
1791 | descriptions of the supported permissions.
- 1792 | • **<Array> PermissionMapping mappings**: contains a list with mappings for the basic CMIS
1793 | permissions to allowed actions.

1794 | 2.1.8.3.1 Supported Permissions

1795 | The list of permission definitions returned by *getRepositoryInfo* lists all the *permissions* a repository
1796 | supports. This list also includes the CMIS permissions if supported by the repository.

1797 | A *PermissionDefinition* holds:

- 1798 | • **String permission**: the (technical) name of the permission (unique within the list of permission
1799 | definitions).
- 1800 | • **(Optional) String description**: an optional description of the permission that should be used as
1801 | the permission's name to be presented to the user.

1802 | 2.1.8.3.2 AllowableActions & Permission Mapping

1803 | CMIS provides a mechanism called “AllowableActions” which allows an application to discover the set of
1804 | service operations that can currently be performed on a particular object, without having to actually invoke
1805 | the service.

1806 | The set of allowable actions on an object at a point in time are affected not only by CMIS ACLs, but also
1807 | by other factors such as:

1808 | 1. Constraints inherent in the CMIS Domain Model based on the object's base type or current
1809 | versioning state.

1810 | 2. Policies or other control mechanisms that are opaque to CMIS.

1811 |
1812 | CMIS defines several services that applications can use at run-time to discover the AllowableActions for
1813 | an object.

1814 | If a Repository supports ACLs, then the repository MUST provide a mapping table that defines how the
1815 | permissions supported by the repository interact with the CMIS allowable actions, i.e. which permissions
1816 | are necessary for a principal to have on one or more objects in order to potentially perform each action,
1817 | subject to the other constraints on allowable actions above.

1818 | This section defines both the allowable actions as well as how those actions are presented in the
1819 | PermissionMapping table.

1820 | The Permission Mapping table contains a set of (*key*, *permissions*) pairs:

1821 | o **String Key:** Because several allowable actions may require permissions on more than one
1822 | object – for example, moving a document from one folder to another may require permissions
1823 | on the document and each of the folders – the mapping table is defined in terms of
1824 | permission “_keys_”, where each key combines the name of the allowable action as the object
1825 | for which the principal needs the required permission.

1826 | o For example – the canMoveObject.Source key indicates the permissions that the
1827 | principal must have on the “_source folder_” to move an object from that folder into
1828 | another folder.

1829 | o **<Array> String permissions:** The names of one or more permissions that the principal MUST
1830 | have. If more than one permission is specified, then the principal MUST be allowed to
1831 | perform the operation if they have ANY of the listed permissions.

1832 | The list below defines all mapping keys, as well as a permissions mapping that repositories SHOULD
1833 | use. Repositories MAY require additional permissions.

1834 | For convenience, the list below groups all mapping entries by the underlying Allowable Actions, and
1835 | includes descriptive information. For each Allowable Action the following information is given:

1836 | Description: The description and name of the service the AllowableAction enables.

1837 | Base Object: The base object-types for which the allowable action MAY be TRUE.

1838 | Operand: The object the permission applies to.

1839 | Key: The permission mapping key.

1840 | Permissions: The permission values.

1841 |

1842 | **Navigation Services:**

1843 | **canGetDescendants**

1844 | Description: Can get the descendants of the folder (*getDescendants*)

1845 | Base Object: cmis:folder

1846 | Operand: cmis:folder

1847 | Key: canGetDescendants.Folder

1848 | Permission: Read

1849 |

1850 | **canGetFolderTree**

1851 | Description: Can get the sub-folder tree of the folder (*getFolderTree*)

1852 | Base Object: cmis:folder

1853 | Operand: cmis:folder

1854 | ~~Key: canGetFolderTree.Folder~~

1855 | ~~Permission: Read~~

1856

1857 **canGetChildren**

1858 Description: Can get the children of the folder (`getChildren`)

1859 Base Object: `cmis:folder`

1860 Operand: `cmis:folder`

1861 Key: `canGetChildren.Folder`

1862 Permission: Read

1863

1864 **canGetFolderParent**

1865 Description: Can get the parent/ancestor folder(s) of the folder (`getFolderParent`)

1866 Base Object: `cmis:folder`

1867 Operand: `cmis:folder`

1868 | Key: `canGetFolderParent.FolderObject`

1869 Permission: Read

1870

1871 **canGetObjectParents**

1872 Description: Can get the parent folders of the object. (`getObjectParents`)

1873 Base Object: `cmis:document`, `cmis:folder`, `cmis:policy`

1874 Operand: Object

1875 | Key: `canGetObjectParents.Object`

1876 Permission: Read

1877

1878 **Object Services:**

1879 **canCreateDocument**

1880 Description: Can create a `cmis:document` Object in the folder (`createDocument`)

1881 Base Object: `cmis:folder`

1882 Operand: Folder

1883 Key: `canCreateDocument.Folder`

1884 Permission: Read

1885

1886 **canCreateFolder**

1887 Description: Can create a `cmis:folder` Object as a child of the specified folder
1888 (`createFolder`)

1889 Base Object: `cmis:folder`

1890 Operand: Folder

1891 Key: `canCreateFolder.Folder`

1892 Permission: Read

1893

1894 **canCreateRelationship**

1895 Description: Can create a Relationship in which this Object is a source
1896 (`createRelationship`)

1897 Base Object: cmis:document, cmis:folder
1898 Operand: Object
1899 Key: canCreateRelationship.Source
1900 Permission: Read

1901
1902 **canCreateRelationship**

1903 Description: Can create a Relationship in which this Object is a target
1904 (createRelationship)
1905 Base Object: cmis:document, cmis:folder
1906 Operand: Object
1907 Key: canCreateRelationship.Target
1908 Permission: Read

1909
1910 **canGetProperties**

1911 Description: Can read the properties of this object (getProperties)
1912 Base Object: cmis:document, cmis:folder, cmis:relationship, cmis:policy
1913 Operand: Object
1914 Key: canGetProperties.Object
1915 Permission: Read

1916
1917 ~~**canGetRenditions**~~

1918 ~~Description: Can retrieve the renditions of this object (getRenditions)~~
1919 ~~Base Object: cmis:document, or cmis:folder~~
1920 ~~Operand: Object~~
1921 ~~Key: canGetRenditions.Object~~
1922 ~~Permission: Read~~

1923
1924 ~~**canGetContentStream**~~

1925 ~~Description: Can get the content stream for the Document object~~
1926 ~~(getContentStream)~~
1927 ~~Base Object: cmis:document~~
1928 ~~Operand: Object~~
1929 ~~Key: canGetContentStream.Object~~
1930 ~~Permission: Read~~

1931
1932 **canUpdateProperties**

1933 Description: Can update the properties of this object (updateProperties)
1934 Base Object: cmis:document, cmis:folder, cmis:relationship, cmis:policy
1935 Operand: Object
1936 Key: canUpdateProperties.Object
1937 Permission: Write

1938
1939 **canMoveObject**

1940 Description: Can move the object (`moveObject`)

1941 Base Object: `cmis:document, cmis:folder, cmis:policy`

1942 Operand: Object

1943 Key: `canMoveObject.Object`

1944 Permission: Write

1945

1946 **canMoveObject**

1947 Description: Can move an object into this folder (`moveObject`)

1948 Base Object: `cmis:folder`

1949 Operand: Folder

1950 Key: `canMoveObject.Target`

1951 Permission: Read

1952

1953 **canMoveObject**

1954 Description: Can move an object from this folder (`moveObject`)

1955 Base Object: `cmis:folder`

1956 Operand: Folder

1957 Key: `canMoveObject.Source`

1958 Permission: Read

1959

1960 **canDeleteObject**

1961 Description: Can delete this object (`deleteObject`)

1962 Base Object: `cmis:document, cmis:folder, cmis:relationship, cmis:policy`

1963 Operand: Object

1964 Key: `canDelete.Object`

1965 Permission: Write

1966

1967 ~~**canDeleteObject**~~

1968 ~~Description: Can delete an object that is a child of this folder (`deleteObject`)~~

1969 ~~Base Object: `cmis:folder`~~

1970 ~~Operand: Folder~~

1971 ~~Key: `canDelete.Folder`~~

1972 ~~Permission: Read~~

1973

1974 **canGetContentStream**

1975 Base Object: `cmis:document`

1976 Action: Can get the content stream for the Document object

1977 (`getContentStream`)

1978 Operand: Object

1979 Key: `canViewContent.Object`

1980 Permission: Read

1981

1982	canSetContentStream	
1983	Description:	Can set the content stream for the Document object
1984		(setContentStream)
1985	Base Object:	cmis:document
1986	Operand:	Object
1987	Key:	canSetContentStream.Document
1988	Permission:	Write
1989		
1990	canDeleteContentStream	
1991	Base Object:	cmis:document
1992	Action:	Can delete the content stream for the Document object
1993		(deleteContentStream)
1994	Operand:	Object
1995	Key:	canDeleteContentStream.Document
1996	Permission:	Write
1997		
1998	canDeleteTree	
1999	Base Object:	cmis:folder
2000	Action:	Can delete the folder and all contained objects (deleteTree)
2001	Operand:	Object
2002	Key:	canDeleteTree.Folder
2003	Permission:	Write
2004		
2005	Filing Services:	
2006	canAddObjectToFolder	
2007	Description:	Can file the document in a folder (addObjectToFolder)
2008	Base Object:	cmis:document, cmis:policy
2009	Operand:	Object
2010	Key:	canAddToFolder.Object
2011	Permission:	Read
2012		
2013	canAddObjectToFolder	
2014	Description:	Can file a document in the specified folder (addObjectToFolder)
2015	Base Object:	cmis:document, cmis:policy
2016	Operand:	Object
2017	Key:	canAddToFolder.Folder
2018	Permission:	Read
2019		
2020	canRemoveObjectFromFolder	
2021	Description:	Can unfile the specified document from a folder
2022		(removeObjectFromFolder)
2023	Base Object:	cmis:document, cmis:policy
2024	Operand:	Object

2025 | Key: canRemoveObjectFromFolder.Object
2026 | Permission: Read

2027

2028 **canRemoveObjectFromFolder**

2029 | Description: Can unfile a document from the specified folder
2030 | (removeObjectFromFolder)

2031 | Base Object: cmis:document, cmis:policy

2032 | Operand: Object

2033 | Key: canRemoveObjectFromFolder.Folder

2034 | Permission: Read

2035

2036 **Versioning Services:**

2037 **canCheckOut**

2038 | Description: Can check out the Document object (checkOut)

2039 | Base Object: cmis:document

2040 | Operand: Object

2041 | Key: ~~canCheckOut~~canCheckout.Document

2042 | Permission: Write

2043

2044 **canCancelCheckOut**

2045 | Description: Can cancel the check out the Document object (cancelCheckOut)

2046 | Base Object: cmis:document

2047 | Operand: Object

2048 | Key: canCancelCheckout.Document

2049 | Permission: Write

2050

2051 **canCheckIn**

2052 | Description: Can check in the Document object (checkIn)

2053 | Base Object: cmis:document

2054 | Operand: Object

2055 | Key: canCheckin.Document

2056 | Permission: Write

2057

2058 **canGetAllVersions**

2059 | Description: Can get the version series for the Document object (getAllVersions)

2060 | Base Object: cmis:document

2061 | Operand: Object

2062 | Key: canGetAllVersions.VersionSeriesDocument

2063 | Permission: Read

2064

2065 **Relationship Services:**

2066 **canGetObjectRelationships**

2067 Description: Can get the relationship in which this object is a source/target
2068 (getObjectRelationships)

2069 Base Object: cmis:document, cmis:folder, cmis:policy

2070 Operand: Object

2071 Key: canGetObjectRelationships.Object

2072 Permission: Read

2073

2074 Policy Services:

2075 **canApplyPolicy**

2076 Description: Can apply a policy to the Object (applyPolicy)

2077 Base Object: cmis:document, cmis:folder

2078 Operand: Object

2079 Key: canAddPolicy.Object

2080 Permission: Read

2081

2082 **canApplyPolicy**

2083 Description: Can apply the specified policy to an Object (applyPolicy)

2084 Base Object: cmis:policy

2085 Operand: Object

2086 Key: canAddPolicy.Policy

2087 Permission: Read

2088

2089 **canRemovePolicy**

2090 Description: Can remove a policy from the specified Object (removePolicy)

2091 Base Object: cmis:document, cmis:folder

2092 Operand: Object

2093 Key: canRemovePolicy.Object

2094 Permission: Read

2095

2096 **canRemovePolicy**

2097 Description: Can remove the specified policy from an Object (removePolicy)

2098 Base Object: cmis:document, cmis:folder

2099 Operand: cmis:policy

2100 Key: canRemovePolicy.Policy

2101 Permission: Read

2102

2103 **canGetAppliedPolicies**

2104 Description: Can get the list of Policies applied to the Object
2105 (getAppliedPolicies)

2106	Base Object:	cmis:document, cmis:folder
2107	Operand:	Object
2108	Key:	canGetAppliedPolicies.Object
2109	Permission:	Read
2110		
2111	ACL Services:	
2112	canGetACL	
2113	Description:	Can get ACL for object (<code>getACL</code>)
2114	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2115	Operand:	Object
2116	Key:	canGetACL.Object
2117	Permission:	Read
2118		
2119	canApplyACL	
2120	Description:	Can apply ACL to this object (<code>applyACL</code>)
2121	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2122	Operand:	Object
2123	Key:	canApplyACL.Object
2124	Permission:	Write
2125		

2126

2127 2.1.9 Versioning

2128 CMIS supports versioning of Document objects. Folder objects, relationship objects, and policy objects
2129 cannot be versioned.

2130 Whether or not a Document object is versionable (i.e. whether or not operations performed on the object
2131 | via the Versioning Services MUST be allowed) is specified by the `“_versionable”` attribute on its Object-
2132 type.

2133 | A **version** of a Document object is an explicit `“_deep”` copy of the object, preserving its state at a certain
2134 point in time. Each version of a Document object is itself a Document object, i.e. has its own *ObjectId*,
2135 property values, MAY be acted upon using all CMIS services that act upon Document objects, etc.

2136 2.1.9.1 Version Series

2137 A **version series** for a Document object is a transitively closed collection of all Document objects that
2138 have been created from an original Document in the Repository. Each version series has a unique,
2139 system-assigned, and immutable **version series ID**.

2140 The version series has transitive closure -- that is, if object B is a version of object A, and object C is a
2141 version of object B, then object C is also a version of object A. The objects in a version series can be
2142 conceptually sequenced by their respective *CreationDate* properties.

2143 Additionally, the repository MAY expose a textual **VersionLabel** that describes to a user the position of
2144 an individual object with respect to the version series. (For example, version 1.0).

2145 *Note:* A Document object that is NOT versionable will always have a single object in its Version Series. A
2146 versionable Document object MAY have one or more objects in its Version Series.

2147 2.1.9.2 Latest Version

2148 The version that has the most recent *LastModificationDate* is called the **Latest Version** of the series, or
2149 equivalently, the latest version of any Document object in the series.

2150 When the latest version of a version series is deleted, a previous version (if there is one) becomes the
2151 latest version.

2152 2.1.9.2.1 Behavioral constraints on non-Latest Versions

2153 Repositories NEED NOT allow the non-latest versions in a Version Series to be updated, queried, or
2154 searched.

2155 2.1.9.3 Major Versions

2156 A Document object in a Version Series MAY be designated as a **Major Version**.

2157 The CMIS specification does not define any semantic/behavioral differences between Major and non-
2158 Major versions in a Version Series. Repositories may enforce/apply additional constraints or semantics for
2159 Major versions, if the effect on CMIS services remains consistent with an allowable behavior of the CMIS
2160 model.

2161 If the Version Series contains one or more Major versions, the one that has the most recent
2162 *LastModificationDate* is the **Latest Major Version** of the version series.

2163 (Note that while a Version Series MUST always have a *Latest Version*, it NEED NOT have a *Latest Major*
2164 *Version*.)

2165 When the latest major version is deleted, a previous major version (if there is one) becomes the latest
2166 major version.

2167 2.1.9.4 Services that modify Version Series

2168 2.1.9.4.1 Checkout

2169 A new version of a versionable Document object is created when the *checkIn* service is invoked on the
2170 Private Working copy (PWC) of this object. A PWC is created by invoking *checkOut* on a versionable
2171 Document object. A repository MAY allow *any* Document object in a version series to be checked out, or
2172 MAY only allow the *Latest Version* to be checked out.

2173 The effects of invoking the *checkout* service MUST be as follows:

- 2174 • A new Document object, referred to herein as the **Private Working Copy (PWC)**, is created.
 - 2175 ○ The PWC NEED NOT be visible to users who have permissions to view other Document
2176 objects in the Version Series.
 - 2177 ○ Until it is checked in (using the *checkIn* service), the PWC MUST NOT be considered the
2178 *LatestMajorVersion* in the Version Series.
 - 2179 ○ The property values for the PWC SHOULD be identical to the properties of the Document
2180 object on which the *checkout* service was invoked. Certain properties such as `cmis:objectId`
2181 may be different. Properties such as `cmis:creationDate` most likely will be different. The
2182 content-stream of the PWC MAY be identical to the content-stream of the Document object
2183 on which the *checkout* service was invoked, or MAY be `"not set"`.

2184 After a successful *checkout* operation is completed, and until such time when the PWC is deleted (via the
2185 *cancelCheckOut* service) or checked-in (via the *checkIn*) service, the effects on other Documents in the
2186 Version Series MUST be as follows:

- 2187 • The repository MUST throw an exception if the *checkout* service is invoked on any Document in
2188 the Version Series. (I.e. there can only be one PWC for a version series at a time.)
- 2189 • The value of the `cmis:isVersionSeriesCheckedOut` property MUST be TRUE.
- 2190 • The value of the `cmis:versionSeriesCheckedOutBy` property MAY be set to a value indicating
2191 which user created the PWC. (The Repository MAY still show the `"not set"` value for this
2192 property.)
- 2193 • The value of the `cmis:versionSeriesCheckedOutId` property MAY be set to the Objectid of the
2194 PWC. (The Repository MAY still show the `"not set"` value for this property).
- 2195 • The repository MAY prevent operations that modify or delete the other Documents in the Version
2196 Series.

2197 2.1.9.4.2 Updates to the Private Working Copy

2198 If the repository supports the optional `"PWCUpdatable"` capability, then the repository MUST allow
2199 authorized users to modify the PWC Object using the Object services (e.g. *UpdateProperties*).

2200 If the repository does NOT support the `"PWCUpdatable"` capability, then the PWC object can only be
2201 modified as part of the *checkIn* service call.

2202 2.1.9.4.3 Discarding Check out

2203 An authorized user MAY discard the check-out using the *cancelCheckOut* service on any Document in
2204 the Version Series or by using the *deleteObject* service on the PWC Object. The effects of discarding a
2205 check-out MUST be as follows:

- 2206 • The PWC Object MUST be deleted.
- 2207 • For all other Documents in the Version Series:
 - 2208 ○ The value of the `cmis:isVersionSeriesCheckedOut` property MUST be FALSE.
 - 2209 ○ The value of the `cmis:versionSeriesCheckedOutBy` property MUST be `"not set"`.
 - 2210 ○ The value of the `cmis:versionSeriesCheckedOutId` property MUST be `"not set"`.
 - 2211 ○ The repository MUST allow authorized users to invoke the *checkout* service.

2212 2.1.9.4.4 Checkin

2213 | An authorized user/application MAY “check in” the Private Working Copy object via the *checkIn* service.
2214 | The *checkIn* service allows users/applications to provide update property values and a content-stream for
2215 | the PWC object.

2216 | The effects of the checkIn service MUST be as follows for successful checkins:

- 2217 | • The PWC object MUST be updated as specified by the inputs to the *checkIn* service. (Note that for
2218 | repositories that do NOT support the “PWCUpdatable” property, this is the only way to update the
2219 | PWC object.)
- 2220 | • The Document object resulting from the *checkIn* operation MUST be considered the *Latest Version*
2221 | in the Version Series.
- 2222 | • If the inputs to the *checkIn* service specified that the PWC MUST be a “major version”, then the
2223 | PWC MUST be considered the *Latest Major Version* in the Version Series.
- 2224 | • If the checkin returns a new cmis:objected, then the PWC object MUST disappear if the *checkIn* call
2225 | was successful and the new checked in version will use the new specified id.
- 2226 | • For all Documents in the Version Series:
 - 2227 | 1. The value of the cmis:isVersionSeriesCheckedOut property MUST be FALSE.
 - 2228 | 2. The value of the cmis:versionSeriesCheckedOutBy property MUST be “not set”.
 - 2229 | 3. The value of the cmis:versionSeriesCheckedOutId property MUST be “not set”.
 - 2230 | 4. The repository MUST allow authorized users to invoke the *checkout* service.

2231 | *Note:* The Repository MAY change the ID of the PWC upon completion of the *checkin* service invocation.

2232 | *Note:* A repository MAY automatically create new versions of Document objects without an explicit
2233 | invocation of the checkout/checkin services.

2234 2.1.9.5 Versioning Properties on Document Objects

2235 | All Document objects will have the following read-only property values pertaining to versioning:

2236		
2237	cmis:isLatestVersion	Boolean
2238	TRUE if the Document object is the <i>Latest Version</i> in its <i>Version Series</i> . FALSE otherwise.	
2239		
2240	cmis:isMajorVersion	Boolean
2241	TRUE if the Document object is a <i>Major Version</i> in its <i>Version Series</i> . FALSE otherwise.	
2242		
2243	cmis:isLatestMajorVersion	Boolean
2244	TRUE if the Document object is the <i>Latest Major Version</i> in its <i>Version Series</i> . FALSE otherwise.	
2245		
2246	cmis:versionLabel	String (optional)
2247	Optional textual description the position of an individual object with respect to the version series.	
2248	(For example, version 1.0).	
2249		
2250	cmis:versionSeriesId	ID
2251	ID of the Version Series for this Object.	
2252		
2253	cmis:isVersionSeriesCheckedOut	Boolean
2254	TRUE if there currently exists a Private Working Copy for this Version Series. FALSE otherwise	

2255

2256 **cmis:versionSeriesCheckedOutBy** String

2257 If IsVersionSeriesCheckedOut is TRUE: then an identifier for the user who created the Private

2258 Working Copy. `""`Not set otherwise.

2259

2260 **cmis:versionSeriesCheckedOutId** ID

2261 If IsVersionSeriesCheckedOut is TRUE: The Identifier for the Private Working Copy. `""`Not set

2262 otherwise.

2263

2264 **cmis:checkinComment** String

2265 Textual comment associated with the given version.

2266 *Note:* Changes made via the Versioning Services that affect the values of these properties MUST NOT

2267 constitute modifications to the Document objects in the Version Series (e.g. MUST NOT affect the

2268 cmis:lastModificationDate, etc.)

2269 **2.1.9.6 Document Creation and Initial Versioning State**

2270 A repository MAY create new Document objects in a `""Private Working Copy""` state when they are

2271 created via the *createDocument* or *createDocumentFromSource* services. This state is logically

2272 equivalent to having a Version Series that contains exactly one object (the PWC) and 0 other documents.

2273 The repository MAY also create new Document objects in a `""Major Version""` state. This state is logically

2274 equivalent to having a Version Series that contains exactly one Major Version and 0 other documents.

2275 The repository MAY also create new Document objects in a `""Non-Major Version""` state. This state is

2276 logically equivalent to having a Version Series that contains exactly one Non-Major Version and 0 other

2277 documents.

2278 If the repository does not support versioning the repository MUST ignore the value of the versioningState

2279 parameter.

2280 **2.1.9.7 Version Specific/Independent membership in Folders**

2281 Repositories MAY treat membership of a Document object in a folder collection as `""version-specific""` or

2282 `""version-independent""`.

2283 Repositories MUST indicate whether they support version-specific membership in a folder via the

2284 `""VersionSpecificFiling""` optional capability flag.

2285 If the repository is treating folder collection membership as `""version-independent""`, then:

- 2286 • Moving or Filing a Document Object into a folder MUST result in ALL Documents in the Version
- 2287 Series being moved/added into the folder.
- 2288 • The Repository MAY return only the latest-version OR latest major-version Document object in a
- 2289 version series in the response to Navigation service requests (*getChildren*, *getDescendants*), and
- 2290 NEED NOT return other Document Objects filed in the folder that are in the Version Series.

2291 If the repository is treating folder collection membership as `""version-specific""`, then moving or Filing a

2292 Document Object into a folder MUST NOT result in other Documents in the Version Series being

2293 moved/added.

2294 **2.1.9.8 Version Specific/Independent membership in Relationships**

2295 A relationship object MAY have either a version-specific or version-independent binding to its source

2296 and/or target objects. This behavior MAY vary between repositories and between individual relationship

2297 types defined for a Repository.

2298 If a relationship object has a version-independent binding to its source/target object, then:

2299 • The getObjectRelationships service invoked on a Document Object MUST return the relationship
2300 if Relationship was source/target is set to ANY Document Object in the Version Series.

2301 If a relationship object has a version-specific binding to its source/target object, then:

2302 • The getObjectRelationships service invoked on a Document Object MUST return the relationship
2303 if Relationship was source/target is set to the ID of the Document Object on which the service was
2304 invoked.

2305 **2.1.9.9 Versioning visibility in Query Services**

2306 Repositories MAY include non-latest-versions of Document Objects in results to the Discovery Services
2307 (query).

2308 Repositories MUST indicate whether they support querying for non-latest-versions via the
2309 `“AllVersionsSearchable”` optional capability flag.

2310 If `“AllVersionsSearchable”` is TRUE then the Repository MUST include in the query results ANY
2311 Document Object in the Version Series that matches the query criteria. (subject to other query constraints
2312 such as security.)

2313 Additionally, repositories MAY include Private Working Copy objects in results in results to the Discovery
2314 Services (query).

2315 Repositories MUST indicate whether they support querying for Private Working Copy objects via the
2316 `“PWCSearchable”` optional capability flag.

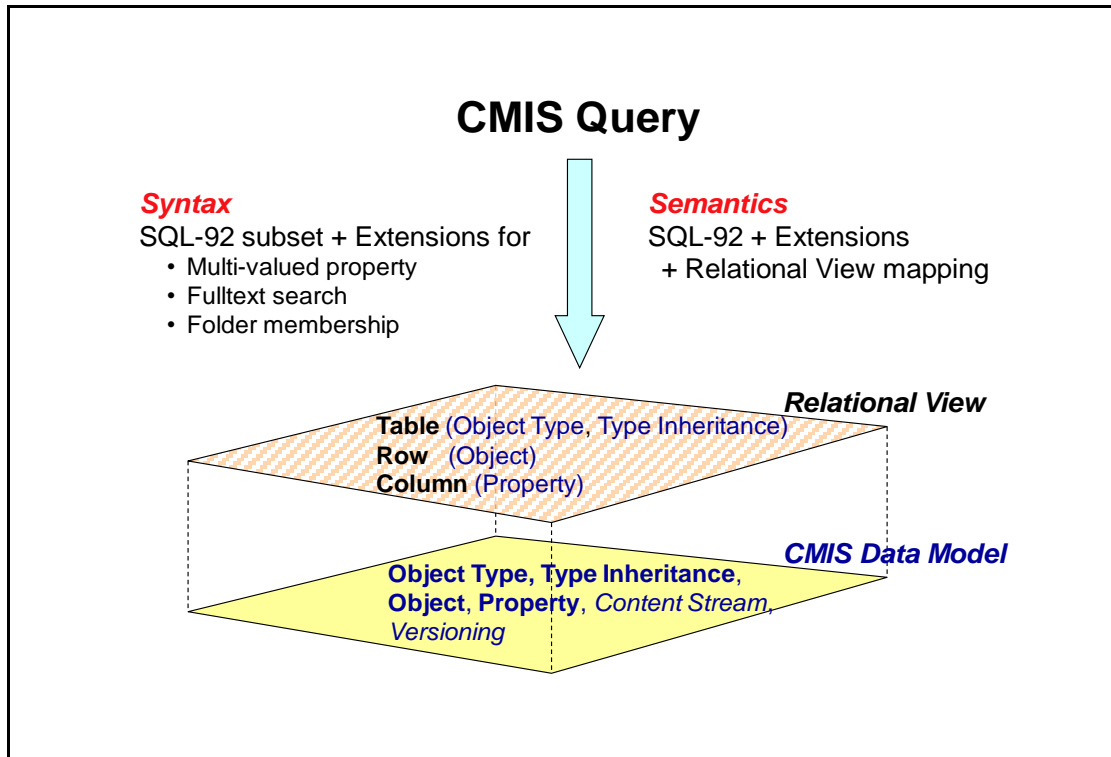
2317 If `“PWCSearchable”` is TRUE then the Repository MUST include in the query results ANY Private
2318 Working Copy Document Objects that matches the query criteria (subject to other query constraints such
2319 as security.)

2320 If `“PWCSearchable”` is FALSE then the Repository MUST NOT include in the query results ANY Private
2321 Working Copy Document Objects that match the query criteria (subject to other query constraints such as
2322 security.)

2323 **2.1.10 Query**

2324 CMIS provides a type-based query service for discovering objects that match specified criteria, by
2325 defining a read-only projection of the CMIS data model into a *Relational View*.

2326 Through this relational view, queries may be performed via a simplified SQL SELECT statement. This
2327 query language is based on a subset of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database
2328 Language SQL), with a few extensions to enhance its filtering capability for the CMIS data model, such as
2329 existential quantification for multi-valued property, full-text search, and folder membership. Other
2330 statements of the SQL language are not adopted by CMIS. The semantics of this query language is
2331 defined by the SQL-92 standard, plus the extensions, in conjunction with the model mapping defined by
2332 CMIS’s relational view.



2333

2334 2.1.10.1 Relational View Projection of the CMIS Data Model

2335 The relational view of a CMIS repository consists of a collection of virtual tables that are defined on top of
2336 the CMIS data model. This relational view is used for query purposes only.

2337 In this relational view a **Virtual Table** is implicitly defined for each *queryable* Object-Type defined in the
2338 repository. (Non-queryable Object-Types are NOT exposed through this Relational View.)

2339 In each **Virtual Table**, a **Virtual Column** is implicitly defined for each property defined in the Object-Type
2340 Definition AND for all properties defined on ANY ancestor-type of the Object-Type but NOT defined in the
2341 Object-Type definition. Virtual Columns for properties defined on ancestor-types of the Object-type but
2342 NOT defined in the Object-Type definition MUST contain the SQL NULL value. Virtual Columns for
2343 properties whose value is `"not set"` MUST contain the SQL NULL value.

2344 An object-type's *queryName* attribute is used as the table name for the corresponding virtual table, and a
2345 property's *queryName* attribute is used as the column name for the corresponding table column. Please
2346 see the restrictions on *queryName* in the appropriate data model section.

2347 The Virtual Column for a multi-valued property MUST contain a single list value that includes all values of
2348 the property.

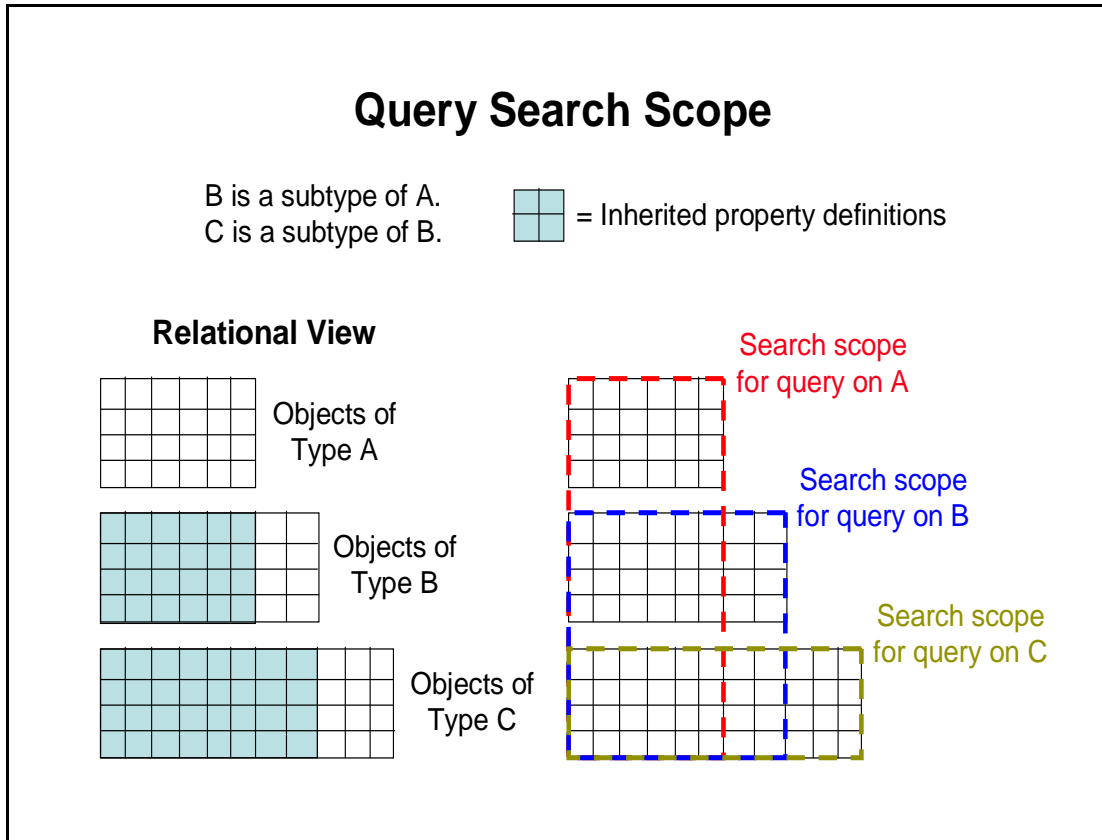
2349 2.1.10.1.1 Object-Type Hierarchy in the Relational View Projection

2350 The Relational View projection of the CMIS Data Model ensures that the Virtual Table for a particular
2351 Object-type is a complete super-set of the Virtual Table for any and all of its ancestor types.

2352 Additionally, an Object-Type definition's `"includedInSupertypeQuery"` specifies whether objects of that
2353 Object-Type MUST be included in the Virtual Table for any of its ancestor types. If the
2354 `"includedInSupertypeQuery"` attribute of the Object-Type is FALSE, then objects of that Object-Type
2355 MUST NOT be included in the Virtual Table for any of its ancestor types.

2356 Thus the Virtual Table for an Object-type includes a row not only for each Object of that type, but all
 2357 | Objects of any of that Object-types' Descendant Types for which the `includedInSupertypeQuery`
 2358 attribute is TRUE.

2359 But since the Virtual Table will include only columns for properties defined in the Object-Type underlying
 2360 the Virtual Table, a row that is a query result representing an Object of a Descendant Type can only
 2361 include those columns for properties defined on the Object-Type underlying the Virtual Table.



2362

2363 **2.1.10.1.2 Content Streams**

2364 Content-streams are NOT exposed through this relational view.

2365 **2.1.10.1.3 Result Set**

2366 When a query is submitted, a set of pseudo CMIS objects will be returned. These pseudo objects are
 2367 comprised of the properties specified in the select clause of the query statement.

2368 For each property in each object in the result set, the Repository MUST include the property definition ID
 2369 as well as either the query name (if no alias is used) or the alias in place of the query name (if an alias is
 2370 used).

2371 If the select clause of the query statement contains properties from a single type reference then the
 2372 repository MAY represent these pseudo-objects with additional object information.

2373 **2.1.10.2 Query Language Definition**

2374 This query languages is based on a subset of the SQL-92 grammar. CMIS-specific language extensions
 2375 to SQL-92 are called out explicitly.

2376 The basic structure of a CMIS query is a SQL statement that MUST include the following clauses:

2377 • **SELECT [virtual columns]:** This clause identifies the set of virtual columns that will be included
2378 in the query results for each row.

2379 • **FROM [Virtual Table Names]:** This clause identifies which Virtual Table(s) the query will run
2380 against.

2381 Additionally, a CMIS query MAY include the following clauses:

2382 • **WHERE [conditions]:** This clause identifies the constraints that rows MUST satisfy to be
2383 considered a result for the query.

2384 • **ORDER BY [sort specification]:** This clause identifies the order in which the result rows MUST
2385 be sorted in the result row set.

2386 2.1.10.2.1 BNF Grammar

2387 | This BNF grammar is a "subset" of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language
2388 SQL), except for some production alternatives. Specifically, except for these extensions, the following
2389 production rules are derived from the SQL-92 grammar. The non-terminals used in this grammar are also
2390 borrowed from the SQL-92 grammar without altering their semantics. Accordingly, the non-terminal
2391 <column name> is used for single-valued properties only so that the semantics of SQL can be preserved
2392 and borrowed. This approach not only facilitates comparison of the two query languages, and simplifies
2393 the translation of a CMIS query to a SQL query for a RDBMS-based implementation, but also allows
2394 future expansion of this query language to cover a larger subset of SQL with minimum conflict. The CMIS
2395 extensions are introduced primarily to support multi-valued properties and full-text search, and to test
2396 folder membership. Multi-valued properties are handled separately from single-valued properties, using
2397 separate non-terminals and separate production rules to prevent the extensions from corrupting SQL-92
2398 semantics.
2399

2400
2401 <CMIS 1.0 query statement> ::= <simple table> [<order by clause>]
2402 <simple table> ::= SELECT <select list> <from clause> [<where clause>]
2403 <select list> ::= "*"
2404 | <select sublist> [{ " " <select sublist> }...]
2405 <select sublist> ::= <value expression> [[AS] <column name>]
2406 | <qualifier> "*"
2407 | <multi-valued-column reference>
2408 <value expression> ::= <column reference> | <numeric value function>
2409 <column reference> ::= [<qualifier> " "] <column name>
2410 <multi-valued-column reference> ::= [<qualifier> " "] <multi-valued-column name>
2411 <numeric value function> ::= SCORE()
2412 <qualifier> ::= <table name> | <correlation name>
2413 <from clause> ::= FROM <table reference>
2414 <table reference> ::= <table name> [[AS] <correlation name>]
2415 | <joined table>
2416 <joined table> ::= "(" <joined table>)"
2417 | <table reference> [<join type>] JOIN <table reference> <join specification>
2418 <join type> ::= INNER | LEFT [OUTER]
2419 <join specification> ::= ON <column reference> "=" <column reference>
2420 <where clause> ::= WHERE <search condition>
2421 <search condition> ::= <boolean term> | <search condition> OR <boolean term>
2422 <boolean term> ::= <boolean factor> | <boolean term> AND <boolean factor>
2423 <boolean factor> ::= [NOT] <boolean test>
2424 <boolean test> ::= <predicate> | "(" <search condition>)"
2425 <predicate> ::= <comparison predicate> | <in predicate> | <like predicate> | <null predicate>
2426 | <quantified comparison predicate> | <quantified in predicate>
2427 | <text search predicate> | <folder predicate>
2428 <comparison predicate> ::= <value expression> <comp op> <literal>
2429 <comp op> ::= "=" | ">" | "<" | ">=" | "<=" | ">="
2430 <literal> ::= <signed numeric literal> | <character string literal> | <datetime literal> | <boolean literal>
2431 <in predicate> ::= <column reference> [NOT] IN "(" <in value list>)"
2432 <in value list> ::= <literal> [{ " " <literal> }...]
2433 <like predicate> ::= <column reference> [NOT] LIKE <character string literal>
2434 <null predicate> ::= { <column reference> | <multi-valued-column reference> } IS [NOT] NULL
2435 <quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>
2436 <quantified in predicate> ::= ANY <multi-valued-column reference> [NOT] IN "(" <in value list>)"
2437 <text search predicate> ::= CONTAINS "("
2438 | [<qualifier> " "] <quote> <text search expression> <quote>)"
2439 <folder predicate> ::= { IN_FOLDER | IN_TREE } "(" [<qualifier> " "] <folder id>)"
2440 <order by clause> ::= ORDER BY <sort specification> [{ " " <sort specification> }...]
2441 <sort specification> ::= <column reference> [ASC | DESC]
2442 <correlation name> ::= <identifier>

2443 <table name> ::= <identifier> !! This MUST be the name of an object-type.

2444 <column name> ::= <identifier> !! This MUST be the name of a single-valued property,
2445 or an alias for a scalar output value.

2446 <multi-valued-column name> ::= <identifier> !! This MUST be the name of a multi-valued property.

2447 <folder id> ::= <character string literal> !! This MUST be the object identity of a folder object.

2448 <identifier> ::= !! As defined by queryName attribute.

2449 <signed numeric literal> ::= !! As defined by SQL-92 grammar.

2450 <character string literal> ::= !! As defined by SQL-92 grammar. (i.e. enclosed in single-quotes)

2451

2452 | !! This is [an independent sub-grammar for](#) full-text search criteria. [It is isolatable from the query](#)
2453 | [statement grammar. \(See 2.1.10.3 Escaping\)](#)

2454 <text search expression> ::= <conjunct> [{<space> OR <space> <conjunct>} ...]

2455 <conjunct> ::= <term> [{<space> <term>} ...]

2456 | <term> ::= ["-"] <simple term>

2457 <simple term> ::= <word> | <phrase>

2458 <word> ::= <non space char> [{<non space char>} ...]

2459 <phrase> ::= <quote> <word> [{<space> <word>} ...] <quote>

2460 <space> ::= <space char> [{<space char>} ...]

2461 <non space char> ::= <char> - <space char>

2462 | <space char> ::= " "

2463 <char> ::= !! Any character

2464

2465 <datetime literal> ::= TIMESTAMP <quote> <datetime string> <quote>

2466 <datetime string> ::= YYYY-MM-DDThh:mm:ss.sss[Z | +hh:mm | -hh:mm]

2467 <boolean literal> ::= TRUE | FALSE | true | false

2468 | <quote> ::= "''"!! Single-quote only, consistent with SQL-92 string literal

2469 **2.1.10.2.2 SELECT Clause**

2470 The SELECT clause MUST contain exactly one of the following:

- 2471 • A comma separated list of one or more column names.
 - 2472 ○ If an explicit column list is provided: A repository MUST include in its result row set all of the
 - 2473 columns specified in the SELECT clause.
- 2474 • * : If this token is specified, then the repository MUST return columns for ALL single-valued
- 2475 properties defined in the Object-Types whose Virtual Tables are listed in the FROM clause, and
- 2476 SHOULD also return all multi-valued properties.

2477 | All column names MUST be valid "queryName" values for properties that are defined as "queryable" in
2478 the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

2479 **2.1.10.2.3 FROM Clause**

2480 The FROM clause identifies which Virtual Table(s) the query will be run against, as described in the
2481 previous section.

2482 The FROM clause MUST contain only the queryNames of Object-Types whose *queryable* attribute value
2483 is TRUE.

2484 **2.1.10.2.3.1 Join Support**

2485 CMIS repositories MAY support the use of SQL JOIN queries, and MUST indicate their support level
2486 using the [Optional Capability](#) attribute `capabilityJoin`.

2487

- If the Repository's value for the capabilityJoin attribute is **none**, then no JOIN clauses can be
2488 used in queries.

2489

- If the Repository's value for the capabilityJoin attribute is **inneronly**, then only inner JOIN
2490 clauses can be used in queries.

2491

- If the Repository's value for the capabilityJoin attribute is **innerandouter**, then inner and/or outer
2492 JOIN clauses can be used in queries.

2493 Only explicit joins using the `JOIN` keyword is supported. Queries MUST NOT include implicit joins as
2494 part of the WHERE clause of a CMIS query.

2495 CMIS queries MUST only support join operations using the `equality` predicate on single-valued
2496 properties.

2497 **2.1.10.2.4 WHERE Clause**

2498 This clause identifies the constraints that rows MUST satisfy to be considered a result for the query.

2499 All column names MUST be valid `queryName` or their aliased values for properties that are defined as
2500 `queryable` in the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

2501 Properties are defined to not support a `null` value, therefore the `<null predicate>` MUST be interpreted
2502 as testing the not set or set state of the specified property.

2503 **2.1.10.2.4.1 Comparisons permitted in the WHERE clause.**

2504 SQL's simple comparison predicate, IN predicate, and LIKE predicate are supported, for single-valued
2505 properties only (so that SQL's semantics is preserved). Boolean conjunction (AND), disjunction (OR), and
2506 negation (NOT) of predicates are also supported.

2507 Repositories SHOULD support the comparisons for the property types as described in the list below.
2508 Repositories MAY support additional comparisons and operators. Any additional operators not specified
2509 are repository-specific:

2510
2511 `<Property Type>`

2512 Supported Operators: `<List of Operators supported on Type>`

2513 Supported Literal: `<Supported type of Literal in comparison>`

2514
2515 String (Single)

2516 Supported Operators: `=, <>, [NOT] LIKE`

2517 Supported Literal: String

2518
2519 String (IN)

2520 Supported Operators: `[NOT] IN`

2521 Supported Literal: List of Strings

2522
2523 Decimal

2524 Supported Operators: `=, <>, <, <=, >, >=`

2525 Supported Literal: Decimal

2526

2527 Decimal (IN)
2528 Supported Operators: [NOT] IN
2529 Supported Literal: List of Decimal
2530
2531 Integer
2532 Supported Operators: =, <>, <, <=, >, >=
2533 Supported Literal: Integer
2534
2535 Integer (IN)
2536 Supported Operators: [NOT] IN
2537 Supported Literal: List of Integer
2538
2539 Boolean
2540 Supported Operators: =
2541 Supported Literal: <boolean literal>
2542
2543 DateTime
2544 Supported Operators: =, <>, <*, <=*, >*, >=*
2545 Supported Literal: <datetime literal>
2546 * - comparison is based on chronological before or after date.
2547
2548 DateTime (IN)
2549 Supported Operators: [NOT] IN
2550 Supported Literal: List of <datetime literal>!
2551
2552 ID
2553 Supported Operators: =, <>
2554 Supported Literal: String
2555
2556 ID (IN)
2557 Supported Operators: [NOT] IN
2558 Supported Literal: List of strings
2559
2560 URI
2561 Supported Operators: =, <>
2562 Supported Literal: String
2563
2564 URI (IN)
2565 Supported Operators: [NOT] IN
2566 Supported Literal: List of strings
2567
2568 URI

2569 Supported Operators: [NOT] LIKE

2570 Supported Literal: String

2571

2572 Operations on the SCORE() output MUST be treated the same as decimal operations.

2573

2574 When using properties in a join statement, comparison MUST be allowed on properties of the same types
2575 as defined by the table above. Repositories MAY extend this behavior.

2576

2577 The ANY operation argument MUST be one of the properties found in the table above which supports
2578 equality operations

2579 **2.1.10.2.4.2 Multi-valued property support (SQL-92 Extension)**

2580 The CMIS query language includes several new non-terminals to expose semantics for querying multi-
2581 valued properties, in a way that does not alter the semantics of existing SQL-92 production rules.

2582 **2.1.10.2.4.2.1 Multi-valued column references**

2583 **BNF grammar structure:** <Multi-valued-column reference>, <multi-valued-column name>

- 2584 • These are non-terminals defined for multi-valued properties whereas SQL-92's <column reference>
2585 and <column name> are retained for single-valued properties only. This is to preserve the single-
2586 value semantics of a regular `"column"` in the SQL-92 grammar.

2587 **2.1.10.2.4.2.2 <Quantified comparison predicate>**

2588 The SQL-92 production rule for <quantified comparison predicate> is extended to accept a multi-valued
2589 property in place of a <table subquery>. This operation is restricted to equality tests only.

2590

2591 <Table subquery> is not supported in CMIS-SQL.

2592

2593 The SQL-92 <quantifier> is restricted to ANY only.

2594

2595 The SQL-92 <row value constructor> is restricted to a literal only.

2596 **Example:**

```
2597 SELECT      Y.CLAIM_NUM, X.PROPERTY_ADDRESS, Y.DAMAGE_ESTIMATES  
2598 | FROM (POLICY AS X JOIN CLAIMS AS Y ON (X.POLICY_NUM = Y.POLICY_NUM )  
2599 | WHERE      ( 100000 = ANY Y.DAMAGE_ESTIMATES )
```

2600 *(Note: DAMAGE_ESTIMATES is a multi-valued Integer property.)*

2601 **2.1.10.2.4.2.3 IN/ANY Predicate**

2602 **BNF grammar structure:** <Quantified in predicate>

2603

2604 CMIS-SQL exposes a new IN predicate defined for a multi-valued property. It is modeled after the SQL-
2605 92 IN predicate, but since the entire predicate is different semantically, it has its own production rule in
2606 the BNF grammar below.

2607

2608 The quantifier is restricted to ANY. The predicate MUST be evaluated to TRUE if at least one of the
2609 | property's values is (or, is not, if NOT is specified) among the given list of literal values. Otherwise the
2610 predicate is evaluated to FALSE.

2611

2612 The ANY operation argument MUST be one of the properties found in the comparison list above which
2613 supports IN operations.

2614 **Example:**

```
2615     SELECT      *  
2616     FROM  CAR_REVIEW  
2617     WHERE      (MAKE = "buick") OR  
2618                ( ANY FEATURES IN ("NAVIGATION SYSTEM", "SATELLITE RADIO", "MP3")  
2619                ) (Note: FEATURES is a multi-valued String property.)
```

2620 2.1.10.2.4.3 CONTAINS() predicate function (CMIS-SQL Extension)

2621 **BNF grammar structure::** CONTAINS ([<qualifier> ,] " <text search expression> ")

2622 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
2623 by a Repository (See [previous section](#).)

2624 **Inputs:**

2625 <Qualifier>

2626 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2627 FROM clause for the query.

2628

- If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
2629 Table, but a repository MAY ignore the value of the parameter.

2630

- If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
2631 throw an exception if the qualifier is not specified.

2632 <Text Search Expression>

2633 The <text search expression> parameter MUST be a character string , specifying the full-text
2634 search criteria.

2635
2636 The Text Search Expression may be a set of terms or phrases with an optional "-" to signal
2637 negation. A phrase is defined as a word or group of words. A group of words must be
2638 surrounded by quotes to be considered a single phrase.

2639
2640 Terms separated by whitespace are AND'ed together.

2641 Terms separated by "OR" are OR'ed together

2642 Implicit "AND" has higher precedence than "OR"

2643 Within a word or phrase, each (single-)quote must also be escaped by a preceding backslash "\"

2644 **Return value:**

2645 The predicate returns a Boolean value.

2646 The predicate MUST return TRUE if the object is considered by the repository as "relevant" with
2647 respect to the given <text search expression> parameter.

2648 The predicate MUST return FALSE if the object is considered by the repository as not "relevant"
2649 with respect to the given <text search expression> parameter.

2650 **Constraints:**

2651 At most one CONTAINS() function MUST be included in a single query statement. The repository
2652 MUST throw an exception if more than one CONTAINS() function is found.

2653
2654 The return value of the CONTAINS() function MAY only be included conjunctively (ANDed) with the
2655 aggregate of all other predicates, if there is any, in the WHERE clause.

2656 **2.1.10.2.4.4 SCORE() predicate function**

2657 **BNF grammar structure:** SCORE ()

2658 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
2659 by a Repository (See previous section.)

2660 **Inputs:** No inputs MUST be provided for this predicate function.

2661 **Return value:**

2662 The SCORE() predicate function returns a decimal value in the interval [0,1] .

2663 A repository MUST return the value 0 if the object is considered by the repository as having
2664 absolutely no relevance with respect to the CONTAINS() function specified in the query.

2665 A repository MUST return the value 1 if the object is considered by the repository as having
2666 absolutely complete relevance with respect to the CONTAINS() function specified in the query.

2667 **Constraints:**

2668 The SCORE() function MUST only be used in queries that also include a CONTAINS() predicate
2669 function

2670 The SCORE() function MUST only be used in the SELECT clause of a query. It MUST NOT be
2671 used in the WHERE clause or in the ORDER BY clauses.

2672 An alias column name defined for the SCORE() function call in the SELECT clause (i.e.,
2673 `""SELECT SCORE() AS column_name ...""`) may be used in the ORDER BY clause.

2674 If SCORE() is included in the SELECT clause and an alias column name is not provided, then a
2675 query name of SEARCH_SCORE is used for the query output, and the property definition ID is
2676 repository-specific.

2677 **2.1.10.2.4.5 IN_FOLDER() predicate function**

2678 **BNF grammar structure:** IN_FOLDER([<qualifier>,] <folder id>)

2679 **Usage:** This is a predicate function that tests whether or not a candidate object is a child-object of the
2680 folder object identified by the given <folder id>.

2681 **Inputs:**

2682 **<qualifier>**

2683 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2684 FROM clause for the query.

- 2685
- 2686 • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual Table,
but a repository MAY ignore the value of the parameter.
 - 2687 • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD throw an
2688 exception if the qualifier is not specified.

2689 **<folder id>**

2690 The value of this parameter MUST be the ID of a folder object in the repository.

2691 **Return value:**

2692 The predicate function MUST return TRUE if the object is a child-object of the folder specified by
2693 <folder id>.

2694 The predicate function MUST return FALSE if the object is a NOT a child-object of the folder
2695 specified by <folder id>.

2696 **2.1.10.2.4.6 IN_TREE() predicate function**

2697 **BNF grammar structure:** IN_TREE([<qualifier>,] <folder id>)

2698 **Usage:** This is a predicate function that tests whether or not a candidate object is a descendant-object of
2699 the folder object identified by the given <folder id>.

2700 **Inputs:**

2701 **<qualifier>**

2702 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2703 FROM clause for the query.

2704 | • o If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
2705 | Table, but a repository MAY ignore the value of the parameter.

2706 | • o If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
2707 | throw an exception if the qualifier is not specified.

2708 **<folder id>**

2709 The value of this parameter MUST be the ID of a folder object in the repository.

2710 **Return value:**

2711 The predicate function MUST return TRUE if the object is a descendant-object of the folder
2712 specified by <folder id>.

2713 The predicate function MUST return FALSE if the object is a NOT a descendant -object of the
2714 folder specified by <folder id>.

2715 **2.1.10.2.5 ORDER BY Clause**

2716 This clause MUST contain a comma separated list of one or more column names.

2717 | All column names referenced in this clause MUST be valid `"_queryName_"` or their aliased values for
2718 | properties defined as *orderable* in the Object-type(s) whose Virtual Tables are listed in the FROM clause.

2719 Only columns in the SELECT clause MAY be in the ORDER BY clause.

2720 Collation rules for the ORDER BY clause are repository specific.

2721 **2.1.10.3 Escaping**

2722 | Character escaping for character strings differs from SQL-92's escaping. A repository MUST support the
2723 | escaping of certain literal characters in a character string, or in a text expression, using a backslash
2724 | character (\) in the following manner. For a <character string literal>, which MUST BE a string enclosed
2725 | in single-quotes according to the SQL-92 grammar, any occurrence of the single-quote character (') and
2726 | the escape character (\) in the string MUST BE escaped. This applies to <folder id>, which is a <character
2727 | string literal>. Furthermore, when a <character string literal> is used in a LIKE predicate, any occurrence
2728 | of the percent character (%) and the underscore character (_) in the string as a literal MUST BE escaped
2729 | also. Therefore, within a quoted string in a query:

2730 | • The double character \ ' represents a literal single-quote (') character.

2731 | • The double character \ \ represents a literal backslash (\) character.

2732 | • Within a LIKE string, the double characters \% and \ _ represent a literal percent (%) character
2733 | and a literal underscore (_) character respectively.

2734 | • All other instances of a backslash (\) character are errors.

2735 | Using double single-quotes (") as a SQL-92 way to escape a literal single-quote (') character SHOULD
2736 | BE supported as an allowable alternative to the double character \.

2737 | For a <text search expression>, a second-level character escaping is required so that the <text search
2738 | expression> sub-grammar is *isolatable* from the query statement-level grammar. When a text search
2739 | expression is used in a query, the <text search expression> sub-grammar is *isolatable* from the query statement-level grammar.

2740 [expression is composed for a query according to the <text search expression> sub-grammar, any](#)
2741 [occurrence of the following three characters in the expression as a literal character MUST BE escaped:](#)
2742 [hyphen \(-\), single-quote \('\), and the escape character \(\\).](#) Then, before this expression is enclosed in
2743 [single-quotes and inserted into a CONTAINS\(\) predicate, the query statement-level escaping rules](#)
2744 [described in the above MUST BE applied. This two-level character escaping allows a query statement](#)
2745 [parser, using statement-level escaping rules, to correctly extract a <text search expression> as a](#)
2746 [character string literal independent of the <text search expression> sub-grammar. This extracted <text](#)
2747 [search expression> can then be correctly interpreted by a full-text search parser independent of the](#)
2748 [query-statement grammar, using second-level escaping rules. Since the <text search expression> sub-](#)
2749 [grammar is isolated from the SQL-92 grammar, double single-quotes is not a valid way to escape a literal](#)
2750 [single-quote character for second-level character escaping.](#)

2751 [An <identifier> in a query statement MUST conform to the SQL-92 identifier syntax, and MUST NOT](#)
2752 [require character escaping.](#)

2753 **Example 1:**

2754 [A query statement that contains a full-text search for the literal string "John'sPresentation-Version2" may](#)
2755 [be composed as:](#)

2756 `SELECT ... FROM ... WHERE ... CONTAINS('John\\'sPresentation\\-Version2') ...`

2757 [A query parser extracts from this statement the text search expression "John'sPresentation-Version2" as](#)
2758 [a character string literal, and passes it to a text-search parser, which interprets it as a single-word full-text](#)
2759 [search criteria: *John'sPresentation-Version2*.](#)

2760 **Example 2:**

2761 [A query statement that contains a full-text search for the phrase "Content Management" may be](#)
2762 [composed as:](#)

2763 `SELECT ... FROM ... WHERE ... CONTAINS('\Content Management')`

2764 **2.1.10.3** [A query parser extracts from this statement the text search expression "Content Management"](#)
2765 [as a character string literal, and passes it to a text-search parser, which interprets it as a full-text search](#)
2766 [criteria consisting of a single phrase: *Content Management*. There is no second-level escaping.](#)

2767 [Repositories MUST support the escaping of characters using a backslash \(\\) in the query statement. The](#)
2768 [backslash character \(\\) will be used to escape characters within quoted strings in the query as follows:](#)

2769 1. \ will represent a single-quote (') character

2770 2. \\ will represent a backslash (\) character

2771 3. Within a LIKE string, \% and _ will represent the literal characters % and _, respectively.

2772 4. All other instances of a \ are errors.

2773 4.

2774 4.

2775 4.

2776 4.

2777 4.

2778 4.

2779 4.

2780 4.

2781 4.


2782 4.

2783 4.


2784 4.

2785 4.

2786 4.

2787 

2788 4.

2789 | [4.](#)
2790 | [4.](#)
2791 | 
2792 | [4.](#)

2793 | 2.1.11 Change Log

2794 | CMIS provides a “change log” mechanism to allow applications to easily discover the set of changes
2795 | that have occurred to objects stored in the repository since a previous point in time. This change log can
2796 | then be used by applications such as search services that maintain an external index of the repository to
2797 | efficiently determine how to synchronize their index to the current state of the repository (rather than
2798 | having to query for all objects currently in the repository).

2799 | Entries recorded in the change log are referred to below as “change events”.

2800 | Note that change events in the change log MUST be returned in ascending order from the time when the
2801 | change event occurred.

2802 | 2.1.11.1 Completeness of the Change Log

2803 | The Change Log mechanism exposed by a repository MAY be able to return an entry for every change
2804 | ever made to content in the repository, or may only be able to return an entry for all changes made since
2805 | a particular point in time. This “completeness” level of the change log is indicated via the [optional](#)
2806 | [changesIncomplete](#) value found on the [getRepositoryInfo](#) service response

2807 | However, repositories MUST ensure that if an application requests the entire contents of the repository’s
2808 | change log, that the contents of the change log includes ALL changes made to any object in the
2809 | repository *after* the first change listed in the change log. (I.e. repositories MAY truncate events from the
2810 | change log on a “first-in first-out” basis, but not in any other order.)

2811 | A Repository MAY record events such as filing/unfiling/moving of Documents as change events on the
2812 | Documents, their parent Folder(s), or both the Documents and the parent Folders.

2813 | 2.1.11.2 Change Log Token

2814 | The primary index into the change log of a repository is the “change log token”. The change log token is
2815 | an opaque string that uniquely identifies a particular change in the change log.

2816 | 2.1.11.2.1 “Latest Change Token” repository information

2817 | Repositories that support the `changeLogToken` event MUST expose the latest change log token (i.e. the
2818 | change log token corresponding to the most recent change to any object in the repository) as a property
2819 | returned by the `getRepositoryInfo` service.

2820 | This will enable applications to begin “subscribing” to the change log for a repository by discovering
2821 | what change log token they should use on a going-forward basis to discover change events to the
2822 | repository.

2823 | 2.1.11.3 Change Event

2824 | A change event represents a single action that occurred to an object in the repository that affected the
2825 | persisted state of the object.

2826 | A Repository that supports the change log capability MUST expose at least the following information for
2827 | each change object:

- 2828 | • **ID ObjectID:** The ObjectID of the object to which the change occurred
- 2829 | • **Enum ChangeType:** An enumeration that indicates the type of the change. Valid values are:
 - 2830 | ○ `created`: The object was created.
 - 2831 | ○ `updated`: The object was updated.

- 2832 o deleted: The object was deleted
- 2833 o security: The access control or security policy for the object were changed.
- 2834 | • <Properties> properties: Additionally, for events of changeType `"_updated"`, the repository MAY
- 2835 optionally include the new values of properties on the object (if any).
- 2836 | Repositories MUST indicate whether they include properties for `"_updated"` change events via the
- 2837 optional `enumCapabilityChanges` `capability`.

2838

2839 2.2 Services

2840 The Services section of the CMIS specification defines a set of services that are described in a
2841 protocol/binding-agnostic fashion.

2842 Every protocol binding of the CMIS specification MUST implement all of the methods described in this
2843 section or explain why the service is not implemented.

2844 However, the details of how each service & method is implemented will be described in those protocol
2845 binding specifications.

2846 2.2.1 Common Service Elements

2847 The following elements are common across many of the CMIS services.

2848 2.2.1.1 Paging

2849 All of the methods that allow for the retrieval of a collection of CMIS objects support paging of their result
2850 sets except where explicitly stated otherwise. The following pattern is used:

2851 Input Parameters:

- 2852 • **(optional) Integer maxItems:** This is the maximum number of items to return in a response. The
2853 repository MUST NOT exceed this maximum. Default is repository-specific.
- 2854 • **(optional) Integer skipCount:** This is the number of potential results that the repository MUST
2855 skip/page over before returning any results. Defaults to 0.

2856 Output Parameters:

- 2857 • **Boolean hasMoreItems:** TRUE if the Repository contains additional items after those contained in
2858 the response. FALSE otherwise. If TRUE, a request with a larger skipCount or larger maxItems is
2859 expected to return additional results (unless the contents of the repository has changed).
- 2860 • **Integer numItems:** If the repository knows the total number of items in a result set, the repository
2861 SHOULD include the number here. If the repository does not know the number of items in a result
2862 set, this parameter SHOULD not be set. The value in the parameter MAY NOT be accurate the next
2863 time the client retrieves the result set or the next page in the result set.

2864 If the caller of a method does not specify a value for maxItems, then the Repository MAY select an
2865 appropriate number of items to return, and MUST use the hasMoreItems output parameter to indicate if
2866 any additional results were not returned.

2867 Repositories MAY return a smaller number of items than the specified value for maxItems.

2868 Each binding will express the above in context and may have different mechanisms for communicating
2869 hasMoreItems and numItems.

2870 2.2.1.2 Retrieving additional information on objects in CMIS service calls

2871 Several CMIS services that return object information have the ability to return dependent object
2872 information as part of their response, such as the Allowable Actions for an object, rendition information,
2873 etc.

2874 The CMIS service methods that support returning a result set of objects will include the ability to return
2875 the following object information:

- 2876 • Properties (retrieves a subset instead of additional information)
- 2877 • Relationships
- 2878 • Renditions
- 2879 • ACLs
- 2880 • AllowableActions

2881

2882 This section describes the input parameter & output pattern for those services. All input parameters are
2883 optional.

2884 2.2.1.2.1 Properties

2885 **Description:** All of the methods that allow for the retrieval of properties for CMIS Objects have a
2886 `"Property Filter"` as an optional parameter, which allows the caller to specify a subset of properties for
2887 Objects that MUST be returned by the repository in the output of the method.

2888 **Optional Input Parameter:**

- 2889 • **String filter:** Value indicating which properties for Objects MUST be returned. Values are:
 - 2890 • `Not set`: The set of properties to be returned MUST be determined by the repository.
 - 2891 • **A comma-delimited list of property definition Query Names: The properties listed MUST be**
2892 **returned.**
 - 2893 • `"*"`: All properties MUST be returned for all objects.

2894 ~~Repositories SHOULD return only the properties specified in the property filter if they exist on the object's~~
2895 ~~type definition. If a property is requested by a filter, a property element MUST be returned for that~~
2896 ~~property. A repository MAY return additional properties. If a returned property is in a "not set" state, a~~
2897 ~~value element MUST NOT be returned for that property.~~

2898

2899 | If a property filter specifies a property that is `not set`, it MUST be represented as a property element
2900 without a value element.

2901 2.2.1.2.2 Relationships

2902 **Description:** Used to retrieve the relationships in which the object(s) are participating.

2903 **Optional Input Parameter:**

- 2904 • **Enum includeRelationships:** Value indicating what relationships in which the objects returned
2905 participate MUST be returned, if any. Values are:
 - 2906 `none`: No relationships MUST be returned. (Default).
 - 2907 `source`: Only relationships in which the objects returned are the source MUST be
2908 returned.
 - 2909 `target`: Only relationships in which the objects returned are the target MUST be
2910 returned.
 - 2911 `both`: Relationships in which the objects returned are the source or the target MUST be
2912 returned.

2913 **Output Parameter for each object:**

- 2914 • **<Array> Relationships:** A collection of the relationship objects.

2915 2.2.1.2.3 Policies

2916 **Description:** Used to retrieve the policies currently applied to the object(s).

2917 **Optional Input Parameter:**

- **Boolean** `includePolicyIds`: If TRUE, then the Repository MUST return the Ids of the policies applied to the object. Defaults to FALSE.

2920 **Output Parameter or each object:**

- **<Array> Policies**: A collection of the policy objects.

2922 **2.2.1.2.4 Renditions**

2923 **Description:** Used to retrieve the renditions of the object(s).

2924 **Optional Input Parameter:**

- **String** `renditionFilter`: The Repository MUST return the set of renditions whose kind matches this filter. See section below for the filter grammar.
 - Defaults to `"cmis:none"`.

2928 **Output Parameter for each object:**

- **<Array> Renditions**: The set of renditions.

2930 **2.2.1.2.4.1 Rendition Filter Grammar**

2931 The Rendition Filter grammar is defined as follows:

```
2932 <renditionInclusion> ::= <none> | <wildcard> | <termlist>
2933 <termlist> ::= <term> | <term> ',',',' <termlist>
2934 <term> ::= <kind> | <mimetype>
2935 <kind> ::= <text>
2936 <mimetype> ::= <type>  '/'  '/'  <subtype>
2937 <type> ::= <text>
2938 <subtype> ::= <text> | <wildcard>
2939 <text> ::=  / * ! !  any char except whitespace  * / 
2940 <wildcard> ::=  ' ' * ' ' 
2941 <none> ::=  ' ' cmis:none ' ' 
```

2942 An inclusion pattern allows:

- **Wildcard** : include all associated Renditions
- **Comma-separated list of Rendition kinds or mimetypes** : include only those Renditions that match one of the specified kinds or mimetypes
- **cmis:none**: (Default) exclude all associated Renditions

2947 Examples:

- `*` (include all Renditions)
- `cmis:thumbnail` (include only Thumbnails)
- `Image/*` (include all image Renditions)
- `application/pdf, application/x-shockwave-flash` (include web ready Renditions)
- `cmis:none` (exclude all Renditions)

2953 **2.2.1.2.5 ACLs**

2954 **Description:** Used to retrieve the ACLs for the object(s) described in the service response.

2955 **Optional Input Parameter:**

- 2956 • **Boolean** `includeACL`: If TRUE, then the Repository MUST return the ACLs for each object in the
2957 result set. Defaults to FALSE.

2958 **Output Parameter for each object:**

- 2959 • **<Array> ACLs**: The list of access control entries of the ACL for the object.

2960 **2.2.1.2.6 Allowable Actions**

2961 **Description:** Used to retrieve the allowable actions for the object(s) described in the service response.

2962 **Optional Input Parameter:**

- 2963 • **Boolean** `includeAllowableActions`: If TRUE, then the Repository MUST return the available
2964 actions for each object in the result set. Defaults to FALSE.

2965 **Output Parameter for each object:**

- 2966 • **<Array> AllowableActions**: [The list of allowable actions for the object. See](#)
2967 [cmisAllowableActionsType in the CMIS schema.](#)

2968 **2.2.1.3 Change Tokens**

2969 | The CMIS base object-type definitions include an opaque string `"_ChangeToken"` property that a
2970 Repository MAY use for optimistic locking and/or concurrency checking to ensure that user updates do
2971 not conflict.

2972 If a Repository provides values for the ChangeToken property for an Object, then all invocations of the
2973 `"_update"` methods on that object (`updateProperties`, `setContentStream`, `deleteContentStream`) MUST
2974 provide the value of the `changeToken` property as an input parameter, and the Repository MUST throw
2975 an `updateConflictException` if the value specified for the `changeToken` does NOT match the
2976 `changeToken` value for the object being updated.

2977 **2.2.1.4 Exceptions**

2978 The following sections list the complete set of exceptions that MAY be returned by a repository in
2979 response to a CMIS service method call.

2980 **2.2.1.4.1 General Exceptions**

2981 The following exceptions MAY be returned by a repository in response to ANY CMIS service method call.

2982 | The `"_Cause"` field indicates the circumstances under which a repository SHOULD return a particular
2983 exception.

2984 **`invalidArgument`**

2985 Cause: One or more of the input parameters to the service method is missing or invalid.

2986

2987 **`objectNotFound`**

2988 Cause: The service call has specified an object that does not exist in the Repository.

2989

2990 **`notSupported`**

2991 Cause: The service method invoked requires [an optional capability](#) not supported by the
2992 repository.

2993

2994 **`permissionDenied`**

2995 Cause: The caller of the service method does not have sufficient permissions to perform the
2996 operation.

2997

2998 **runtime**

2999 Cause: Any other cause not expressible by another CMIS exception.

3000 **2.2.1.4.2 Specific Exceptions**

3001 The following exceptions MAY be returned by a repository in response to one or more CMIS service
3002 methods calls.

3003 For each exception, the general intent is listed as well as a list of the methods which MAY cause the
3004 exception to be thrown.

3005 **constraint**

3006 Intent: The operation violates a Repository- or Object-level constraint defined in the CMIS
3007 domain model.

3008 Methods:

3009 • **Navigation Services:**

3010 ○ getObjectParents

3011 • **Object Services:**

3012 ○ createDocument

3013 ○ createDocumentFromSource

3014 ○ createFolder

3015 ○ createRelationship

3016 ○ createPolicy

3017 ○ updateProperties

3018 ○ moveObject

3019 ○ deleteObject

3020 ○ setContentStream

3021 ○ deleteContentStream

3022 • **Multi-filing Services:**

3023 ○ addObjectToFolder

3024 • **Versioning Services:**

3025 ○ checkOut

3026 ○ cancelCheckOut

3027 ○ checkIn

3028 • **Policy Services:**

3029 ○ applyPolicy

3030 ○ removePolicy

3031 • **Change Log Services:**

3032 ○ getContentChanges

3033

3034 **contentAlreadyExists**

3035 Intent: The operation attempts to set the content stream for a Document that already has a
3036 content stream without explicitly specifying the `overwriteFlag` parameter.

3037 Methods:

3038 • **Object Services:**

3039 ○ setContentStream

3040

3041 **filterNotValid**

3042 Intent: The property filter or rendition filter input to the operation is not valid.

3043 Methods:

3044 • **Navigation Services:**

- 3045 ○ getDescendants
- 3046 ○ getChildren
- 3047 ○ getFolderParent
- 3048 ○ getObjectParents
- 3049 ○ getCheckedOutDocs

3050 • **Object Services:**

- 3051 ○ getProperties
- 3052 ○ getRenditions
- 3053 ○ getObject
- 3054 ○ getObjectByPath

3055 • **Versioning Services:**

- 3056 ○ getPropertiesOfLatestVersion
- 3057 ○ getAllVersions

3058 • **Policy Services:**

- 3059 ○ getAppliedPolicies

3060

3061 **nameConstraintViolation**

3062 Intent: The repository is not able to store the object that the user is creating/updating due to
3063 a name constraint violation.

3064 Methods:

3065 • **Object Services:**

- 3066 ○ createDocument
- 3067 ○ createDocumentFromSource
- 3068 ○ createFolder
- 3069 ○ createRelationship
- 3070 ○ createPolicy
- 3071 ○ updateProperties
- 3072 ○ moveObject

3073

3074 **storage**

3075 Intent: The repository is not able to store the object that the user is creating/updating due to
3076 an internal storage problem.

3077 Methods:

3078 • **Object Services:**

- 3079 ○ createDocument
- 3080 ○ createDocumentFromSource
- 3081 ○ createFolder
- 3082 ○ createRelationship

- 3083 ○ createPolicy
- 3084 ○ updateProperties
- 3085 ○ moveObject
- 3086 ○ setContentStream
- 3087 ○ deleteContentStream

- 3088 • **Versioning Services:**

- 3089 ○ checkOut
- 3090 ○ checkIn

3091

3092 **streamNotSupported**

3093 Intent: The operation is attempting to get or set a contentStream for a Document whose
 3094 Object-type specifies that a content stream is not allowed for Document's of that
 3095 type.

3096 Methods:

- 3097 • **Object Services:**

- 3098 ○ createDocument
- 3099 ○ createDocumentFromSource
- 3100 ○ getContentStream
- 3101 ○ setContentStream

- 3102 • **Versioning Services:**

- 3103 ○ checkIn

3104

3105 **updateConflict**

3106 Intent: The operation is attempting to update an object that is no longer current (as
 3107 determined by the repository).

3108 Methods:

- 3109 • **Object Services:**

- 3110 ○ updateProperties
- 3111 ○ moveObject
- 3112 ○ deleteObject
- 3113 ○ deleteTree
- 3114 ○ setContentStream
- 3115 ○ deleteContentStream

- 3116 • **Versioning Services:**

- 3117 ○ checkOut
- 3118 ○ cancelCheckOut
- 3119 ○ checkIn

3120

3121 **versioning**

3122 Intent: The operation is attempting to perform an action on a [non-current version](#) of a
 3123 Document that cannot be performed on a non-current version.

3124 Methods:

- 3125 • **Object Services:**

- 3126 o updateProperties
- 3127 o moveObject
- 3128 o setContentStream
- 3129 o deleteContentStream
- 3130 • **Versioning Services:**
- 3131 o checkOut
- 3132 o cancelCheckOut
- 3133 o checkIn

3134 **2.2.1.5 ACLs**

3135 Those services which allow for the setting of ACLs may take the optional macro cmis:user which allows
 3136 the caller to indicate the operation applies to the current authenticated user.

3137 **2.2.2 Repository Services**

3138 The Repository Services (getRepositories, getRepositoryInfo, getTypeChildren, getTypeDescendants,
 3139 getTypeDefinition) are used to discover information about the repository, including information about the
 3140 repository and the object-types defined for the repository.

3141 **2.2.2.1 getRepositories**

3142 **Description:** Returns a list of CMIS repositories available from this CMIS service endpoint.

3143 **2.2.2.1.1 Inputs**

3144 None.

3145 **2.2.2.1.2 Outputs**

3146 A list of repository information, with (at least) the following information for each entry:

- 3147 • **ID repositoryId:** The identifier for the Repository.
- 3148 • **String repositoryName:** A display name for the Repository.

3149 **2.2.2.1.3 Exceptions Thrown & Conditions**

3150 See section 2.2.1.4.1 General Exceptions

3151 **2.2.2.2 getRepositoryInfo**

3152 **Description:** Returns information about the CMIS repository, the [optional capabilities](#) it supports and its
 3153 Access Control information if applicable. .

3154 **2.2.2.2.1 Inputs**

3155 **Required:**

- 3156 • **ID repositoryId:** The identifier for the Repository.

3157 **2.2.2.2.2 Outputs**

- 3158 • **ID repositoryId:** The identifier for the Repository.
 - 3159 o **Note:** This MUST be the same identifier as the input to the method.
- 3160 • **String repositoryName:** A display name for the Repository.
- 3161 • **String repositoryDescription:** A display description for the Repository.

- 3162 | • **String** `vendorName`: A display name for the vendor of the Repository's underlying application.
- 3163 | • **String** `productName`: A display name for the Repository's underlying application.
- 3164 | • **String** `productVersion`: A display name for the version number of the Repository's
- 3165 | underlying application.
- 3166 | • **ID** `rootFolderId`: The ID of the Root Folder Object for the Repository.
- 3167 | • **<List of capabilities>**: The set of values for the repository-optional capabilities specified in
- 3168 | [section 2.1.1.1 Optional Capabilities](#)
- 3169 | • **String** `latestChangeLogToken`: The change log token corresponding to the most recent
- 3170 | change event for any object in the repository.
- 3171 | • **String** `cmisVersionSupported`: A decimal that indicates what version of the CMIS
- 3172 | specification this repository supports as specified in 2.1.1.2 Implementation Information.
- 3173 | • **URI** `thinClientURI`: A optional repository-specific URI pointing to the repository's web
- 3174 | interface.
- 3175 | • **Boolean** `changesIncomplete`: Indicates whether or not the repository's change log can return
- 3176 | all changes ever made to any object in the repository or only changes made after a particular
- 3177 | point in time. Applicable when the repository's optional capability `capabilityChanges` is not
- 3178 | none.
 - 3179 | o If FALSE, then the change log can return all changes ever made to every object.
 - 3180 | o If TRUE, then the change log includes all changes made since a particular point in time,
 - 3181 | but not all changes ever made.
- 3182 | • **<List of enum values>** `changesOnType`: Indicates whether changes are available for base
- 3183 | types in the repository. Valid values are from `enumBaseObjectTypes`. See [section 2.1.11](#)
- 3184 | [Change Log](#).
 - 3185 | o **cmis:document**
 - 3186 | o **cmis:folder**
 - 3187 | o **cmis:policy**
 - 3188 | o **cmis:relationship**
- 3189 | • **Enum** `supportedPermissions`: specifies which types of permissions are supported.
 - 3190 | o **basic**: indicates that the CMIS Basic permissions are supported.
 - 3191 | o **repository**: Indicates that repository specific permissions are supported.
 - 3192 | o **both**: indicates that both CMIS basic permissions and repository specific permissions are
 - 3193 | supported.
- 3194 | • **Enum** `propagation`: The list of allowed values for `applyACL`, which control how non-direct
- 3195 | ACEs are handled by the repository:
 - 3196 | o **objectonly**: indicates that the repository is able to apply ACEs without changing the
 - 3197 | ACLs of other objects – i.e. ACEs are applied, potentially [“breaking”](#) the [“sharing”](#)
 - 3198 | dependency for non-direct ACEs.
 - 3199 | o **propagate**: indicates that the repository is able to apply ACEs to a given object and
 - 3200 | propagate this change to all inheriting objects – i.e. ACEs are applied with the (intended)
 - 3201 | side effect to inheriting objects.
 - 3202 | o **repositorydetermined**: indicates that the repository uses its own mechanisms to
 - 3203 | handle non-direct ACEs when applying ACLs.
- 3204 | • **<Array> Permission** `permissions`: The list of repository-specific permissions the repository
- 3205 | supports for managing ACEs (see [section 2.8 Access Control](#)).
- 3206 | • **<Array> PermissionMapping** `mapping`: The list of mappings for the CMIS Basic permissions to
- 3207 | allowable actions (see [section 2.8 Access Control](#)).

- 3208
- **String** `principalAnonymous`: If set, this field holds the principal who is used for anonymous access. This principal can then be passed to the ACL services to specify what permissions anonymous users should have.
- 3209
- 3210
- 3211
- **String** `principalAnyone`: If set, this field holds the principal who is used to indicate any authenticated user. This principal can then be passed to the ACL services to specify what permissions any authenticated user should have.
- 3212
- 3213

3214 The `cmisRepositoryInfoType` schema describes the markup that will be included in all CMIS protocol
3215 bindings to implement this service.

3216 2.2.2.3 Exceptions Thrown & Conditions

3217 See section 2.2.1.4.1 General Exceptions

3218 2.2.2.3 `getTypeChildren`

3219 **Description:** Returns the list of [Object-Types](#) defined for the Repository that are children of the specified
3220 Type.

3221 2.2.2.3.1 Inputs

3222 **Required:**

- **String** `repositoryId`: The identifier for the Repository.

3224 **Optional:**

- **String** `typeId`: The typeId of an Object-Type specified in the Repository.
 - If specified, then the Repository MUST return all of child types of the specified type.
 - If not specified, then the Repository MUST return all Base Object-Types.
- **Boolean** `includePropertyDefinitions`: If TRUE, then the Repository MUST return the property definitions for each Object-Type returned.
 - If FALSE (default), the Repository MUST return only the attributes for each Object-Type.
- **Integer** `maxItems`: See section 2.2.1.1 Paging.
- **Integer** `skipCount`: See section 2.2.1.1 Paging.

3233 2.2.2.3.2 Outputs

3234 **<Array> Object-Types**: The list of child [Object-Types](#) defined for the given typeId.

3235 **Boolean** `hasMoreItems`: See section 2.2.1.1 Paging.

3236 **Optional:**

3237 **Integer** `numItems`: See section 2.2.1.1 Paging.

3238 2.2.2.3.3 Exceptions Thrown & Conditions

3239 See section 2.2.1.4.1 General Exceptions

3240 2.2.2.4 `getTypeDescendants`

3241 **Description:** Returns the set of descendant [Object-Types](#) defined for the Repository under the specified
3242 Type.

3243 **Notes:**

- This method does NOT support paging as defined in the 2.2.1.1 Paging section.
 - The order in which results are returned is repository-specific.
- 3245

3246 **2.2.2.4.1 Inputs**

3247 **Required:**

- 3248 • **String repositoryId:** The identifier for the Repository.

3249 **Optional:**

- 3250 • **String typeId:** The typeId of an Object-Type specified in the Repository.
 - 3251 ○ If specified, then the Repository MUST return all descendant types for the specified type.
 - 3252 ○ If not specified, then the Repository MUST return all types and MUST ignore the value of
 - 3253 the depth parameter
- 3254 • **Integer depth:** The number of levels of depth in the type hierarchy from which to return results.
3255 Valid values are:
 - 3256 ○ **1:** Return only types that are children of the type.
 - 3257 ○ **<Integer value greater than 1>:** Return only types that are children of the type and
 - 3258 descendants up to <value> levels deep.
 - 3259 ○ **-1:** Return ALL descendant types at all depth levels in the CMIS hierarchy.
 - 3260 ○ The default value is repository specific and SHOULD be at least 2 or -1.
- 3261 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
- 3262 definitions for each Object-Type returned.
- 3263 ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

3264 **2.2.2.4.2 Outputs**

3265 **<Array> Object-Types:** The hierarchy of [Object-Types](#) defined for the Repository.

3266 **2.2.2.4.3 Exceptions Thrown & Conditions**

3267 See section 2.2.1.4.1 General Exceptions

- 3268 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
- 3269 an invalid depth.

3270 **2.2.2.5 getTypeDefinition**

3271 **Description:** Gets the definition of the specified Object-Type.Inputs

3272 **2.2.2.5.1 Inputs**

3273 **Required:**

- 3274 • **String repositoryId:** The identifier for the Repository.
- 3275 • **String typeId:** The typeId of an Object-Type specified in the Repository.

3276 **2.2.2.5.2 Outputs**

- 3277 • Object-type including all property definitions. See section 2.1.3.3 (Object-Type Property
- 3278 Definitions) for further details.

3279 **2.2.2.5.3 Exceptions Thrown & Conditions**

3280 See section 2.2.1.4.1 General Exceptions

3281 2.2.3 Navigation Services

3282 The Navigation Services (getDescendants, getChildren, getFolderParent, getObjectParents,
3283 getCheckedoutDocs), are used to traverse the folder hierarchy in a CMIS **Repository**, and to locate
3284 Documents that are checked out.

3285 2.2.3.1 getChildren

3286 **Description:** Gets the list of child objects contained in the specified folder.

3287 **Notes:**

- 3288 | • If the Repository supports the optional `VersionSpecificFiling` capability, then the repository
3289 MUST return the document versions filed in the specified folder.
- 3290 | o Otherwise, the latest version of the documents MUST be returned.

3291 2.2.3.1.1 Inputs

3292 **Required:**

- 3293 | • **ID repositoryId:** The identifier for the Repository.
- 3294 | • **ID folderId:** The identifier for the folder.

3295 **Optional:**

- 3296 | • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3297 | • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3298 | • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and
3299 the ascending modifier `ASC` or the descending modifier `DESC` for each query name. A
3300 repository's handling of the orderBy input is repository-specific.
- 3301 | • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the
3302 matched object if they exist on the matched object type definition and in the filter.
- 3303 | • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3304 | • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3305 | • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3306 | • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
3307 child object for use in constructing that object's path.

3308 2.2.3.1.2 Outputs

- 3309 | • **<Array> ObjectResults:** A list of the child objects for the specified folder. Each object result
3310 MUST include the following elements if they are requested:
 - 3311 | o **<Array> Properties:** The list of properties for the object.
 - 3312 | o **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 3313 | o **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 3314 | o **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3315 | o **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.
- 3316 | • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3317 **Optional:**

3318 **Integer numItems:** See section 2.2.1.1 Paging.

3319 2.2.3.1.3 Exceptions Thrown & Conditions

- 3320 | • See section 2.2.1.4.1 General Exceptions

- 3321 • `filterNotValid`: The Repository MUST throw this exception if the `is_` property [or rendition](#) filter
3322 input parameter is not valid.
- 3323 • `invalidArgument`: if the specified folder is not a folder

3324 2.2.3.2 getDescendants

3325 **Description:** Gets the set of [descendant objects](#) contained in the specified folder or any of its child-
3326 folders.

3327 Notes:

- 3328 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3329 • The order in which results are returned is repository-specific..
- 3330 • If the Repository supports the optional capability `capabilityVersionSpecificFiling`, then
3331 the repository MUST return the document versions filed in the specified folder or its descendant
3332 folders. Otherwise, the latest version of the documents MUST be returned.
- 3333 • If the Repository supports the optional capability `capabilityMultifiling` and the same
3334 document is encountered multiple times in the hierarchy, then the repository MUST return that
3335 document each time is encountered.

3336 2.2.3.2.1 Inputs

3337 Required:

- 3338 • **ID repositoryId:** The identifier for the Repository.
- 3339 • **ID folderId:** The identifier for the folder.

3340 Optional:

- 3341 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results.
3342 Valid values are:
 - 3343 ○ **1:** Return only objects that are children of the folder.
 - 3344 ○ **<Integer value greater than 1>:** Return only objects that are children of the folder and
3345 descendants up to <value> levels deep.
 - 3346 ○ **-1:** Return ALL descendant objects at all depth levels in the CMIS hierarchy.
 - 3347 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 3348 • **String filter:** See section 2.2.1.2.1 Properties.
- 3349 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3350 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3351 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3352 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
3353 child object for use in constructing that object's path.

3354 2.2.3.2.2 Outputs

- 3355 • **<Array> ObjectResults:** A list of the descendant objects for the specified folder. Each object
3356 result MUST include the following elements if they are requested:
 - 3357 ○ **<Array> Properties:** The list of properties for the object.
 - 3358 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 3359 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 3360 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3361 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

3362 2.2.3.2.3 Exceptions Thrown & Conditions

3363 See section 2.2.1.4.1 General Exceptions

- 3364 • `filterNotValid`: The Repository MUST throw this exception if the `is` property `or rendition` filter
- 3365 input parameter is not valid.
- 3366 • `invalidArgument`: The Repository MUST throw this exception if the service is invoked with
- 3367 `"depth = 0"`.
- 3368 • `invalidArgument`: if the specified folder is not a folder

3369 2.2.3.3 getFolderTree

3370 **Description:** Gets the set of descendant folder objects contained in the specified folder.

3371

3372 **Notes:**

- 3373 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3374 • The order in which results are returned is repository-specific..

3375 2.2.3.3.1 Inputs

3376 **Required:**

- 3377 • **ID repositoryId:** The identifier for the Repository.
- 3378 • **ID folderId:** The identifier for the folder.

3379 **Optional:**

- 3380 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results.
- 3381 Valid values are:
 - 3382 ○ **1:** Return only folders that are children of the folder.
 - 3383 ○ **<Integer value greater than 1>:** Return only folders that are children of the folder and
 - 3384 descendant folders up to <value> levels deep.
 - 3385 ○ **-1:** Return ALL descendant folders at all depth levels in the CMIS hierarchy.
 - 3386 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 3387 • **String filter:** See section 2.2.1.2.1 Properties.
- 3388 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3389 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3390 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3391 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
- 3392 child object for use in constructing that object's path.

3393 2.2.3.3.2 Outputs

- 3394 • **<Array> ObjectResults:** A list of the descendant folders for the specified folder. Each object
- 3395 result MUST include the following elements if they are requested:
 - 3396 ○ **<Array> Properties:** The list of properties for the object.
 - 3397 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 3398 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 3399 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3400 ○ **String pathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

3401 **2.2.3.3 Exceptions Thrown & Conditions**

- 3402
- See section 2.2.1.4.1 General Exceptions
 - `filterNotValid`: The Repository MUST throw this exception if the `is` property `or rendition` filter input parameter is not valid.
 - `invalidArgument`: The Repository MUST throw this exception if the service is invoked with an invalid depth
 - `invalidArgument`: if the specified folder is not a folder

3408
3409

3410 **2.2.3.4 getFolderParent**

3411 **Description:** Gets the parent folder object for the specified folder object.

3412 **2.2.3.4.1 Inputs**

3413 **Required:**

- **ID repositoryId**: The identifier for the Repository.
- **ID folderId**: The identifier for the folder.

3416 **Optional:**

- **String filter**: See section 2.2.1.2.1 Properties.

3418 **2.2.3.4.2 Outputs**

- **Object**: The parent folder object of the specified folder.

3420 **2.2.3.4.3 Exceptions Thrown & Conditions**

- 3421
- See section 2.2.1.4.1 General Exceptions
 - `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter is not valid.
 - `invalidArgument`: The Repository MUST throw this exception if the folderId input is the root folder.
- 3425

3426 **2.2.3.5 getObjectParents**

3427 **Description:** Gets the parent folder(s) for the specified non-folder, fileable object.

3428 **2.2.3.5.1 Inputs**

3429 **Required:**

- **ID repositoryId**: The identifier for the Repository.
- **ID objectId**: The identifier for the object.

3432 **Optional:**

- **String filter**: See section 2.2.1.2.1 Properties
 - **Enum includeRelationships**: See section 2.2.1.2.2 Relationships.
 - **String renditionFilter**: See section 2.2.1.2.4 Renditions.
 - **Boolean includeAllowableActions**: See section 2.2.1.2.6 Allowable Actions.
 - **Boolean includeRelativePathSegment**: See section 2.1.5.3 Paths.
- 3437

3438 2.2.3.5.2 Outputs

- 3439 • **<Array> ObjectResults:** A list of the parent folder(s) of the specified objects. Empty for unfiled
3440 objects or for the root folder. Each object result MUST include the following elements if they are
3441 requested:
 - 3442 ○ **<Array> Properties:** The list of properties for the object.
 - 3443 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 3444 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 3445 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3446 ○ **String relativePathSegment:** If includeRelativePathSegment was TRUE. See section
3447 2.1.5.3 Paths.

3448 2.2.3.5.3 Exceptions Thrown & Conditions

- 3449 • See section 2.2.1.4.1 General Exceptions
- 3450 • **constraint:** The Repository MUST throw this exception if this method is invoked on an object
3451 who Object-Type Definition specifies that it is not fileable.
- 3452 • **filterNotValid:** The Repository MUST throw this exception if the [this](#) property [or rendition](#) filter
3453 input parameter is not valid.

3454 2.2.3.6 getCheckedOutDocs

3455 **Description:** Gets the list of documents that are checked out that the user has access to.

3456 2.2.3.6.1 Inputs

3457 Required:

- 3458 • **ID repositoryId:** The identifier for the Repository.

3459 Optional:

- 3460 • **ID folderId:** The identifier for a folder in the repository from which documents should be returned.
 - 3461 ○ If specified, the Repository MUST only return checked out documents that are child-
3462 objects of the specified folder.
 - 3463 ○ If not specified, the Repository MUST return checked out documents from anywhere in
3464 the repository hierarchy.
- 3465 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3466 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3467 • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and
3468 the ascending modifier `"ASC"` or the descending modifier `"DESC"` for each query name. A
3469 repository's handling of the orderBy input is repository-specific.
- 3470 • **String filter:** See section 2.2.1.2.1 Properties.
- 3471 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3472 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3473 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3474 2.2.3.6.2 Outputs

- 3475 • **<Array> ObjectResults:** A list of checked out documents. Each object result MUST include the
3476 following elements if they are requested:
 - 3477 ○ **<Array> Properties:** The list of properties for the object.
 - 3478 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.

- 3479 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3480 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3481 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3482 **Optional:**

- 3483 • **Integer numItems:** See section 2.2.1.1 Paging.

3484

3485 **2.2.3.6.3 Exceptions Thrown & Conditions**

- 3486 • See section 2.2.1.4.1 General Exceptions
- 3487 • `filterNotValid`: The Repository MUST throw this exception if [this-the](#) property [or rendition](#)
- 3488 filter input parameter is not valid.

3489 **2.2.4 Object Services**

3490 CMIS provides ID-based CRUD (**C**reate, **R**etrieve, **U**date, **D**elete), operations on objects in a Repository.

3491 **2.2.4.1 createDocument**

3492 **Description:** Creates a document object of the specified type (given by the `cmis:objectTypeId` property)

3493 in the (optionally) specified location.

3494 **2.2.4.1.1 Inputs**

3495 **Required:**

- 3496 **ID repositoryId:** The identifier for the Repository.
- 3497 **<Array> properties:** The property values that MUST be applied to the newly-created Document
- 3498 Object.

3499 **Optional:**

- 3500 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
- 3501 created Document Object.
 - 3502 ○ This parameter MUST be specified if the Repository does NOT support the optional
 - 3503 `"unfiling"` capability.
- 3504 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the newly-
- 3505 created Document Object. The method of passing the contentStream to the server and the
- 3506 encoding mechanism will be specified by each specific binding. MUST be required if the type
- 3507 requires it.
- 3508 • **Enum versioningState:** An enumeration specifying what the versioning state of the newly-created
- 3509 object MUST be. If the repository does not support versioning, the repository MUST ignore the
- 3510 versioningState parameter. Valid values are:
 - 3511 ○ `none`: The document MUST be created as a non-versionable document.
 - 3512 ○ `checkedout`: The document MUST be created in the checked-out state. The checked-
 - 3513 out document MAY be visible to other users.
 - 3514 ○ `major (default)`: The document MUST be created as a major version
 - 3515 ○ `minor`: The document MUST be created as a minor version.
- 3516 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
- 3517 object.
- 3518 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
- 3519 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.

- 3520
- 3521
- 3522
- **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created Document object, either using the ACL from folderId if specified, or being ignored if no folderId is specified.

3523

2.2.4.1.2 Outputs

3524

ID objectId: The ID of the newly-created document.

3525

2.2.4.1.3 Exceptions Thrown & Conditions

- 3526
- See section 2.2.1.4.1 General Exceptions
- 3527
- **constraint:** The Repository MUST throw this exception if ANY of the following conditions are met:
- 3528
- The cmis:objectId property value is not an Object-Type whose baseType is `"Document"`.
- 3529
- The cmis:objectId property value is NOT in the list of AllowedChildObjectTypes of

3530

 - the parent-folder specified by folderId.

3531

 - The value of any of the properties violates the min/max/required/length constraints

3532

 - specified in the property definition in the Object-Type.

3533

 - The `"contentStreamAllowed"` attribute of the Object-Type definition specified by the

3534

 - cmis:objectId property value is set to `"required"` and no contentStream input

3535

 - parameter is provided.

3536

 - The `"versionable"` attribute of the Object-Type definition specified by the

3537

 - cmis:objectId property value is set to FALSE and a value for the versioningState

3538

 - input parameter is provided that is something other than `"none"`.

3539

 - The `"versionable"` attribute of the Object-Type definition specified by the

3540

 - cmis:objectId property value is set to TRUE and the value for the versioningState

3541

 - input parameter is provided that is `"none"`.

3542

 - The `"controllablePolicy"` attribute of the Object-Type definition specified by the

3543

 - cmis:objectId property value is set to FALSE and at least one policy is provided.

3544

 - The `"controllableACL"` attribute of the Object-Type definition specified by the

3545

 - cmis:objectId property value is set to FALSE and at least one ACE is provided.

3546

 - At least one of the permissions is used in an ACE provided which is not supported by the

3547

 - repository.

3548

 - **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository

3549

 - detects a violation with the given cmis:name property value, the repository MAY throw this

3550

 - exception or chose a name which does not conflict.

3551

 - **storage:** See section 2.2.1.4.2 Specific Exceptions.

3552

 - **streamNotSupported:** The Repository MUST throw this exception if the

3553

 - `"contentStreamAllowed"` attribute of the Object-Type definition specified by the

3554

 - cmis:objectId property value is set to `"not allowed"` and a contentStream input parameter is

3555

 - provided.

3556

3557

2.2.4.2 createDocumentFromSource

3558

Description: Creates a document object as a copy of the given source document in the (optionally) specified location.

3559

3560

3561

2.2.4.2.1 Inputs

3562

Required:

3563

- **ID repositoryId:** The identifier for the Repository.

3564 | **ID sourceId:** The identifier for the source document.

3565 | **Optional:**

- 3566 | • **<Array> properties:** The property values that MUST be applied to the Object. This list of
3567 | properties SHOULD only contain properties whose values differ from the source document.
- 3568 | • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
3569 | created Document Object.
 - 3570 | ○ This parameter MUST be specified if the Repository does NOT support the optional
3571 | `"unfiling"` capability.
- 3572 | • **Enum versioningState:** An enumeration specifying what the versioning state of the newly-
3573 | created object MUST be. [If the repository does not support versioning, the repository MUST](#)
3574 | [ignore the versioningState parameter.](#) Valid values are:
 - 3575 | ○ `none`: The document MUST be created as a non-versionable document.
 - 3576 | ○ `checkedout`: The document MUST be created in the checked-out state.
 - 3577 | ○ `major (default)`: The document MUST be created as a major version
 - 3578 | ○ `minor`: The document MUST be created as a minor version.
- 3579 | • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
3580 | object.
- 3581 | • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
3582 | object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3583 | • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3584 | Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
3585 | specified.

3586 | 2.2.4.2.2 Outputs

3587 | **ID objectId:** The ID of the newly-created document.

3588 | 2.2.4.2.3 Exceptions Thrown & Conditions

- 3589 | • See section 2.2.1.4.1 General Exceptions
- 3590 | • `constraint`: The Repository MUST throw this exception if ANY of the following conditions are
3591 | met:
 - 3592 | ○ The sourceId is not an Object whose baseType is `"Document"`.
 - 3593 | ○ The source document's `cmis:objectId` property value is NOT in the list of
3594 | AllowedChildObjectTypes of the parent-folder specified by folderId.
 - 3595 | ○ The `"versionable"` attribute of the Object-Type definition specified by the
3596 | `cmis:objectId` property value is set to FALSE and a value for the versioningState
3597 | input parameter is provided that is something other than `"none"`.
 - 3598 | ○ The `"versionable"` attribute of the Object-Type definition specified by the
3599 | `cmis:objectId` property value is set to TRUE and the value for the versioningState
3600 | input parameter is provided that is `"none"`.
 - 3601 | ○ The `"controllablePolicy"` attribute of the Object-Type definition specified by the
3602 | `cmis:objectId` property value is set to FALSE and at least one policy is provided.
 - 3603 | ○ The `"controllableACL"` attribute of the Object-Type definition specified by the
3604 | `cmis:objectId` property value is set to FALSE and at least one ACE is provided.
 - 3605 | ○ At least one of the permissions is used in an ACE provided which is not supported by the
3606 | repository.

- 3607 • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. If the repository
3608 detects a violation with the given `cmis:name` property value, the repository MAY throw this
3609 exception or chose a name which does not conflict.
- 3610 • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 3611 • `streamNotSupported`: The Repository MUST throw this exception if the
3612 `"_contentStreamAllowed"` attribute of the Object-Type definition specified by the
3613 `cmis:objectTypeId` property value is set to `"_not allowed"` and a `contentStream` input parameter is
3614 provided.

3615 2.2.4.3 createFolder

3616 **Description:** Creates a folder object of the specified type in the specified location.

3617 2.2.4.3.1 Inputs

3618 Required:

- 3619 • **ID repositoryId:** The identifier for the Repository.
- 3620 • **<Array> properties:** The property values that MUST be applied to the newly-created Folder
3621 Object.
- 3622 • **ID folderId:** The identifier for the folder that MUST be the parent folder for the newly-created
3623 Folder Object.

3624 Optional:

- 3625 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Folder object.
- 3626 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Folder object,
3627 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3628 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3629 Folder object, either using the ACL from folderId if specified, or being ignored if no folderId is
3630 specified.

3631 2.2.4.3.2 Outputs

3632 **ID objectId:** The ID of the newly-created folder.

3633 2.2.4.3.3 Exceptions Thrown & Conditions

- 3634 • See section 2.2.1.4.1 General Exceptions
- 3635 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
3636 met:
 - 3637 ○ The `cmis:objectTypeId` property value is not an Object-Type whose baseType is
3638 `"_Folder"`.
 - 3639 ○ The value of any of the properties violates the min/max/required/length constraints
3640 specified in the property definition in the Object-Type.
 - 3641 ○ The `cmis:objectTypeId` property value is NOT in the list of AllowedChildObjectTypes of
3642 the parent-folder specified by folderId.
 - 3643 ○ The `"_controllablePolicy"` attribute of the Object-Type definition specified by the
3644 `cmis:objectTypeId` property value is set to FALSE and at least one policy is provided.
 - 3645 ○ The `"_controllableACL"` attribute of the Object-Type definition specified by the
3646 `cmis:objectTypeId` property value is set to FALSE and at least one ACE is provided.
 - 3647 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3648 repository.

- 3649 • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. If the repository
3650 detects a violation with the given `cmis:name` property value, the repository MAY throw this
3651 exception or chose a name which does not conflict.
- 3652 • `storage`: See section 2.2.1.4.2 Specific Exceptions.

3653 2.2.4.4 createRelationship

3654 **Description:** Creates a relationship object of the specified type

3655 2.2.4.4.1 Inputs

3656 **Required:**

- 3657 • **ID repositoryId:** The identifier for the Repository.
- 3658 • **<Array> properties:** The property values that MUST be applied to the newly-created Relationship
3659 Object.

3660 **Optional:**

- 3661 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Replationship
3662 object.
- 3663 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Relationship
3664 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
3665 **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3666 Relationship object, either using the ACL from folderId if specified, or being ignored if no folderId
3667 is specified.

3668 2.2.4.4.2 Outputs

- 3669 • **ID objectId:** The ID of the newly-created relationship.

3670 2.2.4.4.3 Exceptions Thrown & Conditions

- 3671 • See section 2.2.1.4.1 General Exceptions
- 3672 • `constraint`: The Repository MUST throw this exception if ANY of the following conditions are
3673 met:
 - 3674 ○ The `cmis:objectId` property value is not an Object-Type whose baseType is
3675 `"Relationship"`.
 - 3676 ○ The value of any of the properties violates the min/max/required/length constraints
3677 specified in the property definition in the Object-Type.
 - 3678 ○ The `sourceObjectId`'s Object-Type is not in the list of `"allowedSourceTypes"` specified by
3679 the Object-Type definition specified by `cmis:objectId` property value.
 - 3680 ○ The `targetObjectId`'s Object-Type is not in the list of `"allowedTargetTypes"` specified by
3681 the Object-Type definition specified by `cmis:objectId` property value.
 - 3682 ○ The `"controllablePolicy"` attribute of the Object-Type definition specified by the
3683 `cmis:objectId` property value is set to FALSE and at least one policy is provided.
 - 3684 ○ The `"controllableACL"` attribute of the Object-Type definition specified by the
3685 `cmis:objectId` property value is set to FALSE and at least one ACE is provided.
 - 3686 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3687 repository.
- 3688 • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. If the repository
3689 detects a violation with the given `cmis:name` property value, the repository MAY throw this
3690 exception or chose a name which does not conflict.
- 3691 • `storage`: See section 2.2.1.4.2 Specific Exceptions.

3692 **2.2.4.5 createPolicy**

3693 **Description:** Creates a policy object of the specified type

3694 **2.2.4.5.1 Inputs**

3695 **Required:**

- 3696 • **ID repositoryId:** The identifier for the Repository.
- 3697 • **<Array> properties:** The property values that MUST be applied to the newly-created Policy
- 3698 Object.

3699 **Optional:**

- 3700 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
- 3701 created Policy Object.
- 3702 • **unfiling:** This parameter MUST be specified if the Repository does NOT support the optional
- 3703 **unfiling** capability.
- 3704 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Policy object.
- 3705 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Policy object,
- 3706 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3707 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
- 3708 Policy object, either using the ACL from folderId if specified, or being ignored if no folderId is
- 3709 specified.

3710 **2.2.4.5.2 Outputs**

- 3711 • **ID objectId:** The ID of the newly-created Policy Object.

3712 **2.2.4.5.3 Exceptions Thrown & Conditions**

- 3713 • See section 2.2.1.4.1 General Exceptions
- 3714 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
- 3715 met:
 - 3716 • The cmis:objectId property value is not an Object-Type whose baseType is
 - 3717 **Policy**.
 - 3718 • The value of any of the properties violates the min/max/required/length constraints
 - 3719 specified in the property definition in the Object-Type.
 - 3720 • The cmis:objectId property value is NOT in the list of AllowedChildObjectTypes of
 - 3721 the parent-folder specified by folderId.
 - 3722 • The **controllablePolicy** attribute of the Object-Type definition specified by the
 - 3723 cmis:objectId property value is set to FALSE and at least one policy is provided.
 - 3724 • The **controllableACL** attribute of the Object-Type definition specified by the
 - 3725 cmis:objectId property value is set to FALSE and at least one ACE is provided.
 - 3726 • At least one of the permissions is used in an ACE provided which is not supported by the
 - 3727 repository.
- 3728 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository
- 3729 detects a violation with the given cmis:name property value, the repository MAY throw this
- 3730 exception or chose a name which does not conflict.
- 3731 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

3732 **2.2.4.6 getAllowableActions**

3733 **Description:** Gets the list of allowable actions for an Object (see section.2.2.1.2.6 Allowable Actions).

3734 **2.2.4.6.1 Inputs**

3735 **Required:**

- 3736 • **ID repositoryId:** The identifier for the Repository.
- 3737 • **ID objectId:** The identifier for the object

3738 **2.2.4.6.2 Outputs**

- 3739 • **<Array> AllowableActions:** see section 2.2.1.2.6 Allowable Actions.

3740 **2.2.4.6.3 Exceptions Thrown & Conditions**

3741 See section 2.2.1.4.1 General Exceptions

3742 **2.2.4.7 getObject**

3743 **Description:** Gets the specified information for the Object.

3744 **2.2.4.7.1 Inputs**

3745 **Required:**

- 3746 • **ID repositoryId:** The identifier for the Repository.
- 3747 • **ID objectId:** The identifier for the object

3748 **Optional:**

- 3749 • **String filter:** See section 2.2.1.2.1 Properties.
- 3750 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3751 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 3752 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3753 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 3754 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3755 **2.2.4.7.2 Outputs**

- 3756 • **<Array> Properties:** The list of properties for the object.
- 3757 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3758 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 3759 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3760 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 3761 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3762 **2.2.4.7.3 Exceptions Thrown & Conditions**

3763 See section 2.2.1.4.1 General Exceptions

3764 | **filterNotValid:** The Repository MUST throw this exception if the [eis](#) property [or rendition](#) filter
3765 | input parameter is not valid.

3766 **2.2.4.8 getProperties**

3767 **Description:** Gets the list of properties for an Object.

3768 **2.2.4.8.1 Inputs**

3769 **Required:**

- 3770 • **ID repositoryId:** The identifier for the Repository.
- 3771 • **ID objectId:** The identifier for the object

3772 **Optional:**

- 3773 • **String filter:** See section 2.2.1.2.1 Properties.

3774 **2.2.4.8.2 Outputs**

3775 **<Array> Properties:** The list of properties for the object.

3776 **2.2.4.8.3 Exceptions Thrown & Conditions**

3777 See section 2.2.1.4.1 General Exceptions

3778 `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter
3779 is not valid.

3780 **2.2.4.9 getObjectByPath**

3781 **Description:** Gets the specified object.

3782 **2.2.4.9.1 Inputs**

3783 **Required:**

- 3784 • **ID repositoryId:** The identifier for the Repository.
- 3785 • **String path:** The path to the object. See section 2.1.5.3 Paths.

3786 **Optional:**

- 3787 • **String filter:** See section 2.2.1.2.1 Properties.
- 3788 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3789 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3790 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3791 • **Boolean includePolicyIds:** See section 2.2.1.2.2 Relationships.
- 3792 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.

3793 **2.2.4.9.2 Outputs**

- 3794 • **<Array> Properties:** The list of properties for the object.
- 3795 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3796 **2.2.4.9.3 Exceptions Thrown & Conditions**

3797 • See section 2.2.1.4.1 General Exceptions

3798 • `filterNotValid`: The Repository MUST throw this exception if the `is` property `or rendition` filter
3799 input parameter is not valid.

3800 **2.2.4.10 getContentStream**

3801 **Description:** Gets the content stream for the specified Document object, or gets a rendition stream for a
3802 specified rendition of a document or folder object.

3803 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,
3804 in a manner appropriate to that protocol.

3805 **2.2.4.10.1 Inputs**

3806 **Required:**

- 3807 • **ID repositoryId:** The identifier for the Repository.
- 3808 • **ID objectId:** The identifier for the object

3809 **Optional:**

- 3810 • **ID streamId:** The identifier for the rendition stream, when used to get a rendition stream. For
3811 Documents, if not provided then this method returns the content stream. For Folders, it MUST be
3812 provided.

3813 **2.2.4.10.2 Outputs**

- 3814 • **<Stream> ContentStream:** The specified content stream or rendition stream for the object.

3815 **2.2.4.10.3 Exceptions Thrown & Conditions**

- 3816 • See section 2.2.1.4.1 General Exceptions
3817 `constraint:` The Repository MUST throw this exception if the object specified by objectId does
3818 NOT have a content stream or rendition stream.

3819 **2.2.4.11 getRenditions**

3820 **Description:** Gets the list of associated Renditions for the specified object. Only rendition attributes are
3821 returned, not rendition stream.

3822 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,
3823 in a manner appropriate to that protocol.

3824 **2.2.4.11.1 Inputs**

3825 **Required:**

- 3826 • **ID repositoryId:** The identifier for the Repository.
- 3827 • **ID objectId:** The identifier for the object

3828 **Optional:**

- 3829 • **String renditionFilter:** See Section 2.2.1.2.4
- 3830 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3831 • **Integer skipCount:** See section 2.2.1.1 Paging.

3832 **2.2.4.11.2 Outputs**

- 3833 • **<Array> Renditions:** The set of renditions available on this object

3834 **2.2.4.11.3 Exceptions Thrown & Conditions**

- 3835 • See section 2.2.1.4.1 General Exceptions
- 3836 • `notSupported:` The service method requires functionality that is not supported by the
3837 repository
- 3838 • `filterNotValid:` The [rendition](#) filter specified is not valid

3839 **2.2.4.12 updateProperties**

3840 **Description:** Updates properties of the specified object.

3841 **Notes:**

- 3842 • A Repository MAY automatically create new Document versions as part of an update properties
3843 operation. Therefore, the objectId output NEED NOT be identical to the objectId input.
- 3844 • Each CMIS protocol bindings MUST specify whether the updateProperties service MUST always
3845 include all updatable properties, or only those properties whose values are different than the
3846 original value of the object.

3847 2.2.4.12.1 Inputs

3848 Required:

- 3849 • **ID repositoryId:** The identifier for the Repository.
- 3850 **ID objectId:** The identifier of the object to be updated.
- 3851 • **<Array> properties:** The updated property values that MUST be applied to the Object.

3852 Optional:

- 3853 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3854 2.2.4.12.2 Outputs

- 3855 • **ID objectId:** The ID of the updated object.
- 3856 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3857 2.2.4.12.3 Exceptions Thrown & Conditions

- 3858 • See section 2.2.1.4.1 General Exceptions
- 3859 • **constraint:** The Repository MUST throw this exception if the value of any of the properties
3860 violates the min/max/required/length constraints specified in the property definition in the Object-
3861 Type.
- 3862 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY
3863 throw this exception or chose a name which does not conflict.
- 3864 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3865 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 3866 • **versioning:** The Repository MUST throw this exception if ANY of the following conditions are
3867 met:
- 3868 ○ The object is not checked out and ANY of the properties being updated are defined in
3869 their Object-Type definition have an attribute value of *Updatability* when checked-out.
 - 3870 ○ Additionally, the repository MAY throw this exception if the object is a non-current
3871 Document Version.

3872 2.2.4.13 moveObject

3873 **Description:** Moves the specified file-able object from one folder to another.

3874 2.2.4.13.1 Inputs

3875 Required:

- 3876 • **ID repositoryId:** The identifier for the Repository.
- 3877 • **ID objectId:** The identifier of the object to be moved.
- 3878 • **ID targetFolderId:** The folder into which the object is to be moved.
- 3879 • **ID sourceFolderId:** The folder from which the object is to be moved.

3880 **2.2.4.13.2 Outputs**

- 3881
- **ID objectId:** The identifier of the object to be moved.

3882 **2.2.4.13.3 Exceptions Thrown & Conditions**

- 3883
- See section 2.2.1.4.1 General Exceptions
- 3884
- **invalidArgument:** The Repository MUST throw this exception if the service is invoked with a missing sourceFolderId or the sourceFolderId doesn't match the specified object's parent folder (or one of the parent folders if the repository supports multifiling.).
- 3885
- 3886
- 3887
- **constraint:** The Repository MUST throw this exception if the cmis:objectId property value of the given object is NOT in the list of AllowedChildObjectTypes of the parent-folder specified by targetFolderId.
- 3888
- 3889
- 3890
- **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY throw this exception or chose a name which does not conflict.
- 3891
- 3892
- **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3893
- **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 3894
- **versioning:** The repository MAY throw this exception if the object is a non-current Document Version.
- 3895

3896 **2.2.4.14 deleteObject**

3897 **Description:** Deletes the specified object.

3898 **2.2.4.14.1 Inputs**

3899 **Required:**

- 3900
- **ID repositoryId:** The identifier for the Repository.
- 3901
- **ID objectId:** The identifier of the object to be deleted.

3902 **Optional:**

- 3903
- **Boolean allVersions:** If TRUE (default), then delete all versions of the document. If FALSE, delete only the document object specified. The Repository MUST ignore the value of this parameter when this service is invoke on a non-document object or non-versionable document object.
- 3904
- 3905
- 3906

3907

3908 **2.2.4.14.2 Exceptions Thrown & Conditions**

- 3909
- See section 2.2.1.4.1 General Exceptions
- 3910
- **constraint:** The Repository MUST throw this exception if the method is invoked on a Folder object that contains one or more objects.
- 3911
- **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 3912

3913 **2.2.4.15 deleteTree**

3914 **Description:** Deletes the specified folder object and all of its child- and descendant-objects.

3915 **Notes:**

- 3916
- A Repository MAY attempt to delete child- and descendant-objects of the specified folder in any order.
- 3917
- Any child- or descendant-object that the Repository cannot delete MUST persist in a valid state in the CMIS domain model.
- 3918
- 3919

- 3920
- This is not atomic.
- 3921
- However, if `deletesinglefiled` is chosen and some objects fail to delete, then single-filed objects are either deleted or kept, never just unfiled. This is so that a user can call this command again to recover from the error by using the same tree.
- 3922
- 3923

3924 2.2.4.15.1 Inputs

3925 Required:

- 3926
- **ID repositoryId:** The identifier for the Repository.
- 3927
- **ID folderId:** The identifier of the folder to be deleted.

3928 Optional:

- 3929
- **Boolean allVersions:** If `TRUE` (default), then delete all versions of the document. If `FALSE`, delete only the document object specified. The Repository **MUST** ignore the value of this parameter when this service is invoke on a non-document object or non-versionable document object.
- 3930
- 3931
- 3932
- **Enum unfileObjects:** An enumeration specifying how the repository **MUST** process file-able child- or descendant-objects. Valid values are:
- 3933
- `unfile:` Unfile all fileable objects.
- 3934
- `deletesinglefiled:` Delete all fileable non-folder objects whose only parent-folders are in the current folder tree. Unfile all other fileable non-folder objects from the current folder tree.
- 3935
- 3936
- `delete` (**default**): Delete all fileable objects.
- 3937
- **boolean continueOnFailure:** If `TRUE`, then the repository **SHOULD** continue attempting to perform this operation even if deletion of a child- or descendant-object in the specified folder cannot be deleted.
- 3938
- If `FALSE` (**default**), then the repository **SHOULD** abort this method when it fails to delete a single child- or descendant-object.
- 3939
- 3940
- 3941
- 3942

3943 2.2.4.15.2 Outputs

- 3944
- **<Array> ID failedToDelete:** A list of identifiers of objects in the folder tree that were not deleted.

3945 2.2.4.15.3 Exceptions Thrown & Conditions

- 3946
- See section 2.2.1.4.1 General Exceptions
- 3947
- `updateConflict:` See section 2.2.1.4.2 Specific Exceptions.

3948 2.2.4.16 setContentStream

3949 **Description:** Sets the content stream for the specified Document object.

3950 **Notes:** A Repository **MAY** automatically create new Document versions as part of this service method. Therefore, the `objectId` output **NEED NOT** be identical to the `objectId` input.

3951

3952 2.2.4.16.1 Inputs

3953 Required:

- 3954
- **ID repositoryId:** The identifier for the Repository.
- 3955
- **ID objectId:** The identifier for the Document object.
- 3956
- **<contentStream> contentStream:** The Content Stream

3957 Optional:

- 3958
- **Boolean overwriteFlag:** If `TRUE` (**default**), then the Repository **MUST** replace the existing content stream for the object (if any) with the input `contentStream`.
- 3959

3960 o If FALSE, then the Repository MUST only set the input contentStream for the object if the
3961 object currently does not have a content-stream.

3962 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3963 **2.2.4.16.2 Outputs**

3964 • **ID objectId:** The ID of the document.

3965 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3966 **2.2.4.16.3 Exceptions Thrown & Conditions**

3967 • See section 2.2.1.4.1 General Exceptions

3968 • **contentAlreadyExists:** The Repository MUST throw this exception if the input parameter
3969 **overwriteFlag** is FALSE and the Object already has a content-stream.

3970 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

3971 • **streamNotSupported:** The Repository MUST throw this exception if the
3972 **"_contentStreamAllowed"** attribute of the Object-Type definition specified by the
3973 **cmis:objectId** property value of the given document is set to **"_notallowed"**.

3974 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

3975 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
3976 Version.

3977 **2.2.4.17 deleteContentStream**

3978 **Description:** Deletes the content stream for the specified Document object.

3979 **Notes:** A Repository MAY automatically create new Document versions as part of this service method.
3980 Therefore, the objectId output NEED NOT be identical to the objectId input.

3981 **2.2.4.17.1 Inputs**

3982 **Required:**

3983 • **ID repositoryId:** The identifier for the Repository.

3984 • **ID objectId:** The identifier for the Document object.

3985 **Optional:**

3986 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3987 **2.2.4.17.2 Outputs**

3988 • **ID objectId:** The ID of the Document object.

3989 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3990 **2.2.4.17.3 Exceptions Thrown & Conditions**

3991 • See section 2.2.1.4.1 General Exceptions

3992 • **constraint:** The Repository MUST throw this exception if the Object's Object-Type definition
3993 **"_contentStreamAllowed"** attribute is set to **"_required"**.

3994 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

3995 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

3996 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
3997 Version.

3998 **2.2.5 Multi-filing Services**

3999 The Multi-filing services (*addObjectToFolder*, *removeObjectFromFolder*) are supported only if the
4000 repository supports the multifiling or unfileing [optional capabilities](#). The Multi-filing Services are used to
4001 file/un-file objects into/from folders.

4002 This service is NOT used to create or delete objects in the repository.

4003 **2.2.5.1 addObjectToFolder**

4004 **Description:** Adds an existing fileable non-folder object to a folder.

4005 **2.2.5.1.1 Inputs**

4006 **Required:**

- 4007 • **ID repositoryId:** The identifier for the Repository.
- 4008 • **ID objectId:** The identifier of the object.
- 4009 • **ID folderId:** The folder into which the object is to be filed.

4010 **Optional:**

- 4011 | **⊖1. Boolean allVersions:** Add all versions of the object to the folder if the repository supports
4012 version-specific filing. Defaults to TRUE.

4013 **2.2.5.1.2 Exceptions Thrown & Conditions**

- 4014 • See section 2.2.1.4.1 General Exceptions.
- 4015 • **constraint:** The Repository MUST throw this exception if the `cmis:objectId` property value
4016 of the given object is NOT in the list of `AllowedChildObjectTypes` of the parent-folder specified
4017 by `folderId`.

4018 **2.2.5.2 removeObjectFromFolder**

4019 **Description:** Removes an existing fileable non-folder object from a folder.

4020 **2.2.5.2.1 Inputs**

4021 **Required:**

- 4022 • **ID repositoryId:** The identifier for the Repository.
- 4023 • **ID objectId:** The identifier of the object.

4024 **Optional:**

- 4025 • **ID folderId:** The folder from which the object is to be removed.
 - 4026 ○ If no value is specified, then the Repository MUST remove the object from all folders in which
4027 it is currently filed.

4028 **2.2.5.2.2 Exceptions Thrown & Conditions**

- 4029 • See section 2.2.1.4.1 General Exceptions

4030 **2.2.6 Discovery Services**

4031 The Discovery Services (*query*) are used to search for query-able objects within the Repository.

4032 **2.2.6.1 query**

4033 **Description:** Executes a CMIS query statement against the contents of the Repository.

4034 2.2.6.1.1 Inputs

4035 Required:

- 4036 • **ID repositoryId:** The identifier for the Repository.
- 4037 • **String statement:** CMIS query to be executed. (See section 2.1.10 Query.)

4038 Optional:

- 4039 • **Boolean searchAllVersions:**
 - 4040 ○ If TRUE, then the Repository MUST include latest and non-latest versions of document
 - 4041 objects in the query search scope.
 - 4042 ○ If FALSE (**default**), then the Repository MUST only include latest versions of documents
 - 4043 in the query search scope.
 - 4044 ○ If the Repository does not support the optional `capabilityAllVersionsSearchable`
 - 4045 capability, then this parameter value MUST be set to FALSE.
- 4046 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
 - 4047 ○ Note: For query statements where the SELECT clause contains properties from only one
 - 4048 virtual table reference (i.e. referenced object-type), any value for this enum may be used.
 - 4049 If the SELECT clause contains properties from more than one table, then the value of this
 - 4050 parameter MUST be `"_none_"`.
- 4051 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
 - 4052 ○ If the SELECT clause contains properties from more than one table, then the value of this
 - 4053 parameter MUST not be set.
- 4054 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 4055 ○ Note: For query statements where the SELECT clause contains properties from only one
 - 4056 virtual table reference (i.e. referenced object-type), any value for this parameter may be
 - 4057 used. If the SELECT clause contains properties from more than one table, then the value
 - 4058 of this parameter MUST be `"_FALSE_"`.
- 4059 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 4060 • **Integer skipCount:** See section 2.2.1.1 Paging.

4061 2.2.6.1.2 Outputs

- 4062 • **<Array> Object QueryResults:** The set of results for the query. (See section 2.1.10 Query.).
- 4063 Each object result MUST include the following elements if they are requested:
 - 4064 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 4065 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 4066 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4067 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4068 Optional:

- 4069 • **Integer numItems:** See section 2.2.1.1 Paging.

4070

4071 2.2.6.1.3 Exceptions Thrown & Conditions

- 4072 • See section 2.2.1.4.1 General Exceptions
- 4073 • If the select clause includes properties from more than a single type reference, then the
- 4074 repository SHOULD throw an exception if `includeRelationships` is something other than `"_none_"`
- 4075 or `includeAllowableActions` is specified as TRUE.

4076 2.2.6.2 getContentChanges

4077 **Description:** Gets a list of content changes. This service is intended to be used by search crawlers or
4078 other applications that need to efficiently understand what has changed in the repository.

4079 **Notes:**

- 4080 • The content stream is NOT returned for any change event.
- 4081 • The definition of the authority needed to call this service is repository specific.
- 4082 • The latest change log token for a repository can be acquired via the getRepositoryInfo service.

4083 2.2.6.2.1 Inputs

4084 **Required:**

- 4085 • **ID repositoryId:** The identifier for the Repository.

4086 **Optional:**

- 4087 • **String changeLogToken:**

- 4088 ○ If specified, then the Repository MUST return the change event corresponding to the
4089 value of the specified change log token as the first result in the output.
- 4090 ○ If not specified, then the Repository MUST return the first change event recorded in the
4091 change log.

- 4092 • **Boolean includeProperties:**

- 4093 ○ If TRUE, then the Repository MUST include the updated property values for `"_updated"`
4094 change events if the repository supports returning property values as specified by
4095 `capabilityChanges`.
- 4096 ○ If FALSE (default), then the Repository MUST NOT include the updated property values
4097 for `"_updated"` change events. The single exception to this is that the `objectId` MUST
4098 always be included.

- 4099 • **Boolean includePolicyIds:**

4100 If TRUE, then the Repository MUST include the IDs of Policies applied to the object referenced in
4101 each change event, if the change event modified the set of policies applied to the object.

4102 If FALSE (default), then the Repository will not include policy information.

- 4103 • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the
4104 matched object if they exist on the matched object type definition and in the filter.
- 4105 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4106 • **Integer maxItems:** See section 2.2.1.1 Paging.

4107 2.2.6.2.2 Outputs

- 4108 • **<Array> changeEvents:** A collection of CMIS objects that MUST include the information as
4109 specified in 2.1.11.3. Each result MUST include the following elements if they are requested:

- 4110 ○ **<Array> policyIDs:** The IDs of Policies applied to the object referenced in the change
4111 event.
- 4112 ○ **<Array> ACLs:** The ACLs applied to the object reference in the change event.

- 4113 • **String latestChangeLogToken:** The change log token corresponding to the last change event in
4114 `changeEvents`.

- 4115 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4116 **Optional:**

- 4117 • **Integer numItems:** See section 2.2.1.1 Paging.

4118 2.2.6.2.3 Exceptions Thrown & Conditions

- 4119 • See section 2.2.1.4.1 General Exceptions
- 4120 • `constraint`: The Repository MUST throw this exception if the event corresponding to the
4121 change log token provided as an input parameter is no longer available in the change log. (E.g.
4122 because the change log was truncated).

4123 2.2.7 Versioning Services

4124 The Versioning services (`checkOut`, `cancelCheckOut`, `getPropertiesOfLatestVersion`, `getAllVersions`,
4125 `deleteAllVersions`) are used to navigate or update a Document Version Series.

4126 2.2.7.1 checkOut

4127 **Description:** Create a private working copy of the document.

4128 2.2.7.1.1 Inputs

4129 **Required:**

- 4130 • **ID repositoryId:** The identifier for the Repository.
- 4131 • **ID objectId:** The identifier of the document version.

4132 2.2.7.1.2 Outputs

- 4133 | • **ID objectId:** The identifier for the `"Private Working Copy"` document.
- 4134 | • **Boolean contentCopied:** TRUE if the content-stream of the Private Working Copy is a copy of
4135 | the contentStream of the Document that was checked out.
- 4136 | • **Boolean contentCopied:** FALSE if the content-stream of the Private Working Copy is `"not set"`.

4137 2.2.7.1.3 Exceptions Thrown & Conditions

- 4138 • See section 2.2.1.4.1 General Exceptions
- 4139 | • `constraint`: The Repository MUST throw this exception if the Document's Object-Type
4140 | definition's `versionable` attribute is FALSE.
- 4141 | • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 4142 | • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.
- 4143 | • `versioning`: The repository MAY throw this exception if the object is a non-current Document
4144 | Version.

4145 2.2.7.2 cancelCheckOut

4146 **Description:** Reverses the effect of a check-out. Removes the private working copy of the checked-out
4147 | document, allowing other documents in the version series to be checked out again. If the private working
4148 | copy has been created by `createDocument`, `cancelCheckOut` MUST delete the created document.

4149 2.2.7.2.1 Inputs

4150 **Required:**

- 4151 • **ID repositoryId:** The identifier for the Repository.
- 4152 • **ID objectId:** The identifier of the Private Working Copy.

4153 2.2.7.2.2 Exceptions Thrown & Conditions

- 4154 • See section 2.2.1.4.1 General Exceptions

- 4155 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type
4156 definition's *versionable* attribute is FALSE.
- 4157 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 4158 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
4159 Version.

4160 2.2.7.3 checkIn

4161 **Description:** Checks-in the Private Working Copy document.

4162 **Notes:**

- 4163 • For repositories that do NOT support the optional `"_capabilityPWCUpdatable"` *capability*, the
4164 *properties* and *contentStream* input parameters MUST be provided on the checkIn method for
4165 updates to happen as part of checkIn.
- 4166 • Each CMIS protocol bindings MUST specify whether the checkin service MUST always include all
4167 updatable properties, or only those properties whose values are different than the original value
4168 of the object.

4169 2.2.7.3.1 Inputs

4170 **Required:**

- 4171 • **ID repositoryId:** The identifier for the Repository.
- 4172 • **ID objectId:** The identifier of the document.

4173 **Optional:**

- 4174 • **Boolean major:** TRUE (**default**) if the checked-in Document Object MUST be a major version.
4175 ○ FALSE if the checked-in Document Object MUST NOT be a major version.
- 4176 • **<Array> properties:** The property values that MUST be applied to the checked-in Document
4177 Object.
- 4178 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the checked-in
4179 Document Object. The method of passing the contentStream to the server and the encoding
4180 mechanism will be specified by each specific binding.
- 4181 • **String checkinComment:** See section 2.1.9.5 Versioning Properties on Document Objects.
- 4182 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
4183 object.
- 4184 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
4185 object.
- 4186 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
4187 Document object.

4188 2.2.7.3.2 Outputs

4189 **ID objectId:** The ID of the checked-in document.

4190 2.2.7.3.3 Exceptions Thrown & Conditions

- 4191 • See section 2.2.1.4.1 General Exceptions
- 4192 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type
4193 definition's *versionable* attribute is FALSE.
- 4194 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 4195 • **streamNotSupported:** The Repository MUST throw this exception if the
4196 `"_contentStreamAllowed"` attribute of the Object-Type definition specified by the

- 4197 | cmis:objectTypeId property value is set to `"not allowed"` and a `contentStream` input parameter is
4198 provided.
4199 • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.

4200 2.2.7.4 getObjectOfLatestVersion

4201 **Description:** Get a the latest Document object in the Version Series.

4202 2.2.7.4.1 Inputs

4203 **Required:**

- 4204 • **ID repositoryId:** The identifier for the Repository.
- 4205 • **ID objectId:** The identifier for the Version Series.

4206 **Optional:**

- 4207 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major
4208 version object in the Version Series.
 - 4209 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
4210 major) version object in the Version Series.
- 4211 • **String filter:** See section 2.2.1.2.1 Properties.
- 4212 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4213 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 4214 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4215 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4216 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4217 2.2.7.4.2 Outputs

- 4218 • **<Array> Properties:** The list of properties for the object.
- 4219 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 4220 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 4221 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 4222 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 4223 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4224 2.2.7.4.3 Exceptions Thrown & Conditions

- 4225 • See section 2.2.1.4.1 General Exceptions
- 4226 | • `filterNotValid`: The Repository MUST throw this exception if the `eis` property [or rendition](#) filter
4227 input parameter is not valid.
- 4228 • `objectNotFound`: The Repository MUST throw this exception if the input parameter `major` is
4229 TRUE and the Version Series contains no major versions.

4230 2.2.7.5 getPropertiesOfLatestVersion

4231 **Description:** Get a subset of the properties for the latest Document Object in the Version Series.

4232 2.2.7.5.1 Inputs

4233 **Required:**

- 4234 • **ID repositoryId:** The identifier for the Repository.

- 4235 • **ID objectId:** The identifier for the Version Series.
- 4236 **Optional:**
- 4237 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major version
- 4238 object in the Version Series.
- 4239 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
- 4240 major) version object in the Version Series.
- 4241 • **String filter:** See section 2.2.1.2.1 Properties.

4242 2.2.7.5.2 Outputs

4243 **<Array> Properties:** The list of properties for the object.

4244 2.2.7.5.3 Exceptions Thrown & Conditions

- 4245 • See section 2.2.1.4.1 General Exceptions
- 4246 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
- 4247 parameter is not valid.
- 4248 • `objectNotFound`: The Repository MUST throw this exception if the input parameter major is
- 4249 TRUE and the Version Series contains no major versions.

4250 2.2.7.6 getAllVersions

4251 **Description:** Returns the list of all Document Objects in the specified Version Series, sorted by

4252 `cmis:creationDate` descending.

4253 Notes:

- 4254 • The result set for this operation MUST include the Private Working Copy, subject to caller's
- 4255 access privileges.

4256 2.2.7.6.1 Inputs

4257 Required:

- 4258 • **ID repositoryId:** The identifier for the Repository.
- 4259 • **ID objectId:** The identifier for the Version Series.

4260 Optional:

- 4261 • **String filter:** See section 2.2.1.2.1 Properties.
- 4262 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4263 2.2.7.6.2 Outputs

- 4264 • **<Array> ObjectResults:** A list of Document Objects in the specified Version Series. Each object
- 4265 result MUST include the following elements if they are requested:
- 4266 ○ **<Array> Properties:** The list of properties for the object.
- 4267 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4268

4269 2.2.7.6.3 Exceptions Thrown & Conditions

- 4270 • See section 2.2.1.4.1 General Exceptions
- 4271 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
- 4272 parameter is not valid.

4273 2.2.8 Relationship Services

4274 The Relationship Services (*getObjectRelationships*) are used to retrieve the dependent Relationship
4275 objects associated with an independent object.

4276 2.2.8.1 getObjectRelationships

4277 **Description:** Gets all or a subset of relationships associated with an independent object.

4278 2.2.8.1.1 Inputs

4279 **Required:**

- 4280 • **ID repositoryId:** The identifier for the Repository.
- 4281 • **ID objectId:** The identifier of the object.

4282

4283 **Optional:**

- 4284 • **Boolean includeSubRelationshipTypes:** If TRUE, then the Repository MUST return all
4285 relationships whose Object-Types are descendant-types of the given object's cmis:objectTypeId
4286 property value as well as relationships of the specified type.
 - 4287 ○ Default is FALSE
 - 4288 ○ If FALSE, then the Repository MUST only return relationships whose Object-Type is
4289 equivalent to the given object's cmis:objectTypeId property value.
- 4290 • **Enum relationshipDirection:** An enumeration specifying whether the Repository MUST
4291 return relationships where the specified Object is the source of the relationship, the target of the
4292 relationship, or both. Valid values are:
 - 4293 ○ **source: (default)** The Repository MUST return only relationship objects where the specified
4294 object is the source object.
 - 4295 ○ **target:** The Repository MUST return only relationship objects where the specified object is
4296 the target object.
 - 4297 ○ **either:** The Repository MUST return relationship objects where the specified object is
4298 either the source or the target object.
- 4299 • **ID typeId:** If specified, then the Repository MUST return only relationships whose Object-Type is
4300 of the type specified
 - 4301 ○ If not specified, then the repository MUST return Relationship objects of all types.
- 4302 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 4303 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 4304 • **String filter:** See section 2.2.1.2.1 Properties.
- 4305 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4306 2.2.8.1.2 Outputs

- 4307 • **<Array> Objects:** A list of the relationship objects. Each object result MUST include the following
4308 elements if they are requested:
 - 4309 ○ **<Array> Properties:** The list of properties for the object.
 - 4310 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4311 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4312 **Optional:**

- 4313 • **Integer numItems:** See section 2.2.1.1 Paging.

4314

4315 2.2.8.1.3 Exceptions Thrown & Conditions

- 4316 • See section 2.2.1.4.1 General Exceptions
- 4317 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
- 4318 parameter is not valid.

4319 2.2.9 Policy Services

4320 The Policy Services (*applyPolicy*, *removePolicy*, *getAppliedPolicies*) are used to apply or remove a policy
4321 object to a *controllablePolicy* object.

4322 2.2.9.1 applyPolicy

4323 **Description:** Applies a specified policy to an object.

4324 2.2.9.1.1 Inputs

4325 **Required:**

- 4326 • **ID repositoryId**: The identifier for the Repository.
- 4327 • **ID policyId**: The identifier for the Policy to be applied.
- 4328 • **ID objectId**: The identifier of the object.

4329 2.2.9.1.2 Exceptions Thrown & Conditions

4330 See section 2.2.1.4.1 General Exceptions

4331 `constraint`: The Repository MUST throw this exception if the specified object's Object-Type
4332 definition's attribute for *controllablePolicy* is FALSE.

4333 2.2.9.2 removePolicy

4334 **Description:** Removes a specified policy from an object.

4335 2.2.9.2.1 Inputs

4336 **Required:**

- 4337 • **ID repositoryId**: The identifier for the Repository.
- 4338 • **ID policyId**: The identifier for the Policy to be removed.
- 4339 • **ID objectId**: The identifier of the object.

4340 2.2.9.2.2 Exceptions Thrown & Conditions

4341 • See section 2.2.1.4.1 General Exceptions

4342 • `constraint`: The Repository MUST throw this exception if the specified object's Object-Type
4343 definition's attribute for *controllablePolicy* is FALSE.

4344 2.2.9.3 getAppliedPolicies

4345 **Description:** Gets the list of policies currently applied to the specified object.

4346 2.2.9.3.1 Inputs

4347 **Required:**

- 4348 • **ID repositoryId**: The identifier for the Repository.
- 4349 • **ID objectId**: The identifier of the object.

4350 **Optional:**
4351 **String filter:** See section 2.2.1.2.1 Properties.

4352 2.2.9.3.2 Outputs

4353 **<Array> Objects:** A list of Policy Objects.

4354 2.2.9.3.3 Exceptions Thrown & Conditions

- 4355 • See section 2.2.1.4.1 General Exceptions
- 4356 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
4357 parameter is not valid.

4358 2.2.10 ACL Services

4359 2.2.10.1 getACL

4360 **Description:** Get the ACL currently applied to the specified document or folder object.

4361 2.2.10.1.1 Inputs

4362 **Required:**

- 4363 • **ID repositoryId:** The identifier for the repository.
- 4364 • **ID objectId:** The identifier for the object

4365 **Optional:**

- 4366 • **Boolean onlyBasicPermissions:** See section 2.8 Access Control. The repository SHOULD
4367 make a best effort to fully express the native security applied to the object
 - 4368 ○ `TRUE`: (default value if not provided) indicates that the client requests that the returned
4369 ACL be expressed using only the CMIS Basic permissions.
 - 4370 ○ `FALSE`: indicates that the server may respond using either solely CMIS Basic
4371 permissions, or repository specific permissions or some combination of both.

4372 2.2.10.1.2 Outputs

- 4373 • **<Array> AccessControlEntryType:** The list of access control entries of the ACL for the object.

4374 **Optional:**

- 4375 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
4376 i.e. there are no other security constraints applied to this object. Not provided defaults to `FALSE`.

4377 2.2.10.1.3 Exceptions Thrown & Conditions

- 4378 • See section 2.2.1.4.1 General Exceptions

4379 2.2.10.1.4 Notes

4380 This service MUST be supported by a repository, if `getRepository` returns `capabilityACL=discover` or
4381 `=manage`.

4382 How an ACL for the object is computed is up to the repository. A client MUST NOT assume that the ACEs
4383 from the ACL as returned by this service can be applied via `applyACL`.

4384 2.2.10.2 applyACL

4385 **Description:** Adds or removes the given ACEs to or from the ACL of document or folder object.

4386 2.2.10.2.1 Inputs

4387 Required:

- 4388 • **ID repositoryId:** The identifier for the repository.
- 4389 • **ID objectId:** The identifier for the object

4390 Optional:

- 4391 • **<Array> AccessControlEntryType addACEs:** The ACEs to be added.
- 4392 • **<Array> AccessControlEntryType removeACEs:** The ACEs to be removed.
- 4393 • **Enum ACLPropagation:** Specifies how ACEs should be handled:
 - 4394 ○ **objectonly:** ACEs must be applied without changing the ACLs of other objects.
 - 4395 | ○ **propagate:** ACEs must be applied by propagate the changes to all **"inheriting"**
 - 4396 | objects.
 - 4397 ○ **repositorydetermined:** **Default value.** Indicates that the client leaves the behavior to
 - 4398 | the repository.

4399 2.2.10.2.2 Outputs

- 4400 • **<Array> AccessControlEntryType:** The list of access control entries of the resulting ACL for the
- 4401 | object

4402 Optional:

- 4403 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
- 4404 | i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.
- 4405 | ~~•String changeToken: See section 2.2.1.3 Change Tokens.~~

4406 2.2.10.2.3 Exceptions Thrown & Conditions

- 4407 • See section 2.2.1.4.1 General Exceptions
- 4408 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
- 4409 | met:
 - 4410 | ○ The specified object's Object-Type definition's attribute for *controllableACL* is FALSE.
 - 4411 | ○ The value for *ACLPropagation* does not match the values as returned via
 - 4412 | *getACLCapabilities*.
 - 4413 | ○ At least one of the specified values for *permission* in ANY of the ACEs does not match
 - 4414 | ANY of the *permissionNames* as returned by *getACLCapability* and is not a CMIS Basic
 - 4415 | *permission*

4416 2.2.10.2.4 Notes

4417 This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=manage*.

4418 How ACEs are added or removed to or from the object is up to the repository – with respect to the

4419 | *ACLPropagation* provided by the client. For **"shared"** ACEs (e.g. via inheritance), the repository MAY

4420 | merge the ACEs provided with the ACEs of the ACL already applied to the object (i.e. the ACEs provided

4421 | MAY not be completely added or removed from the effective ACL for the object).

4422

4423

4424 3 Restful AtomPub Binding

4425 3.1 Overview

4426 This binding is based upon the Atom (RFC4287) and Atom Publishing Protocol (RFC5023).
4427 Implementations of CMIS MUST be compliant with RFC4287 and RFC5023.

4428
4429 In this binding, the client interacts with the repository by acquiring the service document. The client will
4430 request the service document by the URI provided by the vendor. The client will then choose a CMIS
4431 collection, and then start accessing the repository by following the references in the returned documents.

4432
4433 This binding consists of a service document specifying at least CMIS service collections, atom collections,
4434 feeds and entry documents. CMIS extends the Atom and AtomPub documents utilizing the Atom and
4435 AtomPub extension mechanism. CMIS also leverages link tags to specify additional resources related to
4436 the requested resource.

4437
4438 When requesting a resource, optional parameters may be specified to change default behavior via query
4439 parameters.

4440 3.1.1 Namespaces

4441 This specification uses the following namespaces and prefixes when referring to xml or xml schema
4442 elements in the text or examples:

- 4443 • CMIS-Core: <http://docs.oasis-open.org/ns/cmisis/core/200908/>
 - 4444 ○ Prefix: cmis
- 4445 • CMIS-RestAtom: <http://docs.oasis-open.org/ns/cmisis/restatom/200908/>
 - 4446 ○ Prefix: cmisra
- 4447 • Atom : <http://www.w3.org/2005/Atom>
 - 4448 ○ Prefix: atom
- 4449 • AtomPub: <http://www.w3.org/2007/app>
 - 4450 ○ Prefix: app

4451

4452 3.1.2 Authentication

4453 Authentication SHOULD be handled by the transport protocol. Please see AtomPub (RFC5023) section
4454 14.

4455

4456 3.1.3 Response Formats

4457 The client can specify, in HTTP the Accept header, which formats are acceptable to the client. With this
4458 mechanism the client can chose which response format the CMIS implementation should respond with.
4459 The CMIS compliant implementation MUST support the appropriate Media Types specified in this
4460 document.

4461 **3.1.4 Optional Arguments**

4462 The binding supports adding optional parameters to CMIS resources to modify the default behavior.
4463 CMIS implementations **MUST** support arguments being specified as HTTP query string parameters.

4464 Names and valid values for HTTP query string parameters are as described in the appropriate CMIS
4465 Service descriptions [see CMIS Domain Model]. Valid values of enumeration types are also represented
4466 in the CMIS Core XML Schema

4467 **3.1.5 Errors and Exceptions**

4468 Exceptions **MUST** be mapped to the appropriate HTTP status code.

4469 Repositories **SHOULD** provide sufficient information in the body of the HTTP response for a user to
4470 determine corrective action.

4471 See Section 3.2.4 HTTP Status Codes for more information.

4472 **3.1.6 Renditions**

4473 Each Rendition included in a CMIS AtomPub response is represented as an Atom link with relationship
4474 alternate.

4475

4476 The following attributes **SHOULD** be included on the link element:

- 4477 • href: URI to the rendition content stream
- 4478 • type: The Media Type of the Rendition
- 4479 • cmisra:renditionKind: The Rendition Kind for the Rendition

4480

4481 The following attributes **MAY** be included

- 4482 • title: The Filename (or name property if object) of Rendition
- 4483 • length: The length of the rendition

4484 **3.1.7 Content Streams**

4485 The content stream for a document **SHOULD** be referenced by the content src attribute as well as the
4486 edit-media link relation.

4487 A CMIS Repository **MAY** use different URIs for both content src attribute and the edit-media link relation
4488 for the same content stream.

4489 The following attributes **SHOULD** be included on the link element:

- 4490 • href: URI to the content stream
- 4491 • type: The Media Type of the content stream

4492 **3.1.8 Paging of Feeds**

4493 For paging, please see the AtomPub RFC. CMIS leverages first, next, previous, and last link relations to
4494 express paging.

4495 If the repository can include the number of items (numItems in CMIS Domain Model) in a feed, then the
4496 repository **SHOULD** include the cmisra:numItems extension element in the feed.

4497 **3.1.9 Services not Exposed**






4498 The following services are not exposed in this binding:

- 4499 • getRenditions: This is exposed as part of getObject
- 4500 • getProperties: This is exposed as part of getObject

- 4501 • createDocumentFromSource: This is not exposed in this binding except as the client saving the
4502 resource and resubmitting it without the cmis:objectId.
- 4503 • Setting ACL on Create or Checkin operations
 - 4504 ○ This is currently not possible with the REST binding. The Create or Checkin operation
4505 must be performed first. Then the dependent resource, ACL, must be retrieved and
4506 updated.
- 4507 • setContentStream: This does not return the new object id and change token as specified by the
4508 domain model. This is not possible without introducing a new HTTP header.
- 4509 • deleteContentStream: This does not return the new object id and change token as specified by
4510 the domain model. This is not possible without introducing a new HTTP header.
- 4511 • checkOut: This does not return whether or not content was copied. This is not possible without
4512 introducing a new HTTP header.

4513 3.1.9.1 removePolicy

4514 This service is exposed from the domain model in the RESTful Atom Binding. However, it is not as
4515 straightforward. To remove a policy from an object, one must do:

- 4516  Get the object.
- 4517  Fetch the policies collection of the object.
- 4518  Walk through the feed and find the policy object where cmis:objectId == policy id to remove.
- 4519  Get the self lin of this policy object.
- 4520  Perform a DELETE on this URL.

4521
4522 This is also the only case in the RESTful Atom Binding where an URI in a collection (policies) is specific
4523 to that collection.

4524 3.2 HTTP

4525 3.2.1 Entity Tag

4526 CMIS changeTokens are represented as Entity Tags and follow HTTP's use of Entity Tags. CMIS server
4527 implementations SHOULD support Entity Tags. ChangeTokens are also provided as properties and
4528 SHOULD be provided when the object is included inside an atom entry or feed.

4529

4530 On updates, perform the following checks (HTTP & CMIS levels):

4531

4532 1. If If-Match header is sent by client, ETag value is pulled from HTTP header If-Match per
4533 RFC2616. The supplied ETag is compared against the ETag on the server. If the match fails,
4534 then status code 412 is used.

4535 2.
4536 If cmis:changeToken property is supplied by the client, compare the supplied and the
4537 cmis:changeToken on the server. If the comparison fails, then return status code 409 per CMIS.

4538
4539 4-3. If ETag and cmis:changeToken are both specified, the HTTP If-Match check should be performed
4540 first.

4541 3.2.2 HTTP Range

4542 Repositories MAY support HTTP Range requests on Content Streams.

4543 **3.2.3 HTTP OPTIONS Method**

4544 The repository MAY support the HTTP OPTIONS method on all the resources defined in this
4545 specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP
4546 methods specified for that resource in the Allow header.

4547 **3.2.4 HTTP Status Codes**

4548 Please see the HTTP specification for more information on the HTTP status codes. These are provided
4549 as guidance from the HTTP specification. If any conflict arises, the HTTP specification is authoritative.

4550 **3.2.4.1 General CMIS Exceptions**

4551 The following listing defines the HTTP status codes that repositories MUST return for the various common
4552 exceptions defined in CMIS Domain Model.

4553	CMIS Services Exception	HTTP Status Code
4554	invalidArgument	400
4555	objectNotFound	404
4556	permissionDenied	403
4557	notSupported	405
4558	runtime	500
4559	constraint	409
4560	filterNotValid	400
4561	streamNotSupported	403
4562	storage	500
4563	contentAlreadyExists	409
4564	versioning	409
4565	updateConflict	409
4566	nameConstraintViolation	409

4567

4568 **3.2.4.2 Notable HTTP Status Codes**

- 4569 • 415 Unsupported Media Type
 - 4570 | ○ When a document is POST_{ed} to a collection that does not support the media type of the
 - 4571 | document, this status code MUST be returned
- 4572 • 422 Unprocessable Entity (Defined in RFC4918 Section 11.2)
 - 4573 | ○ When a request has been POST_{ed} but cannot be processed, this status code MUST be
 - 4574 | returned

4575

4576 Please see RFC2616 Section 10 for more information.

4577

4578 **3.3 Media Types**

4579 CMIS introduces new media types for:

- 4580 • a CMIS Query document (application/cmismquery+xml)
- 4581 • a CMIS AllowableActions document (application/cmismallowableactions+xml)
- 4582 • an Atom Document (Entry or Feed) with any CMIS Markup (application/cmismatom+xml)

- 4583 • an Atom Feed Document with CMIS Hierarchy extensions (application/cmistree+xml)
- 4584 • a CMIS ACL Document (application/cmisacl+xml)

4585

4586 In addition to those media types specified by CMIS, CMIS also leverages these media types:

- 4587 • AtomPub Service (application/atomsvc+xml)
- 4588 • Atom Entry (application/atom+xml;type=entry)
- 4589 • Atom Feed (application/atom+xml;type=feed)

4590 3.3.1 CMIS Atom

4591 Media Type: application/cmisatom+xml

4592 Starting tag: atom:feed or atom:entry

4593 Type Parameters:

- 4594 • type – the semantics of the type parameter MUST be the same as the media type parameter for
- 4595 atom documents.

4596

4597 This allows clients to differentiate between repositories that require atom media type with CMIS
 4598 extensions (application/cmisatom+xml) for creation and repositories that allow generic atom media type
 4599 without CMIS extensions (application/atom+xml).

4600

4601 This is only used for CMIS repositories to advertise what media types are accepted for adding to a
 4602 collection (e.g., creating resources in a collection). As such CMIS does not require specifying whether an
 4603 atom feed has CMIS markup. It is included to be consistent with the Atom media type.

4604

4605 All feeds and entries from a CMIS repository MUST utilize the atom media type for exposing Atom
 4606 resources. Please see the individual resources for more information on the media type. This provides
 4607 the interoperability with Atom clients.

4608

4609 Example:

4610

```

4611 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4612 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
4613 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
4614 xmlns:atom="http://www.w3.org/2005/Atom"
4615 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4616 open.org/ns/cmisis/restatom/200908/"
4617 <atom:author>
4618 <atom:name>Al Brown</atom:name>
4619 </atom:author>
4620 <atom:id>urn:uuid:efe0542e-8933-4b3e-93f2-4d1caa3fc2d9</atom:id>
4621 <atom:title type="text">CMIS Example Document</atom:title>
4622 <atom:updated>2010-01-25T10:20:58.318-08:00</atom:updated>
4623 <atom:content type="text">some text</atom:content>
4624 <cmisra:object>
4625 <cmis:properties>
4626 <cmis:propertyId localName="rep-cmisis:objectTypeId"
4627 propertyDefinitionId="cmis:objectTypeId">
4628 <cmis:value>invoice</cmis:value>
4629 </cmis:propertyId>
4630 <cmis:propertyString localName="rep-cmisis:name"
4631 propertyDefinitionId="cmis:name">
4632 <cmis:value>CMIS Example Document</cmis:value>
4633 </cmis:propertyString>

```

```
4634     </cmis:properties>
4635     </cmisra:object>
4636 </atom:entry>
4637
```

4638 3.3.2 CMIS Query

4639 Media Type: application/cmismquery+xml

4640 Starting tag: cmis:query

4641

4642 This document contains the representation of a query to be executed in a CMIS repository.

4643

4644 Example:

```
4645 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4646 <cmis:query xmlns:cmis="http://docs.oasis-open.org/ns/cmism/core/200908/"
4647 xmlns:cmism="http://docs.oasis-open.org/ns/cmism/messaging/200908/"
4648 xmlns:atom="http://www.w3.org/2005/Atom"
4649 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4650 open.org/ns/cmism/restatom/200908/">
4651   <cmis:statement>SELECT * FROM cmis:document</cmis:statement>
4652   <cmis:searchAllVersions>true</cmis:searchAllVersions>
4653   <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
4654   <cmis:includeRelationships>none</cmis:includeRelationships>
4655   <cmis:renditionFilter>*</cmis:renditionFilter>
4656   <cmis:maxItems>50</cmis:maxItems>
4657   <cmis:skipCount>0</cmis:skipCount>
4658 </cmis:query>
```

4659

4660 Please also see the example documents included with the schema.

4661

4662 3.3.3 CMIS Allowable Actions

4663 Media Type: application/cmismallowableactions+xml

4664 Starting tag: cmis:allowableActions

4665

4666 This document contains the representation of the allowable actions the user may perform on the
4667 referenced object.

4668

4669 Example:

```
4670 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4671 <cmis:allowableActions xmlns:cmis="http://docs.oasis-
4672 open.org/ns/cmism/core/200908/" xmlns:cmism="http://docs.oasis-
4673 open.org/ns/cmism/messaging/200908/"
4674 xmlns:atom="http://www.w3.org/2005/Atom"
4675 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4676 open.org/ns/cmism/restatom/200908/">
4677   <cmis:canDeleteObject>true</cmis:canDeleteObject>
4678   <cmis:canUpdateProperties>true</cmis:canUpdateProperties>
4679   <cmis:canGetProperties>true</cmis:canGetProperties>
4680   <cmis:canGetObjectRelationships>true</cmis:canGetObjectRelationships>
4681   <cmis:canGetObjectParents>true</cmis:canGetObjectParents>
4682   <cmis:canMoveObject>true</cmis:canMoveObject>
4683   <cmis:canDeleteContentStream>true</cmis:canDeleteContentStream>
4684   <cmis:canCheckOut>true</cmis:canCheckOut>
4685   <cmis:canCancelCheckOut>true</cmis:canCancelCheckOut>
```

```

4686 <cmis:canCheckIn>true</cmis:canCheckIn>
4687 <cmis:canSetContentStream>true</cmis:canSetContentStream>
4688 <cmis:canGetAllVersions>true</cmis:canGetAllVersions>
4689 <cmis:canAddObjectToFolder>true</cmis:canAddObjectToFolder>
4690 <cmis:canRemoveObjectFromFolder>true</cmis:canRemoveObjectFromFolder>
4691 <cmis:canGetContentStream>true</cmis:canGetContentStream>
4692 <cmis:canApplyPolicy>true</cmis:canApplyPolicy>
4693 <cmis:canGetAppliedPolicies>true</cmis:canGetAppliedPolicies>
4694 <cmis:canRemovePolicy>true</cmis:canRemovePolicy>
4695 <cmis:canCreateDocument>true</cmis:canCreateDocument>
4696 </cmis:allowableActions>

```

4697
4698 Please also see the example documents included with the schema.
4699

4700 3.3.4 CMIS Tree

4701 Media Type: application/cmistree+xml

4702 Starting tag: atom:feed

4703
4704 This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.
4705

4706 Please see Section 3.5.2.1 for more information.

4707
4708 Example:

```

4709 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4710 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
4711 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
4712 xmlns:atom="http://www.w3.org/2005/Atom"
4713 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4714 open.org/ns/cmisis/restatom/200908/">
4715 <atom:title type="text">Feed for folder1</atom:title>
4716 <atom:author>
4717 <atom:name>Al Brown</atom:name>
4718 <atom:uri>http://www.ibm.com/</atom:uri>
4719 <atom:email>albertcbrown@us.ibm.com</atom:email>
4720 </atom:author>
4721 <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4722 <atom:id>urn:uuid:4a80905c-f774-4a9e-a57d-bf0dae5a796e</atom:id>
4723 <atom:link type="application/atom+xml;type=feed" rel="self"
4724 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
4725 41e0d92dfc71/3"/>
4726 <atom:link type="application/atomsvc+xml" rel="service"
4727 href="http://cmisexample.oasis-open.org/rep1//service"/>
4728 <atom:link type="application/atom+xml;type=entry" rel="via"
4729 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
4730 41e0d92dfc71"/>
4731 <atom:link type="application/atom+xml;type=feed"
4732 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
4733 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
4734 41e0d92dfc71/foldertree"/>
4735 <atom:link type="application/atom+xml;type=feed" rel="down"
4736 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
4737 41e0d92dfc71/children"/>
4738 <atom:link type="application/atom+xml;type=entry" rel="up"
4739 href="http://cmisexample.oasis-open.org/rep1/bb11830c-7d1e-4b0f-9ff2-
4740 af4857c49200"/>
4741 <atom:entry>

```

```

4742     <atom:author>
4743         <atom:name>Al Brown</atom:name>
4744         <atom:uri>http://www.ibm.com/</atom:uri>
4745         <atom:email>albertcbrown@us.ibm.com</atom:email>
4746     </atom:author>
4747     <atom:content src=""http://cmisexample.oasis-open.org/rep1/63a9c18c-
4748 5e31-4590-8462-86d181e345a4""/>
4749     <atom:id>urn:uuid:63a9c18c-5e31-4590-8462-86d181e345a4</atom:id>
4750     <atom:title type=""text"">CMIS Example Folder as Customer
4751 type</atom:title>
4752     <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4753     <atom:link rel=""self"" href=""http://cmisexample.oasis-
4754 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4""/>
4755     <atom:link rel=""edit"" href=""http://cmisexample.oasis-
4756 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4""/>
4757     <atom:link type=""application/cmisacl+xml;type=allowableActions""
4758 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions""
4759 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4760 86d181e345a4/allowableactions""/>
4761     <atom:link type=""application/atom+xml;type=entry""
4762 rel=""describedby"" href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-
4763 5e31-4590-8462-86d181e345a4/type""/>
4764     <atom:link type=""application/atomsvc+xml"" rel=""service""
4765 href=""http://cmisexample.oasis-open.org/rep1//service""/>
4766     <atom:published>2010-01-25T10:20:58.536-08:00</atom:published>
4767     <atom:summary type=""html"">HTML summary of Entry 63a9c18c-5e31-4590-
4768 8462-86d181e345a4</atom:summary>
4769     <atom:link type=""application/atom+xml;type=entry"" rel=""up""
4770 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4771 86d181e345a4/up""/>
4772     <atom:link type=""application/atom+xml;type=feed"" rel=""down""
4773 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4774 86d181e345a4/children""/>
4775     <atom:link type=""application/cmistree+xml"" rel=""down""
4776 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4777 86d181e345a4/tree""/>
4778     <atom:link type=""application/atom+xml;type=feed""
4779 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/foldertree""
4780 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4781 86d181e345a4/foldertree""/>
4782     <atom:link type=""application/atom+xml;type=feed""
4783 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/relationships""
4784 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4785 86d181e345a4/relationships""/>
4786     <atom:link type=""application/atom+xml;type=feed""
4787 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/policies""
4788 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4789 86d181e345a4/policies""/>
4790     <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
4791 open.org/ns/cmisacl/link/200908/acl"" href=""http://cmisexample.oasis-
4792 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4/acl""/>
4793     <cmisra:object>
4794         <cmis:properties>
4795             <cmis:propertyId localName=""rep-cmis:objectId""
4796 propertyDefinitionId=""cmis:objectId"">
4797                 <cmis:value>63a9c18c-5e31-4590-8462-
4798 86d181e345a4</cmis:value>
4799             </cmis:propertyId>
4800         </cmis:properties>
4801     </cmisra:object>
4802     <cmisra:pathSegment>customer</cmisra:pathSegment>
4803     <cmisra:children>
4804     <atom:feed>

```

```

4805 |         <atom:title type="text">CMIS Example Folder as Customer
4806 | type</atom:title>
4807 |         <atom:author>
4808 |             <atom:name>Al Brown</atom:name>
4809 |             <atom:uri>http://www.ibm.com/</atom:uri>
4810 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
4811 |         </atom:author>
4812 |         <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4813 |         <atom:id>urn:uuid:51b5c0cd-e473-4492-82b3-
4814 | 666fbf913cf0</atom:id>
4815 |         <atom:link type="application/atom+xml;type=feed"
4816 | rel="self" href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-
4817 | 8462-86d181e345a4/3"/>
4818 |         <atom:link type="application/atomsvc+xml" rel="service"
4819 | href="http://cmisexample.oasis-open.org/rep1//service"/>
4820 |         <atom:link type="application/atom+xml;type=entry"
4821 | rel="via" href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-
4822 | 8462-86d181e345a4"/>
4823 |         <atom:link type="application/atom+xml;type=feed"
4824 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
4825 | href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4826 | 86d181e345a4/foldertree"/>
4827 |         <atom:link type="application/atom+xml;type=feed"
4828 | rel="down" href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-
4829 | 8462-86d181e345a4/children"/>
4830 |         <atom:link type="application/atom+xml;type=entry" rel="up"
4831 | href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
4832 | 41e0d92dfc71"/>
4833 |         <atom:entry>
4834 |             <atom:author>
4835 |                 <atom:name>Al Brown</atom:name>
4836 |                 <atom:uri>http://www.ibm.com/</atom:uri>
4837 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
4838 |             </atom:author>
4839 |             <atom:content src="http://cmisexample.oasis-
4840 | open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4841 |             <atom:id>urn:uuid:20cb7e68-0a7e-46ea-87e0-
4842 | 09fb8d85286e</atom:id>
4843 |             <atom:title type="text">CMIS Example Doc as Invoice
4844 | type</atom:title>
4845 |             <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4846 |             <atom:link rel="self" href="http://cmisexample.oasis-
4847 | open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4848 |             <atom:link rel="edit" href="http://cmisexample.oasis-
4849 | open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4850 |             <atom:link
4851 | type="application/cmisis+xml;type=allowableActions" rel="http://docs.oasis-
4852 | open.org/ns/cmisis/link/200908/allowableactions"
4853 | href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4854 | 09fb8d85286e/allowableactions"/>
4855 |             <atom:link type="application/atom+xml;type=entry"
4856 | rel="describedby" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-
4857 | 0a7e-46ea-87e0-09fb8d85286e/type"/>
4858 |             <atom:link type="application/atomsvc+xml"
4859 | rel="service" href="http://cmisexample.oasis-open.org/rep1//service"/>
4860 |             <atom:published>2010-01-25T10:20:58.536-
4861 | 08:00</atom:published>
4862 |             <atom:summary type="html">HTML summary of Entry
4863 | 20cb7e68-0a7e-46ea-87e0-09fb8d85286e</atom:summary>
4864 |             <atom:link rel="edit-media"
4865 | href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4866 | 09fb8d85286e/edit-media"/>

```

```

4867         <atom:link rel="alternate"
4868 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4869 09fb8d85286e/alternate"/>
4870         <atom:link type="application/atom+xml;type=feed"
4871 rel="up" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-
4872 87e0-09fb8d85286e/parents"/>
4873         <atom:link type="application/atom+xml;type=feed"
4874 rel="version-history" href="http://cmisexample.oasis-
4875 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/allversions"/>
4876         <atom:link type="application/atom+xml;type=entry"
4877 rel="current-version" href="http://cmisexample.oasis-
4878 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/latest"/>
4879         <atom:link type="application/atom+xml;type=feed"
4880 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
4881 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4882 09fb8d85286e/relationships"/>
4883         <atom:link type="application/atom+xml;type=feed"
4884 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
4885 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4886 09fb8d85286e/policies"/>
4887         <atom:link type="application/cmisacl+xml"
4888 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/acl"
4889 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4890 09fb8d85286e/acl"/>
4891         <cmisra:object>
4892           <cmis:properties>
4893             <cmis:propertyId localName="rep-cmis:objectId"
4894 propertyDefinitionId="cmis:objectId">
4895 <cmis:value>20cb7e68-0a7e-46ea-87e0-09fb8d85286e</cmis:value>
4896           </cmis:propertyId>
4897         </cmis:properties>
4898       </cmisra:object>
4899       <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
4900     </atom:entry>
4901   </atom:feed>
4902 </cmisra:children>
4903 </atom:entry>
4904 <atom:entry>
4905   <atom:author>
4906     <atom:name>Al Brown</atom:name>
4907     <atom:uri>http://www.ibm.com/</atom:uri>
4908     <atom:email>albertcbrown@us.ibm.com</atom:email>
4909   </atom:author>
4910   <atom:content src="http://cmisexample.oasis-open.org/rep1/1de1d476-
4911 11fb-47bf-b136-8a8d0b4b030a"/>
4912   <atom:id>urn:uuid:1de1d476-11fb-47bf-b136-8a8d0b4b030a</atom:id>
4913   <atom:title type="text">CMIS Example Doc as Invoice
4914 type</atom:title>
4915   <atom:updated>2010-01-25T10:20:58.568-08:00</atom:updated>
4916   <atom:link rel="self" href="http://cmisexample.oasis-
4917 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a"/>
4918   <atom:link rel="edit" href="http://cmisexample.oasis-
4919 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a"/>
4920   <atom:link type="application/cmisis+xml;type=allowableActions"
4921 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
4922 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4923 8a8d0b4b030a/allowableactions"/>
4924   <atom:link type="application/atom+xml;type=entry"
4925 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/1de1d476-
4926 11fb-47bf-b136-8a8d0b4b030a/type"/>
4927   <atom:link type="application/atomsvc+xml" rel="service"
4928 href="http://cmisexample.oasis-open.org/rep1//service"/>
4929   <atom:published>2010-01-25T10:20:58.568-08:00</atom:published>

```



```

4930 | <atom:summary type="html">HTML summary of Entry 1de1d476-11fb-47bf-
4931 | b136-8a8d0b4b030a</atom:summary>
4932 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
4933 | open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/edit-media"/>
4934 | <atom:link rel="alternate" href="http://cmisexample.oasis-
4935 | open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/alternate"/>
4936 | <atom:link type="application/atom+xml;type=feed" rel="up"
4937 | href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4938 | 8a8d0b4b030a/parents"/>
4939 | <atom:link type="application/atom+xml;type=feed" rel="version-
4940 | history" href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-
4941 | b136-8a8d0b4b030a/allversions"/>
4942 | <atom:link type="application/atom+xml;type=entry" rel="current-
4943 | version" href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-
4944 | b136-8a8d0b4b030a/latest"/>
4945 | <atom:link type="application/atom+xml;type=feed"
4946 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
4947 | href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4948 | 8a8d0b4b030a/relationships"/>
4949 | <atom:link type="application/atom+xml;type=feed"
4950 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
4951 | href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4952 | 8a8d0b4b030a/policies"/>
4953 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
4954 | open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
4955 | open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/acl"/>
4956 | <cmisra:object>
4957 | <cmis:properties>
4958 | <cmis:propertyId localName="rep-cmis:objectId"
4959 | propertyDefinitionId="cmis:objectId">
4960 | <cmis:value>1de1d476-11fb-47bf-b136-
4961 | 8a8d0b4b030a</cmis:value>
4962 | </cmis:propertyId>
4963 | </cmis:properties>
4964 | </cmisra:object>
4965 | <cmisra:pathSegment>invoice3.pdf</cmisra:pathSegment>
4966 | </atom:entry>
4967 | </atom:feed>

```

4968

4969 | Note: This media type is used on links with relation down (see section 3.4.3.2 [Hierarchy Navigation](#)
4970 | [Internet Draft Link Relations Hierarchy Navigation Internet Draft Link Relations](#)). When the individual
4971 | resources are returned by the CMIS repository they will use the atom media type (application/atom+xml)

4972

4973 | Please also see the example documents included with the schema.

4974

4975 | 3.3.5 CMIS ACL

4976 | Media Type: application/cmisacl+xml
4977 | Starting tag: cmis:acl

4978

4979 | This document specifies an Access Control List based on the schema in CMIS Domain Model.

4980

4981 | Example:

```

4982 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

```
4983 | <cmis:acl xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4984 | xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4985 | xmlns:atom="http://www.w3.org/2005/Atom"
4986 | xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4987 | open.org/ns/cmis/restatom/200908/">
4988 |   <cmis:permission>
4989 |     <cmis:principal>
4990 |       <cmis:principalId>Al Brown</cmis:principalId>
4991 |     </cmis:principal>
4992 |     <cmis:permission>cmis:read</cmis:permission>
4993 |     <cmis:permission>cmis:write</cmis:permission>
4994 |     <cmis:permission>cmis:all</cmis:permission>
4995 |     <cmis:permission>publish</cmis:permission>
4996 |     <cmis:direct>true</cmis:direct>
4997 |   </cmis:permission>
4998 | </cmis:acl>
```

4999
5000 Please also see the example documents included with the schema.

5001

5002 **3.4 Atom Extensions for CMIS**

5003 **3.4.1 Atom Element Extensions**

5004 **3.4.1.1 AtomPub Workspace**

5005 **3.4.1.1.1 cmisra:collectionType**

5006 This element is included inside the app:collection element. This specifies the cmis collection type.

5007 **3.4.1.1.2 cmisra:repositoryInfo**

5008 This element is included inside the app:workspace element. This specifies information about the CMIS
5009 repository.

5010 **3.4.1.1.3 cmis:uritemplate**

5011 This element is included inside the app:workspace element. This specifies information about URI
5012 templates

5013 **3.4.1.2 Atom Feed**

5014 **3.4.1.2.1 cmisra:numItems**

5015 This element is included inside the atom:feed element. This specifies the number of items in the feed.

5016 **3.4.1.3 Atom Entry**

5017 **3.4.1.3.1 cmisra:children**

5018 This element is included inside the atom:entry element. This includes the children of the atom entry. This
5019 element MUST include an atom:feed element.

5020

5021 **3.4.1.3.2 cmisra:object**

5022 This element is included inside the atom:entry element for CMIS Document, Folder, Relationship and
5023 Policy objects. This specifies the CMIS object information for the atom entry.

5024

5025 **3.4.1.3.3 cmisra:pathSegment**

5026 This element is included inside the atom:entry element for CMIS Type Definitions that are filable. This
5027 specifies the pathSegment for this object in the folder representing the feed.

5028

5029 **3.4.1.3.4 cmisra:relativePathSegment**

5030 This element is included inside the atom:entry element. This specifies the relative pathSegment for the
5031 object in that particular folder. This MUST be used only inside an object parents feed.

5032 **3.4.1.3.5 cmisra:type**

5033 This element is included inside the atom:entry element for CMIS Type Definitions. This specifies the type
5034 definition the atom entry represents.

5035 **3.4.1.3.6 cmisra:content**

5036 This element specifies the content of the atom:entry element. The content is base64 encoded in the
5037 base64 element. The elements of a cmisra:content element are:

- 5038 • mediaType: This contains the media type of the content as described by RFC4288.
- 5039 • base64: This contains the base64 content of the file

5040

5041 This element MUST take precedence over atom:content on submission of an atom entry to a repository.

5042

5043 A repository MUST use the atom:content element to return back to the client the content of the document.

5044

5045 Note: This is required when the client has an XML document stored that is might not be well formed and
5046 thus would not be able to be included inside atom:content element.

5047

5048 **3.4.2 Attributes**

5049 These attributes are in the CMIS RestAtom namespace (cmisra).

5050 **3.4.2.1 cmisra:id**

5051 This attribute is used on the atom:link element to specify the cmis id of the resource. This attribute
5052 SHOULD be on all link relations that point to a CMIS object.

5053

5054 This attribute MAY also be on cmisra:type. The value of the attribute on cmis:type MUST be the same as
5055 the type definition id.

5056

5057 Example:

5058 | `<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`

```
5059 | <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
5060 | xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
5061 | xmlns:atom="http://www.w3.org/2005/Atom"
5062 | xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5063 | open.org/ns/cmisis/restatom/200908/" type="application/atom+xml;type=feed"
5064 | rel="down" href="http://cmisexample.oasis-open.org/rep1//children/e170da7d-
5065 | d322-472d-b1eb-67bdb1ec18ca/1" cmisra:id="e170da7d-d322-472d-b1eb-
5066 | 67bdb1ec18ca"/>
```

5067

5068 Please also see the example documents included with the schema.

5069

5070 3.4.2.2 cmisra:renditionKind

5071 This attribute is used on the atom:link element with relation alternate to specify the renditionKind of the
5072 resource. This attribute SHOULD be on all link elements with relation alternate that are a CMIS rendition.

5073

5074 Example:

```
5075 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5076 | <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
5077 | xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
5078 | xmlns:atom="http://www.w3.org/2005/Atom"
5079 | xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5080 | open.org/ns/cmisis/restatom/200908/" type="text/html" rel="alternate"
5081 | href="http://cmisexample.oasis-open.org/rep1//rendition/e170da7d-d322-472d-
5082 | b1eb-67bdb1ec18ca/1" cmisra:renditionKind="cmis:thumbnail"/>
```

5083

5084 Please also see the example documents included with the schema.

5085 3.4.3 CMIS Link Relations

5086 The listing below outlines the different link relation types in CMIS. This is in addition to the link relations
5087 specified by Atom and Atom Publishing Protocol. The registry for link relations is located at
5088 <http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

5089

5090 The link element with a specified relation MUST be included if client can perform the operation. The
5091 repository SHOULD omit the link relation if the operation is not available. The operation may not be
5092 available due to a variety of reasons such as access control, administrative policies, or other
5093 mechanisms.

5094

5095 Links may have the following attribute in addition to the ones specified by Atom and Atom Publishing
5096 Protocol:

- 5097 • (CMIS) id: Specifies the CMIS ID of the resource referenced by the link. Repositories SHOULD
5098 include this attribute for elements such as atom:link that point to CMIS resources that have an id.

5099

5100 These are the link relation types specified by CMIS:

5101 3.4.3.1 Existing Link Relations

5102 Existing link relations should be used where appropriate by the implementation. In addition, the following
5103 link relations are leveraged for the CMIS specification:

- 5104 • self
- 5105 ○ This link relation provides the URI to retrieve this resource again.

- 5106 ○ Service: The appropriate service that generated the atom entry or feed.
- 5107 ○ Resources: All except AllowableActions, ACL and Content Streams
- 5108 ● service
 - 5109 ○ The service link relation when provided on a CMIS resource MUST point to an AtomPub service document with only one workspace element. This workspace element MUST
 - 5110 represent the repository containing that resource.
 - 5111
 - 5112 ○ Media Type: application/atomsvc+xml
 - 5113 ○ Resources: All except AllowableActions, ACL and Content Streams
- 5114 ● describedby
 - 5115 ○ When used on a CMIS resource, this link relation MUST point to an atom entry that
 - 5116 describes the type of that resource.
 - 5117 ○ Service: getTypeDefinition on specified object
 - 5118 ○ Media Type: application/atom+xml;type=entry
 - 5119 ○ Resources: CMIS Document, CMIS Folder, CMIS Relationship, CMIS Policy objects and
 - 5120 CMIS Types
- 5121 ● via
 - 5122 ○ When used on an Atom Feed document, this link relation MUST point to the atom entry
 - 5123 representing the CMIS resource from whom this feed is derived.
 - 5124 ○ Media Type: application/atom+xml;type=entry
 - 5125 ○ Resources: All CMIS Feeds and Collections
- 5126 ● edit-media
 - 5127 ○ When used on a CMIS document resource, this link relation MUST point to the URI for
 - 5128 content stream of the CMIS document. This URI MUST be used to set or delete the
 - 5129 content stream. This URI MAY be used to retrieve the content stream for the document.
 - 5130 ○ Service: setContentStream (PUT) , deleteContentStream (DELETE)
 - 5131 ○ Media Type: Specific to resource
 - 5132 ○ Resources: CMIS Document
- 5133 ● edit
 - 5134 ○ When used on a CMIS resource, this link relation MUST provide an URI that can be used
 - 5135 with the HTTP PUT method to modify the atom:entry for the CMIS resource
 - 5136 ○ Service: getObject (GET), updateProperties (PUT)
 - 5137 ○ Media Type: application/atom+xml;type=entry
 - 5138 ○ Resources: CMIS Documents, CMIS Folders, CMIS Relationships and CMIS Policies
- 5139 ● alternate
 - 5140 ○ This is used to express Renditions on a CMIS resource. See section 3.1.6 Renditions.
 - 5141 ○ Service: getContentStream for specified rendition
 - 5142 ○ Resources: CMIS Document, CMIS Folder and CMIS Policies
- 5143 ● first
 - 5144 ○ This is used for Paging. Please see the AtomPub specification.
 - 5145 ○ Media Type: application/atom+xml;type=feed
 - 5146 ○ Resources: All Feeds
- 5147 ● previous
 - 5148 ○ This is used for Paging. Please see the AtomPub specification.
 - 5149 ○ Media Type: application/atom+xml;type=feed

- 5150 ○ Resources: All Feeds
- 5151 ● next
 - 5152 ○ This is used for Paging. Please see the AtomPub specification.
 - 5153 ○ Media Type: application/atom+xml;type=feed
 - 5154 ○ Resources: All Feeds
- 5155 ● last
 - 5156 ○ This is used for Paging. Please see the AtomPub specification.
 - 5157 ○ Media Type: application/atom+xml;type=feed
 - 5158 ○ Resources: All Feeds

5159
5160

5161 Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on
5162 these link relations.

5163 3.4.3.2 Hierarchy Navigation Internet Draft Link Relations

5164 CMIS leverages the following link relations:

- 5165 ● up
 - 5166 ○ Service: getFolderParent, getObjectParents, getTypeDefinition, getObject
 - 5167 ○ Media Type: application/atom+xml;type=feed, application/atom+xml;type=entry
 - 5168 ○ Resources: CMIS Document, CMIS Folder, CMIS Type Definitions, CMIS Folder
 - 5169 Children, CMIS Folder Descendants, CMIS FolderTree, CMIS Type Children, CMIS Type
 - 5170 Descendants
 - 5171 ■ This link relation is not included on CMIS Base Type Definitions or the CMIS
 - 5172 Root Folder
- 5173 ● down
 - 5174 ○ Service: getChildren, getDescendants, getTypeChildren, getTypeDescendants
 - 5175 ○ Media Type:
 - 5176 ■ For children: application/atom+xml;type=feed
 - 5177 ■ For descendants: application/cmistree+xml
 - 5178 ● The descendants feed resource when retrieved from the CMIS repository
 - 5179 will use the Atom Feed Media Type (application/atom+xml;type=feed)
 - 5180 ○ Resources: CMIS Folder, Type

5181 3.4.3.3 Versioning Internet Draft Link Relations

5182 CMIS leverages the following link relations from the Internet Draft:

- 5183 ● version-history
 - 5184 ○ Service: getAllVersions
 - 5185 ○ Media Type: application/atom+xml;type=feed
 - 5186 ○ Resources: CMIS Document
- 5187 ● current-version
 - 5188 ○ Service: getObjectForLatestVersion
 - 5189 ○ Media Type: application/atom+xml;type=entry
 - 5190 ○ Resources: CMIS Document
- 5191 ● working-copy

- 5192 ○ Service: getObject for private-working-copy specified by
- 5193 cmis:versionSeriesCheckedOutId property
- 5194 ○ Media Type: application/atom+xml;type=entry
- 5195 ○ Resources: CMIS Document

5196 3.4.3.4 CMIS Specific Link Relations

5197 CMIS defines the following link relations:

- 5198 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>
 - 5199 ▪ This link relation MUST point to a resource containing a CMIS AllowableActions
 - 5200 document for the CMIS resource containing this link relation.
 - 5201 ▪ Service: getAllowableActions
 - 5202 ▪ Media Type: application/cmisallowableactions+xml
 - 5203 ▪ Resources: CMIS Documents, CMIS Folders, CMIS Policies, and CMIS
 - 5204 Relationships
- 5205 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>
 - 5206 ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS
 - 5207 relationship resources for the CMIS resource containing this link relation.
 - 5208 ▪ Service: getObjectRelationships
 - 5209 ▪ Media Type: application/atom+xml;type=feed
 - 5210 ▪ Resources: CMIS Documents, CMIS Folders, and CMIS Policies
- 5211 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/source>
 - 5212 ▪ When used on a CMIS Relationship resource, this link relation MUST point to an
 - 5213 atom entry document for the CMIS Resource specified by the cmis:sourceId
 - 5214 property on the relationship.
 - 5215 ▪ Source Link on Relationship
 - 5216 ▪ Media Type: application/atom+xml;type=entry
 - 5217 ▪ Resources: CMIS Relationships
- 5218 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/target>
 - 5219 ▪ When used on a CMIS Relationship resource, this link relation MUST point to an
 - 5220 atom entry document for the CMIS Resource specified by the cmis:targetId
 - 5221 property on the relationship.
 - 5222 ▪ Target Link on Relationship
 - 5223 ▪ Media Type: application/atom+xml;type=entry
 - 5224 ▪ Resources: CMIS Relationships
- 5225 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/policies>
 - 5226 ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS
 - 5227 Policy resources for the CMIS resource containing this link relation.
 - 5228 ▪ Service: getAppliedPolicies
 - 5229 ▪ Media Type: application/atom+xml;type=feed
 - 5230 ▪ Resources: CMIS Documents and CMIS Folders
- 5231 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/acl>
 - 5232 ▪ This link relation MUST point to a resource containing a CMIS ACL document for
 - 5233 the CMIS resource containing this link relation.
 - 5234 ▪ Service: getACL
 - 5235 ▪ Media Type: application/cmisacl+xml

- 5236 ▪ Resources: CMIS Documents, CMIS Folders, CMIS Relationships, and CMIS
- 5237 Policies that are securable
- 5238 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/changes>
- 5239 ▪ This link relation MUST point to an Atom Feed containing the set of changes
- 5240 ▪ Service: getContentChanges
- 5241 ▪ Media Type: application/atom+xml;type=feed
- 5242 ▪ Resources: AtomPub Workspace Element in Service Document
- 5243 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>
- 5244 ▪ Used in AtomPub Service Document to identify the folder tree for a specified
- 5245 folder
- 5246 ▪ Service: getFolderTree
- 5247 ▪ Media Type: application/atom+xml;type=feed
- 5248 ▪ Resources: CMIS Folder, also used in AtomPub Service Document for root folder
- 5249 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>
- 5250 ▪ Used in AtomPub Service Document to identify the base types descendants
- 5251 ▪ Service: getTypeDescendants
- 5252 ▪ Media Type: application/atom+xml;type=feed
- 5253 ▪ Resources: AtomPub Workspace Element in Service Document
- 5254 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>
- 5255 ▪ Used in AtomPub Service Document to identify the root folder descendants
- 5256 ▪ Service: getDescendants for root folder
- 5257 ▪ Media Type: application/atom+xml;type=feed
- 5258 ▪ Resources: AtomPub Workspace Element in Service Document
- 5259

5260 3.5 Atom Resources

5261 For all Atom Resources used in this specification, the following MUST be followed:

5262 3.5.1 Feeds

5263 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmisis objects:

- 5264 • atom:updated SHOULD be the latest time the folder or its contents was updated. If unknown by
- 5265 the underlying repository, it MUST be the current time.
- 5266 • atom:author/atom:name MUST be the CMIS property cmisis:createdBy
- 5267 • atom:title MUST be the CMIS property cmisis:name
- 5268 • The atom:link with relation self MUST be generated to return the URI of the feed. If paging or any
- 5269 other mechanism is used to filter, sort, or change the representation of the feed, the URI MUST
- 5270 point back a resource with the same representation.
- 5271 • A feed SHOULD contain the element app:collection, describing the appropriate media types
- 5272 supported for creation of new entries in the feed
- 5273 • atom:id SHOULD be derived from cmisis:objectId. This id MUST be compliant with atom:ids
- 5274 specification and be a valid URI.
- 5275 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
- 5276 contained in the feed, then a link with the relation next MUST be included in the feed.
- 5277

- 5278 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis types:
- 5279 • atom:updated SHOULD be the latest time type definition was updated. If unknown by the
 - 5280 underlying repository, it MUST be the current time.
 - 5281 • atom:author/atom:name is repository specific
 - 5282 • atom:title MUST be the displayName attribute of the CMIS Type Definition.
 - 5283 • The atom:link with relation self MUST be generated to return the URI of the feed
 - 5284 • atom:id SHOULD be derived from the id attribute of the CMIS Type Definition. This id MUST be
 - 5285 compliant with atom:s specification and be a valid URI.
 - 5286 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
 - 5287 contained in the feed, then a link with the relation next MUST be included in the feed.

5288

5289 If on the root type, all fields are repository specific.

5290

5291 Ordering of entries in a feed is repository-specific if orderBy argument is not specified. If orderBy

5292 argument is specified, the order of the entries in the feed SHOULD conform to the ordering specified by

5293 the orderBy argument.

5294

5295 Note: Please see feedvalidator.org to validate Atom compliance.

5296 3.5.2 Entries

5297 At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry

5298 document and conform to the guidelines below for a cmis object:

- 5299 • atom:title MUST be the cmis:name property
- 5300 • app:edited MUST be cmis:~~lastModificationDate~~~~lastModifiedDate~~
- 5301 • atom:updated MUST be cmis:~~lastModificationDate~~~~lastModifiedDate~~
- 5302 • atom:published MUST be cmis:~~creationDate~~~~createdDate~~
- 5303 • atom:author/atom:name MUST be cmis:createdBy
- 5304 • All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are
- 5305 duplicated in an atom element
- 5306 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom:s
- 5307 specification and be a valid ~~URI~~~~URI~~.
- 5308 • The repository SHOULD populate the atom:summary tag with text that best represents a
- 5309 summary of the object. For example, an HTML table containing the properties and their values or
- 5310 the description of the document if available.

5311

5312 For Documents that support Content Streams:

- 5313 The repository SHOULD use the atom:content/src attribute to point to the content stream.
- 5314 The client SHOULD use cmisra:content if the content is not well-formed or would have
- 5315 trouble fitting inside an atom:content element. The repository MUST use the
- 5316 cmisra:content element if provided by the client over the atom:content element.

5317

5318 Other Objects (Folders, Relationships, and other Document Types that do not support Content

5319 Streams, etc):

- 5320 The repository MUST comply with the atom specification and have an atom:content
- 5321 element. This is repository specific. Any value in the content field MUST be ignored if the

5322 atom entry represents a non-document object by the CMIS repository when the atom
5323 entry is POST^ued to a collection or sent to the repository via a PUT.

5324
5325 When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding
5326 writable CMIS property. For example, atom:title will overwrite cmis:name.

5327
5328 At any point where an Atom document of CMIS Type is sent or returned, it must be a valid Atom Entry
5329 document and conform to the guidelines below for a cmis type definition:

- 5330 • atom:title MUST be the cmis:displayName
- 5331 • The repository SHOULD populate the atom:summary tag with text that best represents a
5332 summary of the object. For example, the type description if available.
- 5333 • The repository MUST comply with the atom specification and have an atom:content element. This
5334 is repository specific. Any value in the content field MUST be ignored if the atom entry represents
5335 a non-document object by the CMIS repository when the atom entry is POST^ued to a collection or
5336 sent to the repository via a PUT.

5337
5338
5339 Any atom element that is not specified is repository-specific.

5340 3.5.2.1 Hierarchical Atom Entries

5341 The repository SHOULD NOT provide any links to hierarchical objects if those capabilities are not
5342 supported with the exception of getTypeDescendants which is required

5343
5344 For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate
5345 a cmisra:children element in the atom:entry with the enclosing feed of its direct children. This pattern
5346 continues until the depth is satisfied.

5347
5348 The cmisra:children element MUST include an atom:feed element that contains the children entries of this
5349 resource.

5350
5351 If an entry does not contain cmisra:children element, then the entry MAY have children even though it is
5352 not represented in the atom entry.

5353
5354 For Example, here is a minimal Atom Entry with CMIS Children Extension Element:

```
5355 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
5356 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"  
5357 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"  
5358 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
5359 open.org/ns/cmisis/restatom/200908/">  
5360 <atom:author>  
5361 <atom:name>Al Brown</atom:name>  
5362 </atom:author>  
5363 <atom:content src="http://cmisexample.oasis-open.org/repl/af1d8c7f-b554-  
5364 4dfb-bfe1-1f41e4b34fef"/>  
5365 <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef</atom:id>  
5366 <atom:title type="text">CMIS Example Folder as Customer  
5367 type</atom:title>  
5368 <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>  
5369 <cmisra:object>  
5370 <cmis:properties>
```

```

5372 |         <cmis:propertyId localName=""rep-cmis:objectId""
5373 | propertyDefinitionId=""cmis:objectId"">
5374 |             <cmis:value>af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef</cmis:value>
5375 |         </cmis:propertyId>
5376 |     </cmis:properties>
5377 | </cmisra:object>
5378 | <cmisra:pathSegment>customer</cmisra:pathSegment>
5379 | <cmisra:children>
5380 |     <atom:feed>
5381 |         <atom:title type=""text"">CMIS Example Folder as Customer
5382 | type</atom:title>
5383 |         <atom:author>
5384 |             <atom:name>Al Brown</atom:name>
5385 |             <atom:uri>http://www.ibm.com/</atom:uri>
5386 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
5387 |         </atom:author>
5388 |         <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5389 |         <atom:id>urn:uuid:ce2d65af-b246-454b-90ff-0986d9b05178</atom:id>
5390 |         <atom:link type=""application/atom+xml;type=feed"" rel=""self""
5391 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5392 | 1f41e4b34fef/3""/>
5393 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
5394 | href=""http://cmisexample.oasis-open.org/rep1//service""/>
5395 |         <atom:link type=""application/atom+xml;type=entry"" rel=""via""
5396 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5397 | 1f41e4b34fef""/>
5398 |         <atom:link type=""application/atom+xml;type=feed""
5399 | rel=""http://docs.oasis-open.org/ns/cmis/link/200908/foldertree""
5400 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5401 | 1f41e4b34fef/foldertree""/>
5402 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
5403 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5404 | 1f41e4b34fef/children""/>
5405 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
5406 | href=""http://cmisexample.oasis-open.org/rep1/2eb09309-58f7-4627-b735-
5407 | 4d5cf4ba6554""/>
5408 |         <atom:entry>
5409 |             <atom:author>
5410 |                 <atom:name>Al Brown</atom:name>
5411 |             </atom:author>
5412 |             <atom:content src=""http://cmisexample.oasis-
5413 | open.org/rep1/af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef""/>
5414 |             <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-
5415 | 1f41e4b34fef</atom:id>
5416 |             <atom:title type=""text"">CMIS Example Child of
5417 | Folder</atom:title>
5418 |             <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5419 |             <cmisra:object>
5420 |                 <cmis:properties>
5421 |                     <cmis:propertyId localName=""rep-cmis:objectId""
5422 | propertyDefinitionId=""cmis:objectId"">
5423 |                         <cmis:value>af1d8c7f-b554-4dfb-bfe1-
5424 | 1f41e4b34fef</cmis:value>
5425 |                     </cmis:propertyId>
5426 |                 </cmis:properties>
5427 |             </cmisra:object>
5428 |             <cmisra:pathSegment>document</cmisra:pathSegment>
5429 |         </atom:entry>
5430 |     </atom:feed>
5431 | </cmisra:children>
5432 | </atom:entry>
5433 |

```

5434 Please also see the example documents included with the schema.

5435 3.6 AtomPub Service Document (Repository)

5436 The AtomPub Service Document contains the set of repositories that are available. Each repository is
5437 mapped to a app:workspace element in the AtomPub Service document.

5438

5439 CMIS Services exposed:

5440 GET: getRepositories, getRepositoryInfo

5441

5442 Media Type: application/atomsvc+xml

5443

5444 How the client will get the initial AtomPub (APP) service document or the URI for the service document is
5445 repository specific. Examples are via URI, or loading the service document from disk.

5446

5447 The service document will be available from Atom Entry and Atom Feed documents via a link relationship,
5448 service. That AtomPub service document MUST contain only one workspace element which MUST be
5449 the workspace representing the repository containing the Atom Entry or Atom Feed document.

5450

5451 A workspace element for a CMIS repository MUST have a collection element for each of following
5452 collections: Each collection MUST also contain a cmisra:collectionType element with the given value:

- 5453 • Root Folder Children Collection: Root folder of the Repository
 - 5454 ○ `!root!` for the children collection of the root folder
 - 5455 ○ `cmisra:collectiontype=!root!`
- 5456 • Types Children Collection: Collection containing the base types in the repository
 - 5457 ○ `!types!` for the children collection
 - 5458 ○ `cmisra:collectiontype=!types!`

5459

5460 The workspace element SHOULD contain these collections if the repository supports this functionality:

- 5461 • CheckedOut collection: collection containing all checked out documents user can see
 - 5462 ○ `!checkedout!`
 - 5463 ○ `cmisra:collectiontype=!checkedout!`
- 5464 • Query collection: Collection for posting queries to be executed
 - 5465 ○ `!query!`
 - 5466 ○ `cmisra:collectiontype=!query!`
- 5467 • Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
 - 5468 ○ `!unfiled!`
 - 5469 ○ `cmisra:collectiontype=!unfiled!`

5470

5471 The repository MUST include the URI templates in the workspace elements.

5472

5473 The workspace element MUST also contain the following link element with the relation:

- 5474 • <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>: This link relation points to the
5475 types descendants for the base types in the repository.

5476

5477 The workspace element MUST contain the following link elements of the following relations for those
5478 services which are supported by the repository:

- 5479
- 5480
- <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>: This link relation points to the folder tree of the root folder. See Folder Tree resource for more information.
 - <http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>: This link relation points to the Root Folder Descendants Feed for the root folder.
 - <http://docs.oasis-open.org/ns/cmisis/link/200908/changes>: This link relation points to the changes feed for the repository.
- 5481
- 5482
- 5483
- 5484
- 5485

5486 The workspace element may include app:collection element for the collections that represent folders in
5487 the repository. However, an alternative approach, especially for a repository with many folders, is to not
5488 enumerate those collections here, but include the app:collection element per RFC5023 in the Atom Feed
5489 document.

5490 3.6.1 URI Templates

5491

5492 CMIS defines the following URI Templates:

- 5493
- objectbyid
 - objectbypath
 - query
 - typebyid
- 5494
- 5495
- 5496
- 5497

5498 Repositories MUST provide the following URI Templates:

- 5499
- objectbyid
 - objectbypath
 - typebyid
- 5500
- 5501
- 5502

5503 Repositories MUST provide the URI Template query if the repository supports query.

5504

5505 [URI Templates MUST only be used with HTTP GET.](#)

5506

5507 Repositories MAY extend that set of templates. Those URI Template Types will be repository specific.
5508 Repositories MAY have more than one entry per URI Template type if the entries have different media
5509 types.

5510

5511 URI Templates are simple replacement of the template parameter with the specified value. If a client
5512 does not want to specify a value for some of these variables, then the client MUST substitute an empty
5513 string for the variable.

5514

5515 For example, if the URI template that supports the variable {id} is

5516 `http://example.org/repl/getbyid/{id}`

5517

5518 If the client wants to find the entry for an object with an id of "obj_1" then the URI would be:

5519 `http://example.org/repl/getbyid/obj_1`

5520

5521 Arguments that are substituted for URI template parameters MUST be percent escaped according to
5522 RFC3986. Please see that RFC for more information.

5523

5524 All variables MUST be in the template.

5525

5526 Structure of URI Template:

```

5527 <xs:complexType name="cmisUriTemplateType">
5528   <xs:sequence>
5529     <xs:element name="template" type="xs:string" />
5530     <xs:element name="type" type="xs:string" />
5531     <xs:element name="mediatype" type="xs:string" />
5532     <xs:any processContents="lax" namespace="##other"
5533 minOccurs="0"
5534           maxOccurs="unbounded" />
5535   </xs:sequence>
5536 </xs:complexType>

```

5537

5538

5539 Example of URI Template element in an AtomPub Workspace Element:

```

5540 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5541 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5542 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5543 open.org/ns/cmisis/messaging/200908/"
5544 xmlns:atom="http://www.w3.org/2005/Atom"
5545 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5546 open.org/ns/cmisis/restatom/200908/">
5547   <cmisra:template>http://cmisexample.oasis-
5548 open.org/rep1/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
5549 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
5550 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
5551   <cmisra:type>objectbyid</cmisra:type>
5552   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5553 </cmisra:uritemplate>

```

5554

5555 Please also see the example documents included with the schema.

5556

5557 3.6.1.1 Object By Id

5558 This URI template provides a method for creating an URI that directly accesses an atom entry
5559 representing documents, folders, policies or relationship objects. See section 3.10 for more information.

5560

5561 Type: objectbyid

5562 Media Type: application/atom+xml;type=entry

5563

5564 Service: getObjectById

5565

5566 Variables that are supported by the template:

- 5567 • {id}: Id of object
- 5568 • {filter}: Property Filter
- 5569 • {includeAllowableActions}
 - 5570 ○ Valid values: true, false
- 5571 • {includePolicyIds}: Include Policy Ids:
 - 5572 ○ Valid values: true, false

- 5573 • {includeRelationships}: Include relationships
 - 5574 ○ Valid values: See enumIncludeRelationships
- 5575 • {includeACL}: Include ACLs
 - 5576 ○ Valid values: true, false
- 5577 • {renditionFilter}
 - 5578 ○ Valid values: Please see renditionFilter in CMIS Domain Model

5579

5580 Example:

```

5581 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5582 <cmisra:writetemplate xmlns:cmis="http://docs.oasis-
5583 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5584 open.org/ns/cmis/messaging/200908/"
5585 xmlns:atom="http://www.w3.org/2005/Atom"
5586 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5587 open.org/ns/cmis/restatom/200908/">
5588   <cmisra:template>http://cmisexample.oasis-
5589 open.org/rep1/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
5590 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
5591 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
5592   <cmisra:type>objectbyid</cmisra:type>
5593   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5594 </cmisra:writetemplate>
5595

```

5596

5597 Please also see the example documents included with the schema.

5598 3.6.1.2 Object By Path

5599 This URI template provides a method for creating an URI that directly accesses an atom entry
 5600 representing documents, folders or policy objects. See section 3.10 for more information.

5601

5602 Type: objectbypath

5603 Media Type: application/atom+xml;type=entry

5604

5605 Service: getObjectByPath

5606

5607 Variables that are supported by the template:

- 5608 • {path}: Path of Object
- 5609 • {filter}: Property Filter
- 5610 • {includeAllowableActions}: Boolean for include Allowable Actions
 - 5611 ○ Valid values: true, false
- 5612 • {includePolicyIds}: Include Policy Ids:
 - 5613 ○ Valid values: true, false
- 5614 • {includeRelationships}: Include relationships
 - 5615 ○ Valid values: See enumIncludeRelationships
- 5616 • {includeACL}: Include ACLs
 - 5617 ○ Valid values: true, false
- 5618 • {renditionFilter}

5619 o Valid values: Please see renditionFilter in CMIS Domain Model

5620

5621

5622 Example:

```
5623 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5624 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5625 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5626 open.org/ns/cmisis/messaging/200908/"
5627 xmlns:atom="http://www.w3.org/2005/Atom"
5628 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5629 open.org/ns/cmisis/restatom/200908/">
5630   <cmisra:template>http://cmisexample.oasis-
5631 open.org/repl/objectbypath?p={path}&filter={filter}&includeAllowableAc-
5632 tions={includeAllowableActions}&includePolicyIds={includePolicyIds}&in-
5633 cludeRelationships={includeRelationships}&includeACL={includeACL}</cmisra:
5634 template>
5635   <cmisra:type>objectbypath</cmisra:type>
5636   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5637 </cmisra:uritemplate>
5638
```

5639

5640 Please also see the example documents included with the schema.

5641

5642 3.6.1.3 Query

5643 Type: query

5644 Media Type: application/atom+xml;type=feed

5645

5646 Service: query

5647

5648 Variables that are supported by the template:

- 5649 • {q}: CMIS Query Statement
- 5650 • {searchAllVersions}: Boolean, true if to search all versions
- 5651 • {maxItems}: Integer, Max items to return
- 5652 • {skipCount}: Integer, Items to skip
- 5653 • {includeAllowableActions}: Boolean
- 5654 • {includeRelationships}: Boolean

5655

5656

5657 Example:

```
5658 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5659 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5660 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5661 open.org/ns/cmisis/messaging/200908/"
5662 xmlns:atom="http://www.w3.org/2005/Atom"
5663 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5664 open.org/ns/cmisis/restatom/200908/">
```



```
5665     <cmisra:template>http://cmisexample.oasis-
5666 open.org/rep1/query?q={q}&searchAllVersions={searchAllVersions}&maxItems={maxItems}&skipCount={skipCount}&includeAllowableActions={includeAllowableActions}&includeRelationships={includeRelationships}</cmisra:template>
5667     <cmisra:type>query</cmisra:type>
5668     <cmisra:mediatype>application/atom+xml;type=feed</cmisra:mediatype>
5669 </cmisra:uritemplate>
5670
5671
5672
5673
```

5674

5675 Please also see the example documents included with the schema.

5676

5677 3.6.1.4 Type By Id

5678 Type: typebyid

5679 Media Type: application/atom+xml;type=entry

5680

5681 Service: getTypeDefinition

5682

5683 Variables that are supported by the template:

- 5684 • {id}: CMIS Type Id

5685

5686

5687 Example:

```
5688 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5689 <cmisra:uritemplate xmlns:cmisra="http://docs.oasis-
5690 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5691 open.org/ns/cmisis/messaging/200908/"
5692 xmlns:atom="http://www.w3.org/2005/Atom"
5693 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5694 open.org/ns/cmisis/restatom/200908/">
5695   <cmisra:template>http://cmisexample.oasis-
5696 open.org/rep1/type?id={id}</cmisra:template>
5697   <cmisra:type>query</cmisra:type>
5698   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5699 </cmisra:uritemplate>
```

5700

5701 Please also see the example documents included with the schema.

5702

5703

5704 3.6.2 HTTP Methods

5705 3.6.2.1 GET

5706 This retrieves the AtomPub Service document for a specified repository. This exposes the capabilities defined in getRepositories and getRepositoryInfo in the Domain Model.

5707

5708 The optional argument MAY be specified:

- 5709 • repositoryId:

5710

- 5711 ○ This query parameter allows a client to specify a different repository than the one that is
- 5712 referenced by the URI.
- 5713 ○ If specified, the repository MUST return the AtomPub services document for the specified
- 5714 repository if that repository exists.
- 5715 ○ If not specified, the repository MUST return the service document for the repository that is
- 5716 referenced by URI.
- 5717

5718 3.7 Service Collections

5719 These are the collections that are included on an AtomPub Service document in the workspace element.

5720 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
5721 verb in a repository-specific manner.

5722 3.7.1 Root Folder Collection

5723 This is a collection described in the service document. Please see [Folder Children](#).

5724 3.7.2 Query Collection

5725 This is a collection for processing queries. If the implementation supports GET on this collection, then the
5726 implementation SHOULD at least return a feed consisting of zero or more atom entries. These atom
5727 entries should represent persisted objects related to query such as persisted queries, long running
5728 queries or search templates.

5729

5730 CMIS Services exposed via HTTP verbs:

5731 POST: Query

5732

5733 Media Type: application/atom+xml;type=feed

5734 Accept:

- 5735 • MUST support CMIS Query document,
- 5736 • MAY support other media type

5737

5738 Link Relations on resulting feed from Query Collection:

- 5739 • service: Points to service document containing the CMIS repository. The service document
- 5740 MUST contain only one workspace element.
 - 5741 ○ Media Type: application/atomsvc+xml
- 5742 • paging link relations as appropriate: first, next, previous, last

5743

5744 The following CMIS Atom extension element MAY be included inside the atom feed:

- 5745 • cmisra:numItems

5746

5747 The following CMIS Atom extension element MUST be included inside the atom entries:

- 5748 • cmisra:object inside atom:entry

5749

5750 3.7.2.1 POST

5751 This collection MUST accept CMIS Query documents (application/cmisquery+xml).

5752

5753 Upon submission (creation) of a query document, a response must be returned with a Location header
5754 representing the feed for that query. If the query cannot be performed and an atom feed returned, the
5755 repository MUST return the appropriate HTTP status code. In addition, the server SHOULD return the
5756 feed directly. If the server does so, the server should also return the Content-Location header.

5757

5758 The feed returned MUST contain a set of atom entries representing the result set from the query.

5759

5760 The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The
5761 atom entries MUST contain the CMIS extension element (cmisra:object) containing the properties
5762 specified by the query in the select clause of the query statement.

5763

5764 If all the selected properties can be mapped to the same type reference, then the repository MAY include
5765 additional information in the atom entry.

5766

5767 Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

5768

5769 Status Codes:

- 5770 • 201 Success

5771

5772 Headers returned:

- 5773 • Location Header
- 5774 • Content-Location Header

5775

5776 Link Relations on resulting feed from POST to Query Collection:

- 5777 • service: Points to service document containing the CMIS repository. The service document
5778 MUST contain only one workspace element.
 - 5779 ○ Media Type: application/atomsvc+xml
- 5780 • paging link relations as appropriate: first, next, previous, last

5781

5782 Example client request:

5783

```
POST /Query HTTP/1.1
Host: example.org
Content-Length: 756
Content-Type: application/cmisquery+xml
```

5784

5785

5786

5787

5788

5789

5790

5791

5792

5793

5794

5795

5796

5797

5798

5799

5800

5801

5802

5803

5804

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cmis:query xmlns:app="http://www.w3.org/2007/app"
xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
open.org/ns/cmisis/restatom/200908/">
  <cmis:statement>SELECT cmis:objectId FROM cmis:document</cmis:statement>
  <cmis:searchAllVersions>true</cmis:searchAllVersions>
  <cmis:includeAllowableActions>>false</cmis:includeAllowableActions>
  <cmis:includeRelationships>none</cmis:includeRelationships>
  <cmis:renditionFilter>*</cmis:renditionFilter>
  <cmis:maxItems>50</cmis:maxItems>
  <cmis:skipCount>0</cmis:skipCount>
</cmis:query>
```

5805

5806 Example server response:

```
5807 HTTP/1.1 201 Created
5808 Date: Mon, 25 Jan 2010 10:21:00 -0800
5809 Content-Length: 1830
5810 Content-Type: application/atom+xml;type=feed
5811 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/44ce5b47-
5812 ebc3-4513-86e0-d3f46c77d0a8
5813 Location: http://cmisexample.oasis-open.org/rep1/queryresult/44ce5b47-ebc3-
5814 4513-86e0-d3f46c77d0a8
5815
5816
5817 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5818 <atom:feed xmlns:app="http://www.w3.org/2007/app"
5819 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5820 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5821 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5822 open.org/ns/cmisis/restatom/200908/">
5823   <atom:title type="text">CMIS Query Result for SELECT cmis:objectId FROM
5824 cmis:document</atom:title>
5825   <atom:author>
5826     <atom:name>Al Brown</atom:name>
5827     <atom:uri>http://www.ibm.com/</atom:uri>
5828     <atom:email>albertcbrown@us.ibm.com</atom:email>
5829   </atom:author>
5830   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
5831   <atom:id>urn:uuid:811b1b9b-80f5-4788-b46c-aa77564e294b</atom:id>
5832   <atom:link type="application/atom+xml;type=feed" rel="self"
5833 href="http://cmisexample.oasis-open.org/rep1/11355977-434b-4e71-b83a-
5834 77dea9878e04/3"/>
5835   <atom:link type="application/atomsvc+xml" rel="service"
5836 href="http://cmisexample.oasis-open.org/rep1//service"/>
5837   <atom:entry>
5838     <atom:author>
5839       <atom:name>Al Brown</atom:name>
5840     </atom:author>
5841     <atom:content src="http://cmisexample.oasis-open.org/rep1/a3386ea0-
5842 0477-4a74-96bd-70d3da1c483a"/>
5843     <atom:id>urn:uuid:a3386ea0-0477-4a74-96bd-70d3da1c483a</atom:id>
5844     <atom:title type="text">Resulting Document</atom:title>
5845     <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
5846     <cmisra:object>
5847       <cmis:properties>
5848         <cmis:propertyId queryName="cmis:objectId" localName="rep-
5849 cmis:objectId" propertyDefinitionId="cmis:objectId">
5850           <cmis:value>a3386ea0-0477-4a74-96bd-
5851 70d3da1c483a</cmis:value>
5852         </cmis:propertyId>
5853       </cmis:properties>
5854     </cmisra:object>
5855   </atom:entry>
5856 </atom:feed>
5857
```

5858

5859 Please also see the example documents included with the schema.

5860

5861 3.7.3 Checked Out Collection

5862 This is a collection described in the service document that contains the private working copies (PWCs) of
5863 the checkedout documents in the repository.

5864 CMIS Services:

5865 GET: getCheckedOutDocs

5866 POST: checkOut

5867 Media Type: application/atom+xml;type=feed

5868 Accept:

5869 • MUST support Atom Entry Documents with CMIS extensions

5870 ○ application/atom+xml;type=entry or

5871 ○ application/cmisaatom+xml

5872 • MAY support other media type

5873

5874 Link Relations:

5875 • service: Points to service document containing the CMIS repository. The service document
5876 MUST contain only one workspace element.

5877 ○ Media Type: application/atomsvc+xml

5878 • paging link relations as appropriate: first, next, previous, last

5879

5880 The following CMIS Atom extension element MAY be included inside the atom feed:

5881 • cmisra:numItems

5882

5883 The following CMIS Atom extension element MUST be included inside the atom entries:

5884 • cmisra:object inside atom:entry

5885

5886 3.7.3.1 GET

5887 The following arguments may be supplied. Please see the domain model for more information:

5888 • filter

5889 • folderId

5890 • maxItems

5891 • skipCount

5892 • renditionFilter

5893 • includeAllowableActions

5894 • includeRelationships

5895 3.7.3.2 POST

5896 | When an atom entry is POST^{ed} to this collection, the atom entry will be checked out. A Content-
5897 Location header MUST be returned containing the location of the private working copy.

5898

5899 Example client request:

5900 POST /CheckedOut HTTP/1.1

5901 Host: example.org

5902 Content-Length: 1044

5903 Content-Type: application/atom+xml;type=entry

5904

5905

5906 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

5907 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
5908 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5909 | open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5910 | open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5911 | open.org/ns/cmisis/restatom/200908/">
5912 |   <atom:author>
5913 |     <atom:name>Al Brown</atom:name>
5914 |   </atom:author>
5915 |   <atom:content src="http://cmisexample.oasis-open.org/rep1/8d32d716-701b-
5916 | 4491-84e8-ad57c8230940"/>
5917 |   <atom:id>urn:uuid:8d32d716-701b-4491-84e8-ad57c8230940</atom:id>
5918 |   <atom:title type="text">CMIS Example Document to checkout</atom:title>
5919 |   <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
5920 |   <cmisra:object>
5921 |     <cmis:properties>
5922 |       <cmis:propertyId localName="rep-cmis:objectId"
5923 | propertyDefinitionId="cmis:objectId">
5924 |         <cmis:value>8d32d716-701b-4491-84e8-ad57c8230940</cmis:value>
5925 |       </cmis:propertyId>
5926 |     </cmis:properties>
5927 |   </cmisra:object>
5928 | </atom:entry>
5929 |

```

5930

5931 **Example server response:**

```

5932 | HTTP/1.1 201 Created
5933 | Date: Mon, 25 Jan 2010 10:21:00 -0800
5934 | Content-Length: 7846
5935 | Content-Type: application/atom+xml;type=entry
5936 | Content-Location: http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-
5937 | 8fab-4aa6e6797dbe
5938 | Location: http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5939 | 4aa6e6797dbe
5940 |
5941 |
5942 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5943 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
5944 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5945 | open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5946 | open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5947 | open.org/ns/cmisis/restatom/200908/">
5948 |   <atom:author>
5949 |     <atom:name>Al Brown</atom:name>
5950 |     <atom:uri>http://www.ibm.com/</atom:uri>
5951 |     <atom:email>albertcbrown@us.ibm.com</atom:email>
5952 |   </atom:author>
5953 |   <atom:content src="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-
5954 | 491c-8fab-4aa6e6797dbe"/>
5955 |   <atom:id>urn:uuid:6cce57fc-4e31-491c-8fab-4aa6e6797dbe</atom:id>
5956 |   <atom:title type="text">CMIS Example Child of Folder</atom:title>
5957 |   <atom:updated>2010-01-25T10:21:00.396-08:00</atom:updated>
5958 |   <atom:link rel="self" href="http://cmisexample.oasis-
5959 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
5960 |   <atom:link rel="edit" href="http://cmisexample.oasis-
5961 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
5962 |   <atom:link type="application/cmisis+xml;type=allowableActions"
5963 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
5964 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5965 | 4aa6e6797dbe/allowableactions"/>
5966 |   <atom:link type="application/atom+xml;type=entry" rel="describedby"
5967 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5968 | 4aa6e6797dbe/type"/>

```

```

5969 | <atom:link type="application/atomsvc+xml" rel="service"
5970 | href="http://cmisexample.oasis-open.org/rep1//service"/>
5971 | <atom:published>2010-01-25T10:21:00.396-08:00</atom:published>
5972 | <atom:summary type="html">HTML summary of Entry 6cce57fc-4e31-491c-
5973 | 8fab-4aa6e6797dbe</atom:summary>
5974 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
5975 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/edit-media"/>
5976 | <atom:link rel="alternate" href="http://cmisexample.oasis-
5977 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/alternate"/>
5978 | <atom:link type="application/atom+xml;type=feed" rel="up"
5979 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5980 | 4aa6e6797dbe/parents"/>
5981 | <atom:link type="application/atom+xml;type=feed" rel="version-history"
5982 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5983 | 4aa6e6797dbe/allversions"/>
5984 | <atom:link type="application/atom+xml;type=entry" rel="current-
5985 | version" href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-
5986 | 8fab-4aa6e6797dbe/latest"/>
5987 | <atom:link type="application/atom+xml;type=feed"
5988 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
5989 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5990 | 4aa6e6797dbe/relationships"/>
5991 | <atom:link type="application/atom+xml;type=feed"
5992 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
5993 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5994 | 4aa6e6797dbe/policies"/>
5995 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
5996 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
5997 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/acl"/>
5998 | <atom:link type="application/atom+xml;type=feed" rel="working-copy"
5999 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
6000 | 4aa6e6797dbe/pwc"/>
6001 | <cmisra:object>
6002 | <cmis:properties>
6003 | <cmis:propertyId localName="rep-cmis:objectId"
6004 | propertyDefinitionId="cmis:objectId">
6005 | <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
6006 | </cmis:propertyId>
6007 | <cmis:propertyId localName="rep-cmis:objectTypeId"
6008 | propertyDefinitionId="cmis:objectTypeId">
6009 | <cmis:value>customer</cmis:value>
6010 | </cmis:propertyId>
6011 | <cmis:propertyString localName="rep-cmis:name"
6012 | propertyDefinitionId="cmis:name">
6013 | <cmis:value>CMIS Example Child of Folder</cmis:value>
6014 | </cmis:propertyString>
6015 | <cmis:propertyDateTime localName="rep-cmis:creationDate"
6016 | propertyDefinitionId="cmis:creationDate">
6017 | <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
6018 | </cmis:propertyDateTime>
6019 | <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6020 | propertyDefinitionId="cmis:lastModificationDate">
6021 | <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
6022 | </cmis:propertyDateTime>
6023 | <cmis:propertyId localName="rep-cmis:baseTypeId"
6024 | propertyDefinitionId="cmis:baseTypeId">
6025 | <cmis:value>cmis:document</cmis:value>
6026 | </cmis:propertyId>
6027 | <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6028 | propertyDefinitionId="cmis:lastModifiedBy">
6029 | <cmis:value>Al Brown</cmis:value>
6030 | </cmis:propertyString>
6031 | <cmis:propertyString localName="rep-cmis:createdBy"
6032 | propertyDefinitionId="cmis:createdBy">

```

```

6033         <cmis:value>Al Brown</cmis:value>
6034     </cmis:propertyString>
6035     <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
6036 | propertyDefinitionId=""cmis:isLatestVersion"">
6037         <cmis:value>>true</cmis:value>
6038     </cmis:propertyBoolean>
6039     <cmis:propertyBoolean localName=""rep-
6040 | cmis:isVersionSeriesCheckedOut""
6041 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
6042         <cmis:value>>true</cmis:value>
6043     </cmis:propertyBoolean>
6044     <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
6045 | propertyDefinitionId=""cmis:isMajorVersion"">
6046         <cmis:value>>false</cmis:value>
6047     </cmis:propertyBoolean>
6048     <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
6049 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
6050         <cmis:value>>false</cmis:value>
6051     </cmis:propertyBoolean>
6052     <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
6053 | propertyDefinitionId=""cmis:isImmutable"">
6054         <cmis:value>>false</cmis:value>
6055     </cmis:propertyBoolean>
6056     <cmis:propertyString localName=""rep-cmis:checkinComment""
6057 | propertyDefinitionId=""cmis:checkinComment"">
6058         <cmis:value>Checkin comment</cmis:value>
6059     </cmis:propertyString>
6060     <cmis:propertyString localName=""rep-cmis:versionLabel""
6061 | propertyDefinitionId=""cmis:versionLabel"">
6062         <cmis:value>0.1</cmis:value>
6063     </cmis:propertyString>
6064     <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
6065 | propertyDefinitionId=""cmis:contentStreamMimeType"">
6066         <cmis:value>text/plain</cmis:value>
6067     </cmis:propertyString>
6068     <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
6069 | propertyDefinitionId=""cmis:contentStreamFileName"">
6070         <cmis:value>text.txt</cmis:value>
6071     </cmis:propertyString>
6072     <cmis:propertyInteger localName=""rep-cmis:contentStreamLength""
6073 | propertyDefinitionId=""cmis:contentStreamLength"">
6074         <cmis:value>4234</cmis:value>
6075     </cmis:propertyInteger>
6076     <cmis:propertyString displayName=""Keywords for Document""
6077 | localName=""keywords"" propertyDefinitionId=""keywords"">
6078         <cmis:value>document</cmis:value>
6079         <cmis:value>example</cmis:value>
6080         <cmis:value>sample</cmis:value>
6081         <cmis:value>cmis</cmis:value>
6082     </cmis:propertyString>
6083     <cmis:propertyId localName=""rep-cmis:versionSeriesCheckedOutId""
6084 | propertyDefinitionId=""cmis:versionSeriesCheckedOutId"">
6085         <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
6086     </cmis:propertyId>
6087     <cmis:propertyString localName=""rep-
6088 | cmis:versionSeriesCheckedOutBy""
6089 | propertyDefinitionId=""cmis:versionSeriesCheckedOutBy"">
6090         <cmis:value>Al Brown</cmis:value>
6091     </cmis:propertyString>
6092 </cmis:properties>
6093 </cmisra:object>
6094 </atom:entry>
6095

```

6096

6097 Please also see the example documents included with the schema.

6098

6099 **3.7.4 Unfiled Collection**

6100 This is a collection described in the service document that contains all the unfiled documents in the
6101 repository. This collection **MUST** be available if un-filing or multi-filing is supported by the repository.

6102 A repository that supports un-filing **MAY** provide read access (GET). If read access is not provided, the
6103 repository **SHOULD** respond to a read attempt with the HTTP status code 405 (notSupported).

6104 CMIS Services:

6105 POST: removeObjectFromFolder

6106 Media Type: application/atom+xml;type=feed

6107 Accept:

6108 • **MUST** support Atom Entry Documents with CMIS extensions

6109 ○ application/atom+xml;type=entry or

6110 ○ application/cmisatom+xml

6111 • **MAY** support other media type

6112

6113 Link Relations:

6114 • service: Points to service document containing the CMIS repository. The service document
6115 **MUST** contain only one workspace element.

6116 ○ Media Type: application/atomsvc+xml

6117 • paging link relations as appropriate: first, next, previous, last

6118

6119 The following CMIS Atom extension element **MAY** be included inside the atom feed:

6120 • cmisra:numItems

6121

6122 The following CMIS Atom extension element **MUST** be included inside the atom entries:

6123 • cmisra:object inside atom:entry

6124

6125 **3.7.4.1 POST**

6126 This removes the object from all folders in the repository by default. If the optional argument removeFrom
6127 is specified, the object will only be removed from that folder only.

6128

6129 | If the Atom Entry POST^{ed}, does not have the CMIS extensions with a valid cmis:objectId property, the
6130 document does not exist, or the document is not in that folder, the appropriate HTTP status code **MUST**
6131 be returned.

6132

6133 This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

6134 • HTTP Success: 201

6135 • Location Header

6136

6137 The following arguments may be supplied. Please see the domain model for more information:

- 6138 • removeFrom: For repositories which support multi-filing, this parameter identifies which folder to
6139 remove this object from. If specified, it indicates the folder from which the object shall be moved.
6140 If not specified, the object will be removed from all folders.

6141

6142 Example client request:

```
6143 POST /Unfiled HTTP/1.1
6144 Host: example.org
6145 Content-Length: 1043
6146 Content-Type: application/atom+xml;type=entry
6147
6148
6149 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6150 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6151 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6152 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6153 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6154 open.org/ns/cmisis/restatom/200908/">
6155   <atom:author>
6156     <atom:name>Al Brown</atom:name>
6157   </atom:author>
6158   <atom:content src="http://cmisexample.oasis-open.org/rep1/12aa2bec-6f43-
6159 47d1-99ef-21797867173c"/>
6160   <atom:id>urn:uuid:12aa2bec-6f43-47d1-99ef-21797867173c</atom:id>
6161   <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6162   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
6163   <cmisra:object>
6164     <cmis:properties>
6165       <cmis:propertyId localName="rep-cmis:objectId"
6166 propertyDefinitionId="cmis:objectId">
6167         <cmis:value>12aa2bec-6f43-47d1-99ef-21797867173c</cmis:value>
6168       </cmis:propertyId>
6169     </cmis:properties>
6170   </cmisra:object>
6171 </atom:entry>
6172
```

6173

6174 Example server response:

```
6175 HTTP/1.1 201 Created
6176 Date: Mon, 25 Jan 2010 10:21:00 -0800
6177 Content-Length: 7234
6178 Content-Type: application/atom+xml;type=entry
6179 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/15118373-
6180 8911-442b-9774-da3b102f224c
6181 Location: http://cmisexample.oasis-open.org/rep1/queryresult/15118373-8911-
6182 442b-9774-da3b102f224c
6183
6184
6185 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6186 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6187 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6188 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6189 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6190 open.org/ns/cmisis/restatom/200908/">
6191   <atom:author>
6192     <atom:name>Al Brown</atom:name>
6193     <atom:uri>http://www.ibm.com/</atom:uri>
6194     <atom:email>albertcbrown@us.ibm.com</atom:email>
6195   </atom:author>
6196   <atom:content src="http://cmisexample.oasis-open.org/rep1/15118373-8911-
6197 442b-9774-da3b102f224c"/>
6198   <atom:id>urn:uuid:15118373-8911-442b-9774-da3b102f224c</atom:id>
```

```

6199 | <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6200 | <atom:updated>2010-01-25T10:21:00.443-08:00</atom:updated>
6201 | <atom:link rel="self" href="http://cmisexample.oasis-
6202 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c"/>
6203 | <atom:link rel="edit" href="http://cmisexample.oasis-
6204 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c"/>
6205 | <atom:link type="application/cmismedia+xml;type=allowableActions"
6206 | rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/allowableactions"
6207 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
6208 | da3b102f224c/allowableactions"/>
6209 | <atom:link type="application/atom+xml;type=entry" rel="describedby"
6210 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
6211 | da3b102f224c/type"/>
6212 | <atom:link type="application/atomsvc+xml" rel="service"
6213 | href="http://cmisexample.oasis-open.org/rep1//service"/>
6214 | <atom:published>2010-01-25T10:21:00.443-08:00</atom:published>
6215 | <atom:summary type="html">HTML summary of Entry 15118373-8911-442b-
6216 | 9774-da3b102f224c</atom:summary>
6217 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
6218 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c/edit-media"/>
6219 | <atom:link rel="alternate" href="http://cmisexample.oasis-
6220 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c/alternate"/>
6221 | <atom:link type="application/atom+xml;type=feed" rel="up"
6222 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
6223 | da3b102f224c/parents"/>
6224 | <atom:link type="application/atom+xml;type=feed" rel="version-history"
6225 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
6226 | da3b102f224c/allversions"/>
6227 | <atom:link type="application/atom+xml;type=entry" rel="current-
6228 | version" href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-
6229 | 9774-da3b102f224c/latest"/>
6230 | <atom:link type="application/atom+xml;type=feed"
6231 | rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/relationships"
6232 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
6233 | da3b102f224c/relationships"/>
6234 | <atom:link type="application/atom+xml;type=feed"
6235 | rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/policies"
6236 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
6237 | da3b102f224c/policies"/>
6238 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6239 | open.org/ns/cmismedia/link/200908/acl" href="http://cmisexample.oasis-
6240 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c/acl"/>
6241 | <cmisra:object>
6242 | <cmis:properties>
6243 | <cmis:propertyId localName="rep-cmis:objectId"
6244 | propertyDefinitionId="cmis:objectId">
6245 | <cmis:value>15118373-8911-442b-9774-da3b102f224c</cmis:value>
6246 | </cmis:propertyId>
6247 | <cmis:propertyId localName="rep-cmis:objectTypeId"
6248 | propertyDefinitionId="cmis:objectTypeId">
6249 | <cmis:value>customer</cmis:value>
6250 | </cmis:propertyId>
6251 | <cmis:propertyString localName="rep-cmis:name"
6252 | propertyDefinitionId="cmis:name">
6253 | <cmis:value>CMIS Example Document to unfiled</cmis:value>
6254 | </cmis:propertyString>
6255 | <cmis:propertyDateTime localName="rep-cmis:creationDate"
6256 | propertyDefinitionId="cmis:creationDate">
6257 | <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
6258 | </cmis:propertyDateTime>
6259 | <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6260 | propertyDefinitionId="cmis:lastModificationDate">
6261 | <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
6262 | </cmis:propertyDateTime>

```

```

6263 |         <cmis:propertyId localName=""rep-cmis:baseTypeId""
6264 | propertyDefinitionId=""cmis:baseTypeId"">
6265 |             <cmis:value>cmis:document</cmis:value>
6266 |         </cmis:propertyId>
6267 |         <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
6268 | propertyDefinitionId=""cmis:lastModifiedBy"">
6269 |             <cmis:value>Al Brown</cmis:value>
6270 |         </cmis:propertyString>
6271 |         <cmis:propertyString localName=""rep-cmis:createdBy""
6272 | propertyDefinitionId=""cmis:createdBy"">
6273 |             <cmis:value>Al Brown</cmis:value>
6274 |         </cmis:propertyString>
6275 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
6276 | propertyDefinitionId=""cmis:isLatestVersion"">
6277 |             <cmis:value>>true</cmis:value>
6278 |         </cmis:propertyBoolean>
6279 |         <cmis:propertyBoolean localName=""rep-
6280 | cmis:isVersionSeriesCheckedOut""
6281 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
6282 |             <cmis:value>>false</cmis:value>
6283 |         </cmis:propertyBoolean>
6284 |         <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
6285 | propertyDefinitionId=""cmis:isMajorVersion"">
6286 |             <cmis:value>>false</cmis:value>
6287 |         </cmis:propertyBoolean>
6288 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
6289 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
6290 |             <cmis:value>>false</cmis:value>
6291 |         </cmis:propertyBoolean>
6292 |         <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
6293 | propertyDefinitionId=""cmis:isImmutable"">
6294 |             <cmis:value>>false</cmis:value>
6295 |         </cmis:propertyBoolean>
6296 |         <cmis:propertyString localName=""rep-cmis:checkinComment""
6297 | propertyDefinitionId=""cmis:checkinComment"">
6298 |             <cmis:value>Checkin comment</cmis:value>
6299 |         </cmis:propertyString>
6300 |         <cmis:propertyString localName=""rep-cmis:versionLabel""
6301 | propertyDefinitionId=""cmis:versionLabel"">
6302 |             <cmis:value>0.1</cmis:value>
6303 |         </cmis:propertyString>
6304 |         <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
6305 | propertyDefinitionId=""cmis:contentStreamMimeType"">
6306 |             <cmis:value>text/plain</cmis:value>
6307 |         </cmis:propertyString>
6308 |         <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
6309 | propertyDefinitionId=""cmis:contentStreamFileName"">
6310 |             <cmis:value>text.txt</cmis:value>
6311 |         </cmis:propertyString>
6312 |         <cmis:propertyInteger localName=""rep-cmis:contentStreamLength""
6313 | propertyDefinitionId=""cmis:contentStreamLength"">
6314 |             <cmis:value>4234</cmis:value>
6315 |         </cmis:propertyInteger>
6316 |         <cmis:propertyString displayName=""Keywords for Document""
6317 | localName=""keywords"" propertyDefinitionId=""keywords"">
6318 |             <cmis:value>document</cmis:value>
6319 |             <cmis:value>example</cmis:value>
6320 |             <cmis:value>sample</cmis:value>
6321 |             <cmis:value>cmis</cmis:value>
6322 |         </cmis:propertyString>
6323 |     </cmis:properties>
6324 | </cmisra:object>
6325 | </atom:entry>
6326 |

```

6327

6328 Please also see the example documents included with the schema.

6329

6330 **3.7.5 Types Children Collection**

6331 This is a collection described in the service document that contains the types in the repository under the
6332 specified parent type. If no parent type is specified, then the base types are returned in the feed. This
6333 feed does not include any nesting and is a flat feed.

6334 CMIS Services:

6335 GET: getTypeChildren

6336 Media Type: application/atom+xml;type=feed

6337

6338 Link Relations:

- 6339 • service: Points to service document containing the CMIS repository. The service document
6340 MUST contain only one workspace element.
 - 6341 ○ Media Type: application/atomsvc+xml
- 6342 • via: points to the type definition entry whose children represent this feed
- 6343 • down: points to the atom feed document representing the descendents collection for this same
6344 type with media type of application/cmistree+xml
- 6345 • paging link relations as appropriate: first, next, previous, last
- 6346 • up: points to the parent type definition
 - 6347 ○ If this is a children feed for a base object type, this link is not present.

6348

6349 This feed contains a set of atom entries for each child type definition.

6350

6351 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6352 • cmisra:numItems

6353

6354 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6355 • cmisra:type inside atom:entry

6356

6357

6358 **3.7.5.1 GET**

6359 The following arguments may be supplied. Please see the domain model for more information:

- 6360 • includePropertyDefinitions
- 6361 • maxItems
- 6362 • skipCount
- 6363 • typeId

6364 **3.8 Collections**

6365 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
6366 verb in a repository-specific manner.

6367

6368 **3.8.1 Relationships Collection**

6369 This is the set of relationships available (either source or target or both) from a specific item such as a
6370 document, folder or policy.

6371 CMIS Services:

6372 GET: getObjectRelationships

6373 POST: createRelationship

6374 Media Type: application/atom+xml;type=feed

6375 Accept:

6376 • MUST support Atom Entry Documents with CMIS extensions

6377 ○ application/atom+xml;type=entry or

6378 ○ application/cmisaatom+xml

6379 • MAY support other media type

6380

6381 Link Relations:

6382 • service: Points to service document containing the CMIS repository. The service document
6383 MUST contain only one workspace element.

6384 ○ Media Type: application/atomsvc+xml

6385 • paging link relations as appropriate: first, next, previous, last

6386

6387 The following CMIS Atom extension element MAY be included inside the atom feed:

6388 • cmisra:numItems

6389

6390 The following CMIS Atom extension element MUST be included inside the atom entries:

6391 • cmisra:object inside atom:entry

6392

6393 **3.8.1.1 GET**

6394 The following arguments may be supplied. Please see the domain model for more information:

6395 • typeId

6396 • includeSubRelationshipTypes

6397 • relationshipDirection

6398 • maxItems

6399 • skipCount

6400 • filter

6401 • includeAllowableActions

6402 **3.8.1.2 POST**

6403 When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new
6404 CMIS relationship, then that relationship will be created.

6405 The server MUST return the appropriate HTTP status code if the source is different than the sourceId or
6406 target different than the targetId for the source and targets specified in this collection.

6407 The server MUST return the appropriate status code if the cmis:objectTypeId is not specified.

6408

6409 Example client request:

```
6410 POST /relationships/source/dbf0316c-47b5-47c9-a2fa-f005eb93f0a4 HTTP/1.1
6411 Host: example.org
6412 Content-Length: 1432
6413 Content-Type: application/atom+xml;type=entry
6414
6415
6416 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6417 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6418 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6419 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6420 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6421 open.org/ns/cmisis/restatom/200908/">
6422   <atom:author>
6423     <atom:name>Al Brown</atom:name>
6424   </atom:author>
6425   <atom:content src="http://cmisexample.oasis-open.org/repl/dab97641-8c94-
6426 4a12-a604-7532980f05cb"/>
6427   <atom:id>urn:uuid:dab97641-8c94-4a12-a604-7532980f05cb</atom:id>
6428   <atom:title type="text">New Relationship</atom:title>
6429   <atom:updated>2010-01-25T10:20:58.864-08:00</atom:updated>
6430   <cmisra:object>
6431     <cmis:properties>
6432       <cmis:propertyId localName="rep-cmis:objectTypeId"
6433 propertyDefinitionId="cmis:objectTypeId">
6434         <cmis:value>customerRelationships</cmis:value>
6435       </cmis:propertyId>
6436       <cmis:propertyId localName="rep-cmis:sourceId"
6437 propertyDefinitionId="cmis:sourceId">
6438         <cmis:value>dbf0316c-47b5-47c9-a2fa-f005eb93f0a4</cmis:value>
6439       </cmis:propertyId>
6440       <cmis:propertyId localName="rep-cmis:targetId"
6441 propertyDefinitionId="cmis:targetId">
6442         <cmis:value>b9baac7d-7584-445e-bcd1-29af9b25bf2f</cmis:value>
6443       </cmis:propertyId>
6444     </cmis:properties>
6445   </cmisra:object>
6446 </atom:entry>
6447
```

6448

6449 Example server response:

```
6450 HTTP/1.1 201 Created
6451 Date: Mon, 25 Jan 2010 10:20:58 -0800
6452 Content-Length: 4684
6453 Content-Type: application/atom+xml;type=entry
6454 Content-Location: http://cmisexample.oasis-open.org/repl/b3006a8f-345b-4c27-
6455 86df-3f4b157bb495
6456 Location: http://cmisexample.oasis-open.org/repl/b3006a8f-345b-4c27-86df-
6457 3f4b157bb495
6458
6459
6460 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6461 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6462 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6463 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6464 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6465 open.org/ns/cmisis/restatom/200908/">
6466   <atom:author>
6467     <atom:name>Al Brown</atom:name>
6468     <atom:uri>http://www.ibm.com/</atom:uri>
6469     <atom:email>albertcbrown@us.ibm.com</atom:email>
6470   </atom:author>
```

```

6471 | <atom:content src="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-
6472 | 4c27-86df-3f4b157bb495"/>
6473 | <atom:id>urn:uuid:b3006a8f-345b-4c27-86df-3f4b157bb495</atom:id>
6474 | <atom:title type="text">New Relationship</atom:title>
6475 | <atom:updated>2010-01-25T10:20:58.880-08:00</atom:updated>
6476 | <atom:link rel="self" href="http://cmisexample.oasis-
6477 | open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
6478 | <atom:link rel="edit" href="http://cmisexample.oasis-
6479 | open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
6480 | <atom:link type="application/cmism+xml;type=allowableActions"
6481 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
6482 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6483 | 3f4b157bb495/allowableactions"/>
6484 | <atom:link type="application/atom+xml;type=entry" rel="describedby"
6485 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6486 | 3f4b157bb495/type"/>
6487 | <atom:link type="application/atomsvc+xml" rel="service"
6488 | href="http://cmisexample.oasis-open.org/rep1//service"/>
6489 | <atom:published>2010-01-25T10:20:58.880-08:00</atom:published>
6490 | <atom:summary type="html">HTML summary of Entry b3006a8f-345b-4c27-
6491 | 86df-3f4b157bb495</atom:summary>
6492 | <atom:link type="application/atom+xml;type=entry"
6493 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/source"
6494 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6495 | 3f4b157bb495/source"/>
6496 | <atom:link type="application/atom+xml;type=entry"
6497 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/target"
6498 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6499 | 3f4b157bb495/target"/>
6500 | <atom:link type="application/atom+xml;type=feed"
6501 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
6502 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6503 | 3f4b157bb495/policies"/>
6504 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6505 | open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
6506 | open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/acl"/>
6507 | <cmisra:object>
6508 | <cmis:properties>
6509 | <cmis:propertyId localName="rep-cmis:objectId"
6510 | propertyDefinitionId="cmis:objectId">
6511 | <cmis:value>b3006a8f-345b-4c27-86df-3f4b157bb495</cmis:value>
6512 | </cmis:propertyId>
6513 | <cmis:propertyId localName="rep-cmis:objectTypeId"
6514 | propertyDefinitionId="cmis:objectTypeId">
6515 | <cmis:value>customerRelationships</cmis:value>
6516 | </cmis:propertyId>
6517 | <cmis:propertyString localName="rep-cmis:name"
6518 | propertyDefinitionId="cmis:name">
6519 | <cmis:value>New Relationship</cmis:value>
6520 | </cmis:propertyString>
6521 | <cmis:propertyDateTime localName="rep-cmis:creationDate"
6522 | propertyDefinitionId="cmis:creationDate">
6523 | <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
6524 | </cmis:propertyDateTime>
6525 | <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6526 | propertyDefinitionId="cmis:lastModificationDate">
6527 | <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
6528 | </cmis:propertyDateTime>
6529 | <cmis:propertyId localName="rep-cmis:baseTypeId"
6530 | propertyDefinitionId="cmis:baseTypeId">
6531 | <cmis:value>cmis:relationship</cmis:value>
6532 | </cmis:propertyId>
6533 | <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6534 | propertyDefinitionId="cmis:lastModifiedBy">

```



```

6535         <cmis:value>Al Brown</cmis:value>
6536     </cmis:propertyString>
6537     <cmis:propertyString localName=""_rep-cmis:createdBy""
6538 propertyDefinitionId=""_cmis:createdBy"">
6539         <cmis:value>Al Brown</cmis:value>
6540     </cmis:propertyString>
6541     <cmis:propertyId localName=""_rep-cmis:sourceId""
6542 propertyDefinitionId=""_cmis:sourceId"">
6543         <cmis:value>d4551c6d-30bd-4fc2-9c84-a55f11559e89</cmis:value>
6544     </cmis:propertyId>
6545     <cmis:propertyId localName=""_rep-cmis:targetId""
6546 propertyDefinitionId=""_cmis:targetId"">
6547         <cmis:value>fe7e056f-c4bf-42f1-a03e-3ababcf2491d</cmis:value>
6548     </cmis:propertyId>
6549 </cmis:properties>
6550 </cmisra:object>
6551 </atom:entry>
6552

```

6553

6554 Please also see the example documents included with the schema.

6555

6556 3.8.2 Folder Children Collection

6557 This is a collection comprised of all the direct children of a particular folder represented as a feed.

6558 CMIS Services:

6559 GET: getChildren

6560 POST:

6561 createDocument

6562 or createFolder

6563 or createPolicy

6564 or moveObject

6565 or addObjectToFolder

6566

6567 Media Type: application/atom+xml;type=feed

6568

6569 Accept:

- 6570 • MUST support Atom Entry Documents with CMIS extensions

- 6571 • MAY support other media type

6572

6573 Link Relations:

- 6574 • service: Points to service document containing the CMIS repository. The service document
6575 MUST contain only one workspace element.

- 6576 ○ Media Type: application/atomsvc+xml

- 6577 • via: points to the atom entry of the folder generating this collection

- 6578 • up: points to the atom entry document for this folder's parent

- 6579 ○ If the root folder, this link relation MUST NOT be included.

- 6580 ○ Media Type: application/atom+xml;type=entry

- 6581 • down: points to the atom feed document representing the descendents feed with a media type of
6582 application/cmistree+xml

- 6583 ○ If a repository does not support capabilityGetDescendants, then this link SHOULD NOT
6584 be included.
- 6585 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder.
6586 This is represented as a feed with CMIS hierarchy extensions.
- 6587 ○ Media Type: application/atom+xml;type=feed
- 6588 • paging link relations as appropriate: first, next, previous, last

6589

6590 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6591 • cmisra:numItems

6592

6593 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6594 • cmisra:object inside atom:entry
- 6595 • cmisra:pathSegment inside atom:entry if pathSegment is not false

6596

6597 **3.8.2.1 GET**

6598 HTTP Code:

- 6599 • 200 OK (Success)

6600

6601 The following arguments may be supplied. Please see the domain model for more information:

- 6602 • maxItems
- 6603 • skipCount
- 6604 • filter
- 6605 • includeAllowableActions
- 6606 • includeRelationships
- 6607 • renditionFilter
- 6608 ○ If specified, renditions will be returned as links with relation alternate.
- 6609 • orderBy
- 6610 • includePathSegment

6611 **3.8.2.2 POST**

6612 CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please
6613 see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 6614 • HTTP Success: 201
- 6615 • Location Header

6616

6617 The following arguments MAY be supplied.

- 6618 • sourceFolderId: This parameter indicates the folder from which the object shall be moved from to
6619 the current specified folder. This parameter is not allowed for create operations.
- 6620 ○ If specified moveObject will be performed.
- 6621 ○ If not specified, addObjectToFolder will be performed.
- 6622 • versioningState: The optional argument versioningState MAY specify additional versioning
6623 behavior such as checkIn as major or minor. Please see CMIS Domain Model for more
6624 information on this parameter.

6625

6626 POSTing an Atom Entry document with CMIS markup:

6627 Adding a document to a folder:
6628 If the atom entry has a cmis property cmis:objectId that is valid for the repository, the object will
6629 be added to the folder.

6630
6631 When an object is added to the folder, in repositories that do not support multi-filing it will be
6632 removed from the previous folder and the operation treated as move. If the repository supports
6633 multiple folders, it will be added to the new folder.

6634 If the optional argument sourceFolderId is specified, then the object will be removed from the
6635 folder specified.

6636
6637 If atom:content is missing from the request, the repository MUST treat the missing atom:content
6638 element as an empty atom:content element.

6639 Example client request:

```
6640 POST /obj/1cd0d82f-d579-4897-9b0a-ad0917595445?sourceFolderId=313fd58d-2eab-
6641 41af-9517-06dadb010d49 HTTP/1.1
6642 Host: example.org
6643 Content-Length: 1227
6644 Content-Type: application/atom+xml;type=entry
```

```
6645
6646
6647 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6648 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6649 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6650 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6651 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6652 open.org/ns/cmis/restatom/200908/">
6653   <atom:author>
6654     <atom:name>Al Brown</atom:name>
6655   </atom:author>
6656   <atom:id>urn:uuid:1cd0d82f-d579-4897-9b0a-ad0917595445</atom:id>
6657   <atom:title type="text">Document - To Be Moved</atom:title>
6658   <atom:updated>2010-01-25T10:20:58.708-08:00</atom:updated>
6659   <atom:content src="http://cmisexample.oasis-
6660 open.org/rep1/content/1cd0d82f-d579-4897-9b0a-ad0917595445"/>
6661   <cmisra:object>
6662     <cmis:properties>
6663       <cmis:propertyId localName="rep-cmis:objectId"
6664 propertyDefinitionId="cmis:objectId">
6665         <cmis:value>1cd0d82f-d579-4897-9b0a-ad0917595445</cmis:value>
6666       </cmis:propertyId>
6667       <cmis:propertyId localName="rep-cmis:objectTypeId"
6668 propertyDefinitionId="cmis:objectTypeId">
6669         <cmis:value>invoice</cmis:value>
6670       </cmis:propertyId>
6671     </cmis:properties>
6672   </cmisra:object>
6673 </atom:entry>
```

6675
6676 Example server response:

```
6677 HTTP/1.1 201 Created
6678 Date: Mon, 25 Jan 2010 10:20:58 -0800
6679 Content-Length: 7213
6680 Content-Type: application/atom+xml;type=entry
6681 Content-Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-
6682 8141-4aed91d28b5b
6683 Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
6684 4aed91d28b5b
```

6685
6686
6687
6688
6689
6690
6691
6692
6693
6694
6695
6696
6697
6698
6699
6700
6701
6702
6703
6704
6705
6706
6707
6708
6709
6710
6711
6712
6713
6714
6715
6716
6717
6718
6719
6720
6721
6722
6723
6724
6725
6726
6727
6728
6729
6730
6731
6732
6733
6734
6735
6736
6737
6738
6739
6740
6741
6742
6743
6744
6745
6746
6747
6748

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:entry xmlns:app="http://www.w3.org/2007/app"
xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
open.org/ns/cmismessaging/200908/">
  <atom:author>
    <atom:name>Al Brown</atom:name>
    <atom:uri>http://www.ibm.com/</atom:uri>
    <atom:email>albertcbrown@us.ibm.com</atom:email>
  </atom:author>
  <atom:content src="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-
49fb-8141-4aed91d28b5b"/>
  <atom:id>urn:uuid:b4423b8a-e46e-49fb-8141-4aed91d28b5b</atom:id>
  <atom:title type="text">Document - To Be Moved</atom:title>
  <atom:updated>2010-01-25T10:20:58.786-08:00</atom:updated>
  <atom:link rel="self" href="http://cmisexample.oasis-
open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b"/>
  <atom:link rel="edit" href="http://cmisexample.oasis-
open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b"/>
  <atom:link type="application/cmismessaging+xml;type=allowableActions"
rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/allowableactions"
href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
4aed91d28b5b/allowableactions"/>
  <atom:link type="application/atom+xml;type=entry" rel="describedby"
href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
4aed91d28b5b/type"/>
  <atom:link type="application/atomsvc+xml" rel="service"
href="http://cmisexample.oasis-open.org/rep1//service"/>
  <atom:published>2010-01-25T10:20:58.786-08:00</atom:published>
  <atom:summary type="html">HTML summary of Entry b4423b8a-e46e-49fb-
8141-4aed91d28b5b</atom:summary>
  <atom:link rel="edit-media" href="http://cmisexample.oasis-
open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/edit-media"/>
  <atom:link rel="alternate" href="http://cmisexample.oasis-
open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/alternate"/>
  <atom:link type="application/atom+xml;type=feed" rel="up"
href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
4aed91d28b5b/parents"/>
  <atom:link type="application/atom+xml;type=feed" rel="version-history"
href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
4aed91d28b5b/allversions"/>
  <atom:link type="application/atom+xml;type=entry" rel="current-
version" href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-
8141-4aed91d28b5b/latest"/>
  <atom:link type="application/atom+xml;type=feed"
rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/relationships"
href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
4aed91d28b5b/relationships"/>
  <atom:link type="application/atom+xml;type=feed"
rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/policies"
href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
4aed91d28b5b/policies"/>
  <atom:link type="application/cmismessaging+acl+xml" rel="http://docs.oasis-
open.org/ns/cmismessaging/link/200908/acl" href="http://cmisexample.oasis-
open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/acl"/>
  <cmisra:object>
    <cmis:properties>
      <cmis:propertyId localName="rep-cmis:objectId"
propertyDefinitionId="cmis:objectId">
        <cmis:value>b4423b8a-e46e-49fb-8141-4aed91d28b5b</cmis:value>
      </cmis:propertyId>
    </cmis:properties>
  </cmisra:object>
</atom:entry>
```

```

6749 |         <cmis:propertyId localName=""rep-cmis:objectId""
6750 | propertyDefinitionId=""cmis:objectId"">
6751 |             <cmis:value>invoice</cmis:value>
6752 |         </cmis:propertyId>
6753 |         <cmis:propertyString localName=""rep-cmis:name""
6754 | propertyDefinitionId=""cmis:name"">
6755 |             <cmis:value>Document - To Be Moved</cmis:value>
6756 |         </cmis:propertyString>
6757 |         <cmis:propertyDateTime localName=""rep-cmis:creationDate""
6758 | propertyDefinitionId=""cmis:creationDate"">
6759 |             <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
6760 |         </cmis:propertyDateTime>
6761 |         <cmis:propertyDateTime localName=""rep-cmis:lastModificationDate""
6762 | propertyDefinitionId=""cmis:lastModificationDate"">
6763 |             <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
6764 |         </cmis:propertyDateTime>
6765 |         <cmis:propertyId localName=""rep-cmis:baseTypeId""
6766 | propertyDefinitionId=""cmis:baseTypeId"">
6767 |             <cmis:value>cmis:document</cmis:value>
6768 |         </cmis:propertyId>
6769 |         <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
6770 | propertyDefinitionId=""cmis:lastModifiedBy"">
6771 |             <cmis:value>Al Brown</cmis:value>
6772 |         </cmis:propertyString>
6773 |         <cmis:propertyString localName=""rep-cmis:createdBy""
6774 | propertyDefinitionId=""cmis:createdBy"">
6775 |             <cmis:value>Al Brown</cmis:value>
6776 |         </cmis:propertyString>
6777 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
6778 | propertyDefinitionId=""cmis:isLatestVersion"">
6779 |             <cmis:value>>true</cmis:value>
6780 |         </cmis:propertyBoolean>
6781 |         <cmis:propertyBoolean localName=""rep-
6782 | cmis:isVersionSeriesCheckedOut""
6783 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
6784 |             <cmis:value>>false</cmis:value>
6785 |         </cmis:propertyBoolean>
6786 |         <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
6787 | propertyDefinitionId=""cmis:isMajorVersion"">
6788 |             <cmis:value>>false</cmis:value>
6789 |         </cmis:propertyBoolean>
6790 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
6791 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
6792 |             <cmis:value>>false</cmis:value>
6793 |         </cmis:propertyBoolean>
6794 |         <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
6795 | propertyDefinitionId=""cmis:isImmutable"">
6796 |             <cmis:value>>false</cmis:value>
6797 |         </cmis:propertyBoolean>
6798 |         <cmis:propertyString localName=""rep-cmis:checkinComment""
6799 | propertyDefinitionId=""cmis:checkinComment"">
6800 |             <cmis:value>Checkin comment</cmis:value>
6801 |         </cmis:propertyString>
6802 |         <cmis:propertyString localName=""rep-cmis:versionLabel""
6803 | propertyDefinitionId=""cmis:versionLabel"">
6804 |             <cmis:value>0.1</cmis:value>
6805 |         </cmis:propertyString>
6806 |         <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
6807 | propertyDefinitionId=""cmis:contentStreamMimeType"">
6808 |             <cmis:value>text/plain</cmis:value>
6809 |         </cmis:propertyString>
6810 |         <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
6811 | propertyDefinitionId=""cmis:contentStreamFileName"">
6812 |             <cmis:value>text.txt</cmis:value>

```

```

6813         </cmis:propertyString>
6814         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6815 propertyDefinitionId="cmis:contentStreamLength"
6816         <cmis:value>4234</cmis:value>
6817     </cmis:propertyInteger>
6818     <cmis:propertyString displayName="Keywords for Document"
6819 localName="keywords" propertyDefinitionId="keywords">
6820         <cmis:value>document</cmis:value>
6821         <cmis:value>example</cmis:value>
6822         <cmis:value>sample</cmis:value>
6823         <cmis:value>cmis</cmis:value>
6824     </cmis:propertyString>
6825 </cmis:properties>
6826 </cmisra:object>
6827 </atom:entry>
6828

```

6829

6830 Please also see the example documents included with the schema.

6831

6832 Creating a CMIS Object (in that folder):

6833 If the cmis:objectId property is missing, the object will be created and then added to the folder. If
6834 the cmis:objectId property is present but not a valid object Id, the repository MUST return the
6835 appropriate HTTP status code.

6836

6837 For Documents:

6838 If Content Stream is not provided and it is required by the type definition, the repository
6839 MUST return the appropriate HTTP status code.

6840

6841 Content Streams MAY be provided by any of the following mechanisms:

- 6842 ○ As part of the atom entry via the src attribute on the content element (AtomPub)
 - 6843 ■ src attribute: Implementers MAY support external references to content
 - 6844 ■ If the URI in the src attribute is not reachable, then an appropriate http
 - 6845 status code should be returned.
- 6846 ○ As part of the atom entry inlining via the content element (AtomPub)
 - 6847 ■ Please see the AtomPub specification RFC5023 for the processing
 - 6848 model of the content element.
- 6849 ○ If the cmisra:content is provided by the client inside the atom:entry, the
6850 cmisra:content element MUST take precedence over the atom:content element.
6851 (CMIS)
 - 6852 ● This element cmisra:content is base64 encoded
- 6853 ○ At a later time (AtomPub)
 - 6854 ■ At a later time by replacing the edit-media link with a new content

6855

6856 The optional argument versioningState MAY specify additional versioning behavior such
6857 as checkin.

6858

6859 Example client request:

```

6860 POST /obj/bb2b208b-3acd-4abe-9788-8078a239f228 HTTP/1.1
6861 Host: example.org
6862 Content-Length: 1190
6863 Content-Type: application/atom+xml;type=entry

```

```

6864
6865
6866 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6867 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6868 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6869 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6870 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6871 open.org/ns/cmisis/restatom/200908/">
6872   <atom:author>
6873     <atom:name>Al Brown</atom:name>
6874   </atom:author>
6875   <atom:id>urn:uuid:bb2b208b-3acd-4abe-9788-8078a239f228</atom:id>
6876   <atom:title type="text">New Invoice</atom:title>
6877   <atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
6878   <atom:content type="text">this is the content of the new
6879 document</atom:content>
6880   <cmisra:object>
6881     <cmis:properties>
6882       <cmis:propertyId localName="rep-cmis:objectId"
6883 propertyDefinitionId="cmis:objectId">
6884         <cmis:value>bb2b208b-3acd-4abe-9788-8078a239f228</cmis:value>
6885       </cmis:propertyId>
6886       <cmis:propertyId localName="rep-cmis:objectTypeId"
6887 propertyDefinitionId="cmis:objectTypeId">
6888         <cmis:value>invoice</cmis:value>
6889       </cmis:propertyId>
6890     </cmis:properties>
6891   </cmisra:object>
6892 </atom:entry>
6893

```

6894

6895 **Example server response:**

```

6896 HTTP/1.1 201 Created
6897 Date: Mon, 25 Jan 2010 10:20:58 -0800
6898 Content-Length: 7191
6899 Content-Type: application/atom+xml;type=entry
6900 Content-Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-
6901 ad0b-10ea94c4b93d
6902 Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6903 10ea94c4b93d
6904
6905
6906 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6907 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6908 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6909 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6910 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6911 open.org/ns/cmisis/restatom/200908/">
6912   <atom:author>
6913     <atom:name>Al Brown</atom:name>
6914     <atom:uri>http://www.ibm.com/</atom:uri>
6915     <atom:email>albertcbrown@us.ibm.com</atom:email>
6916   </atom:author>
6917   <atom:content src="http://cmisexample.oasis-open.org/rep1/13475008-6a20-
6918 4454-ad0b-10ea94c4b93d"/>
6919   <atom:id>urn:uuid:13475008-6a20-4454-ad0b-10ea94c4b93d</atom:id>
6920   <atom:title type="text">New Invoice</atom:title>
6921   <atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
6922   <atom:link rel="self" href="http://cmisexample.oasis-
6923 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>
6924   <atom:link rel="edit" href="http://cmisexample.oasis-
6925 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>

```

```

6926 |     <atom:link type=""application/cmism+xml;type=allowableActions""
6927 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions""
6928 | href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6929 | 10ea94c4b93d/allowableactions""/>
6930 |     <atom:link type=""application/atom+xml;type=entry"" rel=""describedby""
6931 | href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6932 | 10ea94c4b93d/type""/>
6933 |     <atom:link type=""application/atomsvc+xml"" rel=""service""
6934 | href=""http://cmismexample.oasis-open.org/rep1//service""/>
6935 |     <atom:published>2010-01-25T10:20:58.833-08:00</atom:published>
6936 |     <atom:summary type=""html"">HTML summary of Entry 13475008-6a20-4454-
6937 | ad0b-10ea94c4b93d</atom:summary>
6938 |     <atom:link rel=""edit-media"" href=""http://cmismexample.oasis-
6939 | open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/edit-media""/>
6940 |     <atom:link rel=""alternate"" href=""http://cmismexample.oasis-
6941 | open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/alternate""/>
6942 |     <atom:link type=""application/atom+xml;type=feed"" rel=""up""
6943 | href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6944 | 10ea94c4b93d/parents""/>
6945 |     <atom:link type=""application/atom+xml;type=feed"" rel=""version-history""
6946 | href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6947 | 10ea94c4b93d/allversions""/>
6948 |     <atom:link type=""application/atom+xml;type=entry"" rel=""current-
6949 | version"" href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-
6950 | ad0b-10ea94c4b93d/latest""/>
6951 |     <atom:link type=""application/atom+xml;type=feed""
6952 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/relationships""
6953 | href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6954 | 10ea94c4b93d/relationships""/>
6955 |     <atom:link type=""application/atom+xml;type=feed""
6956 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/policies""
6957 | href=""http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6958 | 10ea94c4b93d/policies""/>
6959 |     <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
6960 | open.org/ns/cmism/link/200908/acl"" href=""http://cmismexample.oasis-
6961 | open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/acl""/>
6962 |     <cmisra:object>
6963 |         <cmis:properties>
6964 |             <cmis:propertyId localName=""rep-cmism:objectId""
6965 | propertyDefinitionId=""cmism:objectId"">
6966 |                 <cmis:value>13475008-6a20-4454-ad0b-10ea94c4b93d</cmis:value>
6967 |             </cmis:propertyId>
6968 |             <cmis:propertyId localName=""rep-cmism:objectTypeId""
6969 | propertyDefinitionId=""cmism:objectTypeId"">
6970 |                 <cmis:value>invoice</cmis:value>
6971 |             </cmis:propertyId>
6972 |             <cmis:propertyString localName=""rep-cmism:name""
6973 | propertyDefinitionId=""cmism:name"">
6974 |                 <cmis:value>New Invoice</cmis:value>
6975 |             </cmis:propertyString>
6976 |             <cmis:propertyDateTime localName=""rep-cmism:creationDate""
6977 | propertyDefinitionId=""cmism:creationDate"">
6978 |                 <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
6979 |             </cmis:propertyDateTime>
6980 |             <cmis:propertyDateTime localName=""rep-cmism:lastModificationDate""
6981 | propertyDefinitionId=""cmism:lastModificationDate"">
6982 |                 <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
6983 |             </cmis:propertyDateTime>
6984 |             <cmis:propertyId localName=""rep-cmism:baseTypeId""
6985 | propertyDefinitionId=""cmism:baseTypeId"">
6986 |                 <cmis:value>cmism:document</cmis:value>
6987 |             </cmis:propertyId>
6988 |             <cmis:propertyString localName=""rep-cmism:lastModifiedBy""
6989 | propertyDefinitionId=""cmism:lastModifiedBy"">

```



```

6990         <cmis:value>Al Brown</cmis:value>
6991     </cmis:propertyString>
6992     <cmis:propertyString localName="rep-cmis:createdBy"
6993 | propertyDefinitionId="cmis:createdBy">
6994         <cmis:value>Al Brown</cmis:value>
6995     </cmis:propertyString>
6996     <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6997 | propertyDefinitionId="cmis:isLatestVersion">
6998         <cmis:value>>true</cmis:value>
6999     </cmis:propertyBoolean>
7000     <cmis:propertyBoolean localName="rep-
7001 | cmis:isVersionSeriesCheckedOut"
7002 | propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7003         <cmis:value>>false</cmis:value>
7004     </cmis:propertyBoolean>
7005     <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
7006 | propertyDefinitionId="cmis:isMajorVersion">
7007         <cmis:value>>false</cmis:value>
7008     </cmis:propertyBoolean>
7009     <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
7010 | propertyDefinitionId="cmis:isLatestMajorVersion">
7011         <cmis:value>>false</cmis:value>
7012     </cmis:propertyBoolean>
7013     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7014 | propertyDefinitionId="cmis:isImmutable">
7015         <cmis:value>>false</cmis:value>
7016     </cmis:propertyBoolean>
7017     <cmis:propertyString localName="rep-cmis:checkinComment"
7018 | propertyDefinitionId="cmis:checkinComment">
7019         <cmis:value>Checkin comment</cmis:value>
7020     </cmis:propertyString>
7021     <cmis:propertyString localName="rep-cmis:versionLabel"
7022 | propertyDefinitionId="cmis:versionLabel">
7023         <cmis:value>0.1</cmis:value>
7024     </cmis:propertyString>
7025     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
7026 | propertyDefinitionId="cmis:contentStreamMimeType">
7027         <cmis:value>text/plain</cmis:value>
7028     </cmis:propertyString>
7029     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
7030 | propertyDefinitionId="cmis:contentStreamFileName">
7031         <cmis:value>text.txt</cmis:value>
7032     </cmis:propertyString>
7033     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7034 | propertyDefinitionId="cmis:contentStreamLength">
7035         <cmis:value>4234</cmis:value>
7036     </cmis:propertyInteger>
7037     <cmis:propertyString displayName="Keywords for Document"
7038 | localName="keywords" propertyDefinitionId="keywords">
7039         <cmis:value>document</cmis:value>
7040         <cmis:value>example</cmis:value>
7041         <cmis:value>sample</cmis:value>
7042         <cmis:value>cmis</cmis:value>
7043     </cmis:propertyString>
7044 </cmis:properties>
7045 </cmisra:object>
7046 </atom:entry>
7047

```

7048

7049 Please also see the example documents included with the schema.

7050

7051 POSTing other document formats: (AtomPub)

7052 The behavior is repository specific when a non Atom entry or an atom document without the
7053 CMIS elements is posted to a folder collection.
7054 For example, the repository MAY auto-create a document with a specific type (document) the
7055 client could edit.
7056 If the repository does not support this scenario or another exception occurs, then the repository
7057 MUST return the appropriate HTTP status code.

7058
7059 Optional arguments:

- 7060 • versioningState (for createDocument)
- 7061 • sourceFolderId (for moveObject)

7062

7063 3.8.3 Policies Collection

7064 This is an atom feed of all the policy objects currently applied to a specific object. This is the only
7065 collection where the URIs of the objects in the collection MUST be specific to that collection. A DELETE
7066 on the policy object in the collection is a removal of the policy from the object NOT a deletion of the policy
7067 object itself.

7068

7069 CMIS Services:

- 7070 GET: getAppliedPolicies
- 7071 POST: applyPolicy (to object representing this collection of policies)
- 7072 DELETE: removePolicy

7073 Media Type: application/atom+xml;type=feed

7074 Accept:

- 7075 • MUST support Atom Entry Documents with CMIS extensions
 - 7076 ○ application/atom+xml;type=entry or
 - 7077 ○ application/cmisaatom+xml
- 7078 • MAY support other media type

7079

7080 Link Relations:

- 7081 • service: Points to service document containing the CMIS repository. The service document
7082 MUST contain only one workspace element.
 - 7083 ○ Media Type: application/atomsvc+xml
- 7084 • via: points to the atom entry of the resource generating this collection
- 7085 • paging link relations as appropriate: first, next, previous, last

7086

7087 The policy entries displayed here are specific to the object generating this collection. A DELETE method
7088 on those URIs will invoke removePolicy().

7089

7090 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7091 • cmisra:numItems

7092

7093 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7094 • cmisra:object inside atom:entry

7095

7096 3.8.3.1 GET

7097 The following arguments may be supplied. Please see the domain model for more information:

- 7098 • filter

7099 3.8.3.2 POST

7100 When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the
7101 object.

7102

7103 Example client request:

```
7104 POST /policies/f3670f66-62ee-487f-b733-999a69237024 HTTP/1.1
7105 Host: example.org
7106 Content-Length: 1039
7107 Content-Type: application/atom+xml;type=entry
7108
7109
7110 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7111 <atom:entry xmlns:app="http://www.w3.org/2007/app"
7112 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
7113 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
7114 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
7115 open.org/ns/cmisis/restatom/200908/">
7116   <atom:author>
7117     <atom:name>Al Brown</atom:name>
7118   </atom:author>
7119   <atom:content src="http://cmisexample.oasis-open.org/rep1/f3670f66-62ee-
7120 487f-b733-999a69237024"/>
7121   <atom:id>urn:uuid:f3670f66-62ee-487f-b733-999a69237024</atom:id>
7122   <atom:title type="text">Security Policy for Invoices</atom:title>
7123   <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
7124   <cmisra:object>
7125     <cmis:properties>
7126       <cmis:propertyId localName="rep-cmis:objectId"
7127 propertyDefinitionId="cmis:objectId">
7128         <cmis:value>f3670f66-62ee-487f-b733-999a69237024</cmis:value>
7129       </cmis:propertyId>
7130     </cmis:properties>
7131   </cmisra:object>
7132 </atom:entry>
7133
```

7134

7135 Example server response:

```
7136 HTTP/1.1 201 Created
7137 Date: Mon, 25 Jan 2010 10:20:58 -0800
7138 Content-Length: 4043
7139 Content-Type: application/atom+xml;type=entry
7140 Content-Location: http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-
7141 bdf6-690761576116
7142 Location: http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-bdf6-
7143 690761576116
7144
7145
7146 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

7147 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
7148 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
7149 | open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
7150 | open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
7151 | open.org/ns/cmisis/restatom/200908/">
7152 |   <atom:author>
7153 |     <atom:name>Al Brown</atom:name>
7154 |     <atom:uri>http://www.ibm.com/</atom:uri>
7155 |     <atom:email>albertcbrown@us.ibm.com</atom:email>
7156 |   </atom:author>
7157 |   <atom:content src="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-
7158 | 4354-bdfe-690761576116"/>
7159 |   <atom:id>urn:uuid:55cca51b-6cfa-4354-bdfe-690761576116</atom:id>
7160 |   <atom:title type="text">Security Policy for Invoices</atom:title>
7161 |   <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
7162 |   <atom:link rel="self" href="http://cmisexample.oasis-
7163 | open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116"/>
7164 |   <atom:link rel="edit" href="http://cmisexample.oasis-
7165 | open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116"/>
7166 |   <atom:link type="application/cmisis+xml;type=allowableActions"
7167 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7168 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
7169 | 690761576116/allowableactions"/>
7170 |   <atom:link type="application/atom+xml;type=entry" rel="describedby"
7171 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
7172 | 690761576116/type"/>
7173 |   <atom:link type="application/atomsvc+xml" rel="service"
7174 | href="http://cmisexample.oasis-open.org/repl//service"/>
7175 |   <atom:published>2010-01-25T10:20:58.849-08:00</atom:published>
7176 |   <atom:summary type="html">HTML summary of Entry 55cca51b-6cfa-4354-
7177 | bdfe-690761576116</atom:summary>
7178 |   <atom:link type="application/atom+xml;type=feed" rel="up"
7179 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
7180 | 690761576116/parents"/>
7181 |   <atom:link type="application/atom+xml;type=feed"
7182 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
7183 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
7184 | 690761576116/relationships"/>
7185 |   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
7186 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
7187 | open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116/acl"/>
7188 |   <cmisra:object>
7189 |     <cmis:properties>
7190 |       <cmis:propertyId localName="rep-cmisis:objectId"
7191 | propertyDefinitionId="cmisis:objectId">
7192 |         <cmis:value>55cca51b-6cfa-4354-bdfe-690761576116</cmis:value>
7193 |       </cmis:propertyId>
7194 |       <cmis:propertyId localName="rep-cmisis:objectTypeId"
7195 | propertyDefinitionId="cmisis:objectTypeId">
7196 |         <cmis:value>generalSecurityPolicy</cmis:value>
7197 |       </cmis:propertyId>
7198 |       <cmis:propertyString localName="rep-cmisis:name"
7199 | propertyDefinitionId="cmisis:name">
7200 |         <cmis:value>Security Policy for Invoices</cmis:value>
7201 |       </cmis:propertyString>
7202 |       <cmis:propertyDateTime localName="rep-cmisis:creationDate"
7203 | propertyDefinitionId="cmisis:creationDate">
7204 |         <cmis:value>2010-01-25T10:20:58.849-08:00</cmis:value>
7205 |       </cmis:propertyDateTime>
7206 |       <cmis:propertyDateTime localName="rep-cmisis:lastModificationDate"
7207 | propertyDefinitionId="cmisis:lastModificationDate">
7208 |         <cmis:value>2010-01-25T10:20:58.864-08:00</cmis:value>
7209 |       </cmis:propertyDateTime>

```

```

7210 |         <cmis:propertyId localName=""rep-cmis:baseTypeId""
7211 | propertyDefinitionId=""cmis:baseTypeId"">
7212 |             <cmis:value>cmis:policy</cmis:value>
7213 |         </cmis:propertyId>
7214 |         <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
7215 | propertyDefinitionId=""cmis:lastModifiedBy"">
7216 |             <cmis:value>Al Brown</cmis:value>
7217 |         </cmis:propertyString>
7218 |         <cmis:propertyString localName=""rep-cmis:createdBy""
7219 | propertyDefinitionId=""cmis:createdBy"">
7220 |             <cmis:value>Al Brown</cmis:value>
7221 |         </cmis:propertyString>
7222 |     </cmis:properties>
7223 | </cmisra:object>
7224 | </atom:entry>
7225 |

```

7226

7227 Please also see the example documents included with the schema.

7228 **3.8.3.3 DELETE**

7229 | This is the only collection where the URI's of the objects in the collection MUST be specific to that
7230 | collection. A DELETE on the policy object in the collection is a removal of the policy from the object NOT
7231 | a deletion of the policy object itself.

7232

7233 **3.9 Feeds**

7234 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
7235 verb in a repository-specific manner.

7236 **3.9.1 Object Parents Feed**

7237 This is the set of parents for a specific object.

7238 CMIS Services:

7239 GET: getObjectParents

7240 Media Type: application/atom+xml;type=feed

7241

7242 Link Relations:

- 7243 • service: Points to service document containing the CMIS repository. The service document
7244 MUST contain only one workspace element.
 - 7245 ○ Media Type: application/atomsvc+xml
- 7246 • via: points to the atom entry of object who's parents are represented by this collection

7247

7248 This feed contains a set of atom entries for each parent of the object that MUST contain:

- 7249 • cmisra:object inside atom:entry
- 7250 • cmisra:relativePathSegment inside atom:entry for the name of the object inside the folder

7251

7252 Example:

```

7253 | <?xml version=""1.0"" encoding=""UTF-8"" standalone=""yes""?>

```

7254 | <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7255 | xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7256 | xmlns:atom="http://www.w3.org/2005/Atom"
7257 | xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7258 | open.org/ns/cmis/restatom/200908/">
7259 | <atom:title type="text">Parent Feed for 268d30b5-91a0-47f0-b985-
7260 | 6765e178f0bb</atom:title>
7261 | <atom:author>
7262 | <atom:name>Al Brown</atom:name>
7263 | <atom:uri>http://www.ibm.com/</atom:uri>
7264 | <atom:email>albertcbrown@us.ibm.com</atom:email>
7265 | </atom:author>
7266 | <atom:updated>2010-01-25T10:20:59.818-08:00</atom:updated>
7267 | <atom:id>urn:uuid:6f541940-4abf-471b-99f0-8e6f66d53789</atom:id>
7268 | <atom:link type="application/atom+xml;type=feed" rel="self"
7269 | href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7270 | 6765e178f0bb/3"/>
7271 | <atom:link type="application/atomsvc+xml" rel="service"
7272 | href="http://cmisexample.oasis-open.org/rep1//service"/>
7273 | <atom:link type="application/atom+xml;type=entry" rel="via"
7274 | href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7275 | 6765e178f0bb"/>
7276 | <atom:link type="application/atom+xml;type=feed" rel="first"
7277 | href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7278 | 6765e178f0bb/first"/>
7279 | <atom:link type="application/atom+xml;type=feed" rel="next"
7280 | href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7281 | 6765e178f0bb/4"/>
7282 | <atom:link type="application/atom+xml;type=feed" rel="previous"
7283 | href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7284 | 6765e178f0bb/2"/>
7285 | <atom:link type="application/atom+xml;type=feed" rel="last"
7286 | href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7287 | 6765e178f0bb/last"/>
7288 | <cmisra:numItems>1</cmisra:numItems>
7289 | <atom:entry>
7290 | <atom:author>
7291 | <atom:name>Al Brown</atom:name>
7292 | <atom:uri>http://www.ibm.com/</atom:uri>
7293 | <atom:email>albertcbrown@us.ibm.com</atom:email>
7294 | </atom:author>
7295 | <atom:content src="http://cmisexample.oasis-open.org/rep1/661d6945-
7296 | 8f75-4dea-8799-7ba07b0e510e"/>
7297 | <atom:id>urn:uuid:661d6945-8f75-4dea-8799-7ba07b0e510e</atom:id>
7298 | <atom:title type="text">Customer Folder</atom:title>
7299 | <atom:updated>2010-01-25T10:20:59.833-08:00</atom:updated>
7300 | <atom:link rel="self" href="http://cmisexample.oasis-
7301 | open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
7302 | <atom:link rel="edit" href="http://cmisexample.oasis-
7303 | open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
7304 | <atom:link type="application/cmis+xml;type=allowableActions"
7305 | rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7306 | href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7307 | 7ba07b0e510e/allowableactions"/>
7308 | <atom:link type="application/atom+xml;type=entry"
7309 | rel="describedby" href="http://cmisexample.oasis-open.org/rep1/661d6945-
7310 | 8f75-4dea-8799-7ba07b0e510e/type"/>
7311 | <atom:link type="application/atomsvc+xml" rel="service"
7312 | href="http://cmisexample.oasis-open.org/rep1//service"/>
7313 | <atom:published>2010-01-25T10:20:59.833-08:00</atom:published>
7314 | <atom:summary type="html">HTML summary of Entry 661d6945-8f75-4dea-
7315 | 8799-7ba07b0e510e</atom:summary>

```

7316     <atom:link type="application/atom+xml;type=entry" rel="up"
7317 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7318 7ba07b0e510e/up" />
7319     <atom:link type="application/atom+xml;type=feed" rel="down"
7320 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7321 7ba07b0e510e/children" />
7322     <atom:link type="application/cmistree+xml" rel="down"
7323 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7324 7ba07b0e510e/tree" />
7325     <atom:link type="application/atom+xml;type=feed"
7326 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7327 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7328 7ba07b0e510e/foldertree" />
7329     <atom:link type="application/atom+xml;type=feed"
7330 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
7331 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7332 7ba07b0e510e/relationships" />
7333     <atom:link type="application/atom+xml;type=feed"
7334 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
7335 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7336 7ba07b0e510e/policies" />
7337     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7338 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
7339 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e/acl" />
7340     <cmisra:object>
7341       <cmis:properties>
7342         <cmis:propertyId localName="rep-cmis:objectId"
7343 propertyDefinitionId="cmis:objectId">
7344           <cmis:value>661d6945-8f75-4dea-8799-
7345 7ba07b0e510e</cmis:value>
7346         </cmis:propertyId>
7347         <cmis:propertyId localName="rep-cmis:objectTypeId"
7348 propertyDefinitionId="cmis:objectTypeId">
7349           <cmis:value>customer</cmis:value>
7350         </cmis:propertyId>
7351         <cmis:propertyString localName="rep-cmis:name"
7352 propertyDefinitionId="cmis:name">
7353           <cmis:value>Customer Folder</cmis:value>
7354         </cmis:propertyString>
7355         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7356 propertyDefinitionId="cmis:creationDate">
7357           <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
7358         </cmis:propertyDateTime>
7359         <cmis:propertyDateTime localName="rep-
7360 cmis:lastModificationDate"
7361 propertyDefinitionId="cmis:lastModificationDate">
7362           <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
7363         </cmis:propertyDateTime>
7364         <cmis:propertyId localName="rep-cmis:baseTypeId"
7365 propertyDefinitionId="cmis:baseTypeId">
7366           <cmis:value>cmis:folder</cmis:value>
7367         </cmis:propertyId>
7368         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7369 propertyDefinitionId="cmis:lastModifiedBy">
7370           <cmis:value>Al Brown</cmis:value>
7371         </cmis:propertyString>
7372         <cmis:propertyString localName="rep-cmis:createdBy"
7373 propertyDefinitionId="cmis:createdBy">
7374           <cmis:value>Al Brown</cmis:value>
7375         </cmis:propertyString>
7376         <cmis:propertyId localName="rep-cmis:parentId"
7377 propertyDefinitionId="cmis:parentId">
7378           <cmis:value>661d6945-8f75-4dea-8799-
7379 7ba07b0e510eup</cmis:value>

```

```
7380         </cmis:propertyId>
7381         </cmis:properties>
7382     </cmisra:object>
7383     <cmisra:relativePathSegment>customer1</cmisra:relativePathSegment>
7384 </atom:entry>
7385 </atom:feed>
```

7386

7387 Please also see the example documents included with the schema.

7388 **3.9.1.1 GET**

7389 The following arguments may be supplied. Please see the domain model for more information:

- 7390 • filter
- 7391 • includeAllowableActions
- 7392 • includeRelationships
- 7393 • renditionFilter
- 7394 • includeRelativePathSegment
- 7395 ○ If true, then the cmisra:relativePathSegment element MUST be included in the response.

7396 **3.9.2 Changes**

7397 This is a link relationship described in the service document that contains the changes in the repository in
7398 the workspace element. The link relation pointing to this feed is [http://docs.oasis-
open.org/ns/cmis/link/200908/changes](http://docs.oasis-
7399 open.org/ns/cmis/link/200908/changes).

7400

7401 *The ChangeLog Token is specified in the URI specified by the paging link notations. Through this binding
7402 it is not possible to retrieve the ChangeLog Token from the URIs.*

7403

7404 CMIS Services:

7405 GET: getContentChanges()

7406 Media Type: application/atom+xml;type=feed

7407 Link Relations:

- 7408 • service: Points to service document containing the CMIS repository. The service document
7409 MUST contain only one workspace element.
 - 7410 ○ Media Type: application/atomsvc+xml
- 7411 • paging link relations as appropriate: first, next, previous, last
 - 7412 ○ ChangeLogToken is incorporated into the URI specified by the next link relation

7413

7414 This feed MUST be ordered from oldest first to newest.

7415

7416 If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is
7417 not available, the client should revisit the feed in the future and look for new items and the next link
7418 relation.

7419

7420 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7421 • cmisra:numItems

7422

7423 The following CMIS Atom extension element MUST be included inside the atom entries:

7424 • cmisra:object inside atom:entry

7425

7426 Example:

```

7427 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7428 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7429 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7430 xmlns:atom="http://www.w3.org/2005/Atom"
7431 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7432 open.org/ns/cmisis/restatom/200908/">
7433   <atom:title type="text">changelog feed</atom:title>
7434   <atom:author>
7435     <atom:name>Al Brown</atom:name>
7436     <atom:uri>http://www.ibm.com/</atom:uri>
7437     <atom:email>albertcbrown@us.ibm.com</atom:email>
7438   </atom:author>
7439   <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7440   <atom:id>urn:uuid:0bfc5306-fc76-4cd8-a0c0-7653dd43c0ff</atom:id>
7441   <atom:link type="application/atom+xml;type=feed" rel="self"
7442 href="http://cmisexample.oasis-open.org/rep1/oId/3"/>
7443   <atom:link type="application/atomsvc+xml" rel="service"
7444 href="http://cmisexample.oasis-open.org/rep1//service"/>
7445   <atom:link type="application/atom+xml;type=feed" rel="first"
7446 href="http://cmisexample.oasis-open.org/rep1/oId/first"/>
7447   <atom:link type="application/atom+xml;type=feed" rel="next"
7448 href="http://cmisexample.oasis-open.org/rep1/oId/4"/>
7449   <atom:link type="application/atom+xml;type=feed" rel="previous"
7450 href="http://cmisexample.oasis-open.org/rep1/oId/2"/>
7451   <atom:link type="application/atom+xml;type=feed" rel="last"
7452 href="http://cmisexample.oasis-open.org/rep1/oId/last"/>
7453   <cmisra:numItems>2</cmisra:numItems>
7454   <atom:entry>
7455     <atom:author>
7456       <atom:name>Al Brown</atom:name>
7457       <atom:uri>http://www.ibm.com/</atom:uri>
7458       <atom:email>albertcbrown@us.ibm.com</atom:email>
7459     </atom:author>
7460     <atom:content src="http://cmisexample.oasis-open.org/rep1/3f724c1d-
7461 12c8-43f2-919f-674df52b6ebd"/>
7462     <atom:id>urn:uuid:3f724c1d-12c8-43f2-919f-674df52b6ebd</atom:id>
7463     <atom:title type="text">CMIS Example Folder as Customer Policy
7464 type</atom:title>
7465     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7466     <atom:link rel="self" href="http://cmisexample.oasis-
7467 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
7468     <atom:link rel="edit" href="http://cmisexample.oasis-
7469 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
7470     <atom:link type="application/cmisis+xml;type=allowableActions"
7471 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7472 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7473 674df52b6ebd/allowableactions"/>
7474     <atom:link type="application/atom+xml;type=entry"
7475 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/3f724c1d-
7476 12c8-43f2-919f-674df52b6ebd/type"/>
7477     <atom:link type="application/atomsvc+xml" rel="service"
7478 href="http://cmisexample.oasis-open.org/rep1//service"/>
7479     <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
7480     <atom:summary type="html">HTML summary of Entry 3f724c1d-12c8-43f2-
7481 919f-674df52b6ebd</atom:summary>
7482     <atom:link type="application/atom+xml;type=entry" rel="up"
7483 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7484 674df52b6ebd/up"/>

```

```

7485     <atom:link type=""application/atom+xml;type=feed"" rel=""down""
7486 href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7487 674df52b6ebd/children""/>
7488     <atom:link type=""application/cmistree+xml"" rel=""down""
7489 href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7490 674df52b6ebd/tree""/>
7491     <atom:link type=""application/atom+xml;type=feed""
7492 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree""
7493 href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7494 674df52b6ebd/foldertree""/>
7495     <atom:link type=""application/atom+xml;type=feed""
7496 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships""
7497 href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7498 674df52b6ebd/relationships""/>
7499     <atom:link type=""application/atom+xml;type=feed""
7500 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies""
7501 href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7502 674df52b6ebd/policies""/>
7503     <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
7504 open.org/ns/cmisis/link/200908/acl"" href=""http://cmisexample.oasis-
7505 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd/acl""/>
7506     <cmisra:object>
7507         <cmis:properties>
7508             <cmis:propertyId localName=""rep-cmis:objectId""
7509 propertyDefinitionId=""cmis:objectId"">
7510                 <cmis:value>3f724c1d-12c8-43f2-919f-
7511 674df52b6ebd</cmis:value>
7512             </cmis:propertyId>
7513             <cmis:propertyId localName=""rep-cmis:objectTypeId""
7514 propertyDefinitionId=""cmis:objectTypeId"">
7515                 <cmis:value>customerpolicy</cmis:value>
7516             </cmis:propertyId>
7517             <cmis:propertyString localName=""rep-cmis:name""
7518 propertyDefinitionId=""cmis:name"">
7519                 <cmis:value>CMIS Example Folder as Customer Policy
7520 type</cmis:value>
7521             </cmis:propertyString>
7522             <cmis:propertyDateTime localName=""rep-cmis:creationDate""
7523 propertyDefinitionId=""cmis:creationDate"">
7524                 <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
7525             </cmis:propertyDateTime>
7526             <cmis:propertyDateTime localName=""rep-
7527 cmis:lastModificationDate""
7528 propertyDefinitionId=""cmis:lastModificationDate"">
7529                 <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
7530             </cmis:propertyDateTime>
7531             <cmis:propertyId localName=""rep-cmis:baseTypeId""
7532 propertyDefinitionId=""cmis:baseTypeId"">
7533                 <cmis:value>cmis:folder</cmis:value>
7534             </cmis:propertyId>
7535             <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
7536 propertyDefinitionId=""cmis:lastModifiedBy"">
7537                 <cmis:value>Al Brown</cmis:value>
7538             </cmis:propertyString>
7539             <cmis:propertyString localName=""rep-cmis:createdBy""
7540 propertyDefinitionId=""cmis:createdBy"">
7541                 <cmis:value>Al Brown</cmis:value>
7542             </cmis:propertyString>
7543             <cmis:propertyId localName=""rep-cmis:parentId""
7544 propertyDefinitionId=""cmis:parentId"">
7545                 <cmis:value>3f724c1d-12c8-43f2-919f-
7546 674df52b6ebdup</cmis:value>
7547             </cmis:propertyId>
7548         </cmis:properties>

```

```

7549         <cmis:changeEventInfo>
7550             <cmis:changeType>updated</cmis:changeType>
7551             <cmis:changeTime>2010-01-25T10:20:59.255-
7552 08:00</cmis:changeTime>
7553         </cmis:changeEventInfo>
7554     </cmisra:object>
7555     <cmisra:pathSegment>policy</cmisra:pathSegment>
7556 </atom:entry>
7557 <atom:entry>
7558     <atom:author>
7559         <atom:name>Al Brown</atom:name>
7560         <atom:uri>http://www.ibm.com/</atom:uri>
7561         <atom:email>albertcbrown@us.ibm.com</atom:email>
7562     </atom:author>
7563     <atom:content src=""http://cmisexample.oasis-open.org/rep1/6e27bada-
7564 b5a2-4a39-be2c-269806eb0d42"">
7565     <atom:id>urn:uuid:6e27bada-b5a2-4a39-be2c-269806eb0d42</atom:id>
7566     <atom:title type=""text"">CMIS Example Document</atom:title>
7567     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7568     <atom:link rel=""self"" href=""http://cmisexample.oasis-
7569 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"">
7570     <atom:link rel=""edit"" href=""http://cmisexample.oasis-
7571 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"">
7572     <atom:link type=""application/cmisacl+xml;type=allowableActions""
7573 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions""
7574 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7575 269806eb0d42/allowableactions"">
7576     <atom:link type=""application/atom+xml;type=entry""
7577 rel=""describedby"" href=""http://cmisexample.oasis-open.org/rep1/6e27bada-
7578 b5a2-4a39-be2c-269806eb0d42/type"">
7579     <atom:link type=""application/atomsvc+xml"" rel=""service""
7580 href=""http://cmisexample.oasis-open.org/rep1//service"">
7581     <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
7582     <atom:summary type=""html"">HTML summary of Entry 6e27bada-b5a2-4a39-
7583 be2c-269806eb0d42</atom:summary>
7584     <atom:link rel=""edit-media"" href=""http://cmisexample.oasis-
7585 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/edit-media"">
7586     <atom:link rel=""alternate"" href=""http://cmisexample.oasis-
7587 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/alternate"">
7588     <atom:link type=""application/atom+xml;type=feed"" rel=""up""
7589 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7590 269806eb0d42/parents"">
7591     <atom:link type=""application/atom+xml;type=feed"" rel=""version-
7592 history"" href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-
7593 be2c-269806eb0d42/allversions"">
7594     <atom:link type=""application/atom+xml;type=entry"" rel=""current-
7595 version"" href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-
7596 be2c-269806eb0d42/latest"">
7597     <atom:link type=""application/atom+xml;type=feed""
7598 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/relationships""
7599 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7600 269806eb0d42/relationships"">
7601     <atom:link type=""application/atom+xml;type=feed""
7602 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/policies""
7603 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7604 269806eb0d42/policies"">
7605     <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
7606 open.org/ns/cmisacl/link/200908/acl"" href=""http://cmisexample.oasis-
7607 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/acl"">
7608     <cmisra:object>
7609         <cmis:properties>
7610             <cmis:propertyId localName=""rep-cmis:objectId""
7611 propertyDefinitionId=""cmis:objectId"">

```

```

7612         <cmis:value>6e27bada-b5a2-4a39-be2c-
7613 269806eb0d42</cmis:value>
7614     </cmis:propertyId>
7615     <cmis:propertyId localName="rep-cmis:objectId"
7616 | propertyDefinitionId="cmis:objectId">
7617         <cmis:value>document</cmis:value>
7618     </cmis:propertyId>
7619     <cmis:propertyString localName="rep-cmis:name"
7620 | propertyDefinitionId="cmis:name">
7621         <cmis:value>CMIS Example Document</cmis:value>
7622     </cmis:propertyString>
7623     <cmis:propertyDateTime localName="rep-cmis:creationDate"
7624 | propertyDefinitionId="cmis:creationDate">
7625         <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
7626     </cmis:propertyDateTime>
7627     <cmis:propertyDateTime localName="rep-
7628 | cmis:lastModificationDate"
7629 | propertyDefinitionId="cmis:lastModificationDate">
7630         <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
7631     </cmis:propertyDateTime>
7632     <cmis:propertyId localName="rep-cmis:baseTypeId"
7633 | propertyDefinitionId="cmis:baseTypeId">
7634         <cmis:value>cmis:document</cmis:value>
7635     </cmis:propertyId>
7636     <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7637 | propertyDefinitionId="cmis:lastModifiedBy">
7638         <cmis:value>Al Brown</cmis:value>
7639     </cmis:propertyString>
7640     <cmis:propertyString localName="rep-cmis:createdBy"
7641 | propertyDefinitionId="cmis:createdBy">
7642         <cmis:value>Al Brown</cmis:value>
7643     </cmis:propertyString>
7644     <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
7645 | propertyDefinitionId="cmis:isLatestVersion">
7646         <cmis:value>>true</cmis:value>
7647     </cmis:propertyBoolean>
7648     <cmis:propertyBoolean localName="rep-
7649 | cmis:isVersionSeriesCheckedOut"
7650 | propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7651         <cmis:value>>false</cmis:value>
7652     </cmis:propertyBoolean>
7653     <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
7654 | propertyDefinitionId="cmis:isMajorVersion">
7655         <cmis:value>>false</cmis:value>
7656     </cmis:propertyBoolean>
7657     <cmis:propertyBoolean localName="rep-
7658 | cmis:isLatestMajorVersion"
7659 | propertyDefinitionId="cmis:isLatestMajorVersion">
7660         <cmis:value>>false</cmis:value>
7661     </cmis:propertyBoolean>
7662     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7663 | propertyDefinitionId="cmis:isImmutable">
7664         <cmis:value>>false</cmis:value>
7665     </cmis:propertyBoolean>
7666     <cmis:propertyString localName="rep-cmis:checkinComment"
7667 | propertyDefinitionId="cmis:checkinComment">
7668         <cmis:value>Checkin comment</cmis:value>
7669     </cmis:propertyString>
7670     <cmis:propertyString localName="rep-cmis:versionLabel"
7671 | propertyDefinitionId="cmis:versionLabel">
7672         <cmis:value>0.1</cmis:value>
7673     </cmis:propertyString>

```

```

7674 |         <cmis:propertyString localName=""rep-
7675 | cmis:contentStreamMimeType""
7676 | propertyDefinitionId=""cmis:contentStreamMimeType"">
7677 |         <cmis:value>text/plain</cmis:value>
7678 |     </cmis:propertyString>
7679 |     <cmis:propertyString localName=""rep-
7680 | cmis:contentStreamFileName""
7681 | propertyDefinitionId=""cmis:contentStreamFileName"">
7682 |         <cmis:value>text.txt</cmis:value>
7683 |     </cmis:propertyString>
7684 |     <cmis:propertyInteger localName=""rep-
7685 | cmis:contentStreamLength"" propertyDefinitionId=""cmis:contentStreamLength"">
7686 |         <cmis:value>4234</cmis:value>
7687 |     </cmis:propertyInteger>
7688 |     <cmis:propertyString displayName=""Keywords for Document""
7689 | localName=""keywords"" propertyDefinitionId=""keywords"">
7690 |         <cmis:value>document</cmis:value>
7691 |         <cmis:value>example</cmis:value>
7692 |         <cmis:value>sample</cmis:value>
7693 |         <cmis:value>cmis</cmis:value>
7694 |     </cmis:propertyString>
7695 | </cmis:properties>
7696 | <cmis:changeEventInfo>
7697 |     <cmis:changeType>updated</cmis:changeType>
7698 |     <cmis:changeTime>2010-01-25T10:20:59.271-
7699 | 08:00</cmis:changeTime>
7700 | </cmis:changeEventInfo>
7701 | </cmisra:object>
7702 |     <cmisra:pathSegment>invoice.pdf</cmisra:pathSegment>
7703 | </atom:entry>
7704 | </atom:feed>

```

7705

7706 Please also see the example documents included with the schema.

7707 3.9.2.1 GET

7708 The following optional parameters may be supplied:

- 7709 • filter
- 7710 • maxItems
- 7711 • includeACL
- 7712 • includePolicyIds
- 7713 • includeProperties
- 7714 • changeLogToken: If this parameter is specified, start the changes from the specified token. The
- 7715 changeLogToken is embedded in the paging link relations for normal iteration through the change
- 7716 list.

7717 3.9.3 Folder Descendants

7718 This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available
7719 via the link relation down with the application/cmistree+xml media type. Please see the Hierarchical Atom
7720 Entries for more information on format.

7721

7722 If a repository does not support capabilityGetDescendants, then these resources SHOULD NOT be
7723 exposed.

7724

7725 CMIS Services:

- 7726 GET: getDescendants
- 7727 DELETE: deleteTree
- 7728 Media Type: application/atom+xml;type=feed
- 7729
- 7730 Link Relations:
- 7731 • service: Points to service document containing the CMIS repository. The service document
 - 7732 MUST contain only one workspace element.
 - 7733 ○ Media Type: application/atomsvc+xml
 - 7734 • via: points to the atom entry of the folder generating this collection
 - 7735 • up: points to the atom entry document for this folder's parent
 - 7736 ○ Media Type: application/atom+xml;type=entry
 - 7737 ○ If the root folder, this link relation MUST not be included.
 - 7738 • down:
 - 7739 ○ points to the atom feed document representing the children feed for this same folder with
 - 7740 media type of application/atom+xml;type=entry
 - 7741 ○ Since this is the descendants, the descendants link SHOULD NOT be included
 - 7742 • paging link relations MAY be included as appropriate: first, next, previous, last
 - 7743 ○ Repositories may support these paging link relations on a particular cmisra:children
 - 7744 element.
 - 7745 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

7746

7747 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7748 • cmisra:numItems

7749

7750 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7751 • cmisra:object inside atom:entry
- 7752 • cmisra:pathSegment inside atom:entry
- 7753 • cmisra:children inside atom:entry

7754

7755 Example:

```

7756 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7757 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7758 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7759 xmlns:atom="http://www.w3.org/2005/Atom"
7760 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7761 open.org/ns/cmis/restatom/200908/">
7762   <atom:title type="text">Feed for folder1</atom:title>
7763   <atom:author>
7764     <atom:name>Al Brown</atom:name>
7765     <atom:uri>http://www.ibm.com/</atom:uri>
7766     <atom:email>albertcbrown@us.ibm.com</atom:email>
7767   </atom:author>
7768   <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
7769   <atom:id>urn:uuid:cb0a47d4-8d09-46f9-9b09-584acad684af</atom:id>
7770   <atom:link type="application/atom+xml;type=feed" rel="self"
7771 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7772 040ec0c7c05a/3"/>
7773   <atom:link type="application/atomsvc+xml" rel="service"
7774 href="http://cmisexample.oasis-open.org/rep1//service"/>

```

```

7775     <atom:link type=""application/atom+xml;type=entry" rel=""via"
7776 href=""http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7777 040ec0c7c05a"/>
7778     <atom:link type=""application/atom+xml;type=feed"
7779 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7780 href=""http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7781 040ec0c7c05a/foldertree"/>
7782     <atom:link type=""application/atom+xml;type=feed" rel=""down"
7783 href=""http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7784 040ec0c7c05a/children"/>
7785     <atom:link type=""application/atom+xml;type=entry" rel=""up"
7786 href=""http://cmisexample.oasis-open.org/rep1/03dcf5b8-5f82-45a1-b276-
7787 44d88069eec3"/>
7788     <cmisra:numItems>1</cmisra:numItems>
7789     <atom:entry>
7790         <atom:author>
7791             <atom:name>Al Brown</atom:name>
7792             <atom:uri>http://www.ibm.com/</atom:uri>
7793             <atom:email>albertcbrown@us.ibm.com</atom:email>
7794         </atom:author>
7795         <atom:content src=""http://cmisexample.oasis-open.org/rep1/8e5a512c-
7796 8f2d-4387-a283-f3f30bbc312f"/>
7797         <atom:id>urn:uuid:8e5a512c-8f2d-4387-a283-f3f30bbc312f</atom:id>
7798         <atom:title type=""text">CMIS Example Folder as Customer
7799 type</atom:title>
7800         <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
7801         <atom:link rel=""self" href=""http://cmisexample.oasis-
7802 open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
7803         <atom:link rel=""edit" href=""http://cmisexample.oasis-
7804 open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
7805         <atom:link type=""application/cmisis+xml;type=allowableActions"
7806 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7807 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7808 f3f30bbc312f/allowableactions"/>
7809         <atom:link type=""application/atom+xml;type=entry"
7810 rel=""describedby" href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-
7811 8f2d-4387-a283-f3f30bbc312f/type"/>
7812         <atom:link type=""application/atomsvc+xml" rel=""service"
7813 href=""http://cmisexample.oasis-open.org/rep1//service"/>
7814         <atom:published>2010-01-25T10:20:59.380-08:00</atom:published>
7815         <atom:summary type=""html">HTML summary of Entry 8e5a512c-8f2d-4387-
7816 a283-f3f30bbc312f</atom:summary>
7817         <atom:link type=""application/atom+xml;type=entry" rel=""up"
7818 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7819 f3f30bbc312f/up"/>
7820         <atom:link type=""application/atom+xml;type=feed" rel=""down"
7821 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7822 f3f30bbc312f/children"/>
7823         <atom:link type=""application/cmistree+xml" rel=""down"
7824 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7825 f3f30bbc312f/tree"/>
7826         <atom:link type=""application/atom+xml;type=feed"
7827 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7828 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7829 f3f30bbc312f/foldertree"/>
7830         <atom:link type=""application/atom+xml;type=feed"
7831 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
7832 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7833 f3f30bbc312f/relationships"/>
7834         <atom:link type=""application/atom+xml;type=feed"
7835 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
7836 href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7837 f3f30bbc312f/policies"/>

```

```

7838 |         <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
7839 | open.org/ns/cmisacl/link/200908/acl"" href=""http://cmisexample.oasis-
7840 | open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f/acl""/>
7841 |         <cmisra:object>
7842 |             <cmis:properties>
7843 |                 <cmis:propertyId localName=""rep-cmis:objectId""
7844 | propertyDefinitionId=""cmis:objectId"">
7845 |                     <cmis:value>8e5a512c-8f2d-4387-a283-
7846 | f3f30bbc312f</cmis:value>
7847 |                 </cmis:propertyId>
7848 |                 <cmis:propertyId localName=""rep-cmis:objectTypeId""
7849 | propertyDefinitionId=""cmis:objectTypeId"">
7850 |                     <cmis:value>customer</cmis:value>
7851 |                 </cmis:propertyId>
7852 |                 <cmis:propertyString localName=""rep-cmis:name""
7853 | propertyDefinitionId=""cmis:name"">
7854 |                     <cmis:value>CMIS Example Folder as Customer
7855 | type</cmis:value>
7856 |                 </cmis:propertyString>
7857 |                 <cmis:propertyDateTime localName=""rep-cmis:creationDate""
7858 | propertyDefinitionId=""cmis:creationDate"">
7859 |                     <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7860 |                 </cmis:propertyDateTime>
7861 |                 <cmis:propertyDateTime localName=""rep-
7862 | cmis:lastModificationDate""
7863 | propertyDefinitionId=""cmis:lastModificationDate"">
7864 |                     <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7865 |                 </cmis:propertyDateTime>
7866 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
7867 | propertyDefinitionId=""cmis:baseTypeId"">
7868 |                     <cmis:value>cmis:folder</cmis:value>
7869 |                 </cmis:propertyId>
7870 |                 <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
7871 | propertyDefinitionId=""cmis:lastModifiedBy"">
7872 |                     <cmis:value>Al Brown</cmis:value>
7873 |                 </cmis:propertyString>
7874 |                 <cmis:propertyString localName=""rep-cmis:createdBy""
7875 | propertyDefinitionId=""cmis:createdBy"">
7876 |                     <cmis:value>Al Brown</cmis:value>
7877 |                 </cmis:propertyString>
7878 |                 <cmis:propertyId localName=""rep-cmis:parentId""
7879 | propertyDefinitionId=""cmis:parentId"">
7880 |                     <cmis:value>8e5a512c-8f2d-4387-a283-
7881 | f3f30bbc312fup</cmis:value>
7882 |                 </cmis:propertyId>
7883 |             </cmis:properties>
7884 |         </cmisra:object>
7885 |         <cmisra:pathSegment>customer</cmisra:pathSegment>
7886 |         <cmisra:children>
7887 |             <atom:feed>
7888 |                 <atom:title type=""text"">CMIS Example Folder as Customer
7889 | type</atom:title>
7890 |                 <atom:author>
7891 |                     <atom:name>Al Brown</atom:name>
7892 |                     <atom:uri>http://www.ibm.com/</atom:uri>
7893 |                     <atom:email>albertcbrown@us.ibm.com</atom:email>
7894 |                 </atom:author>
7895 |                 <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
7896 |                 <atom:id>urn:uuid:67ee5e9f-d2e3-457d-9dec-
7897 | be718e780452</atom:id>
7898 |                 <atom:link type=""application/atom+xml;type=feed""
7899 | rel=""self"" href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-
7900 | a283-f3f30bbc312f/3""/>

```



```

7901         <atom:link type="application/atomsvc+xml" rel="service"
7902 href="http://cmisexample.oasis-open.org/rep1//service"/>
7903         <atom:link type="application/atom+xml;type=entry"
7904 rel="via" href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-
7905 a283-f3f30bbc312f"/>
7906         <atom:link type="application/atom+xml;type=feed"
7907 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7908 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7909 f3f30bbc312f/foldertree"/>
7910         <atom:link type="application/atom+xml;type=feed"
7911 rel="down" href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-
7912 a283-f3f30bbc312f/children"/>
7913         <atom:link type="application/atom+xml;type=entry" rel="up"
7914 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7915 040ec0c7c05a"/>
7916         <cmisra:numItems>1</cmisra:numItems>
7917         <atom:entry>
7918             <atom:author>
7919                 <atom:name>Al Brown</atom:name>
7920                 <atom:uri>http://www.ibm.com</atom:uri>
7921                 <atom:email>albertcbrown@us.ibm.com</atom:email>
7922             </atom:author>
7923             <atom:content src="http://cmisexample.oasis-
7924 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7925             <atom:id>urn:uuid:8c2dbba5-ea33-469f-a052-
7926 9f01e636c72a</atom:id>
7927             <atom:title type="text">CMIS Example Doc as Invoice
7928 type</atom:title>
7929             <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
7930             <atom:link rel="self" href="http://cmisexample.oasis-
7931 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7932             <atom:link rel="edit" href="http://cmisexample.oasis-
7933 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7934             <atom:link
7935 type="application/cmisis+xml;type=allowableActions" rel="http://docs.oasis-
7936 open.org/ns/cmisis/link/200908/allowableactions"
7937 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7938 9f01e636c72a/allowableactions"/>
7939             <atom:link type="application/atom+xml;type=entry"
7940 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-
7941 ea33-469f-a052-9f01e636c72a/type"/>
7942             <atom:link type="application/atomsvc+xml"
7943 rel="service" href="http://cmisexample.oasis-open.org/rep1//service"/>
7944             <atom:published>2010-01-25T10:20:59.380-
7945 08:00</atom:published>
7946             <atom:summary type="html">HTML summary of Entry
7947 8c2dbba5-ea33-469f-a052-9f01e636c72a</atom:summary>
7948             <atom:link rel="edit-media"
7949 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7950 9f01e636c72a/edit-media"/>
7951             <atom:link rel="alternate"
7952 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7953 9f01e636c72a/alternate"/>
7954             <atom:link type="application/atom+xml;type=feed"
7955 rel="up" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-
7956 a052-9f01e636c72a/parents"/>
7957             <atom:link type="application/atom+xml;type=feed"
7958 rel="version-history" href="http://cmisexample.oasis-
7959 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a/allversions"/>
7960             <atom:link type="application/atom+xml;type=entry"
7961 rel="current-version" href="http://cmisexample.oasis-
7962 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a/latest"/>

```

```

7963 |         <atom:link type=""application/atom+xml;type=feed""
7964 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships""
7965 | href=""http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7966 | 9f01e636c72a/relationships""/>
7967 |         <atom:link type=""application/atom+xml;type=feed""
7968 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies""
7969 | href=""http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7970 | 9f01e636c72a/policies""/>
7971 |         <atom:link type=""application/cmisacl+xml""
7972 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/acl""
7973 | href=""http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7974 | 9f01e636c72a/acl""/>
7975 |         <cmisra:object>
7976 |             <cmis:properties>
7977 |                 <cmis:propertyId localName=""rep-cmis:objectId""
7978 | propertyDefinitionId=""cmis:objectId"">
7979 | <cmis:value>8c2dbba5-ea33-469f-a052-9f01e636c72a</cmis:value>
7980 |                 </cmis:propertyId>
7981 |                 <cmis:propertyId localName=""rep-
7982 | cmis:objectTypeId"" propertyDefinitionId=""cmis:objectTypeId"">
7983 | <cmis:value>invoice</cmis:value>
7984 |                 </cmis:propertyId>
7985 |                 <cmis:propertyString localName=""rep-cmis:name""
7986 | propertyDefinitionId=""cmis:name"">
7987 | <cmis:value>CMIS Example Doc as Invoice type</cmis:value>
7988 |                 </cmis:propertyString>
7989 |                 <cmis:propertyDateTime localName=""rep-
7990 | cmis:creationDate"" propertyDefinitionId=""cmis:creationDate"">
7991 | <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7992 |                 </cmis:propertyDateTime>
7993 |                 <cmis:propertyDateTime localName=""rep-
7994 | cmis:lastModificationDate""
7995 | propertyDefinitionId=""cmis:lastModificationDate"">
7996 | <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7997 |                 </cmis:propertyDateTime>
7998 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
7999 | propertyDefinitionId=""cmis:baseTypeId"">
8000 | <cmis:value>cmis:document</cmis:value>
8001 |                 </cmis:propertyId>
8002 |                 <cmis:propertyString localName=""rep-
8003 | cmis:lastModifiedBy"" propertyDefinitionId=""cmis:lastModifiedBy"">
8004 | <cmis:value>Al Brown</cmis:value>
8005 |                 </cmis:propertyString>
8006 |                 <cmis:propertyString localName=""rep-
8007 | cmis:createdBy"" propertyDefinitionId=""cmis:createdBy"">
8008 | <cmis:value>Al Brown</cmis:value>
8009 |                 </cmis:propertyString>
8010 |                 <cmis:propertyBoolean localName=""rep-
8011 | cmis:isLatestVersion"" propertyDefinitionId=""cmis:isLatestVersion"">
8012 | <cmis:value>>true</cmis:value>
8013 |                 </cmis:propertyBoolean>
8014 |                 <cmis:propertyBoolean localName=""rep-
8015 | cmis:isVersionSeriesCheckedOut""
8016 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
8017 | <cmis:value>>false</cmis:value>
8018 |                 </cmis:propertyBoolean>
8019 |                 <cmis:propertyBoolean localName=""rep-
8020 | cmis:isMajorVersion"" propertyDefinitionId=""cmis:isMajorVersion"">
8021 | <cmis:value>>false</cmis:value>
8022 |                 </cmis:propertyBoolean>
8023 |                 <cmis:propertyBoolean localName=""rep-
8024 | cmis:isLatestMajorVersion""
8025 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
8026 | <cmis:value>>false</cmis:value>

```

```

8027         </cmis:propertyBoolean>
8028         <cmis:propertyBoolean localName="__rep-
8029 cmis:isImmutable__" propertyDefinitionId="__cmis:isImmutable__">
8030 <cmis:value>>false</cmis:value>
8031         </cmis:propertyBoolean>
8032         <cmis:propertyString localName="__rep-
8033 cmis:checkinComment__" propertyDefinitionId="__cmis:checkinComment__">
8034 <cmis:value>Checkin comment</cmis:value>
8035         </cmis:propertyString>
8036         <cmis:propertyString localName="__rep-
8037 cmis:versionLabel__" propertyDefinitionId="__cmis:versionLabel__">
8038 <cmis:value>0.1</cmis:value>
8039         </cmis:propertyString>
8040         <cmis:propertyString localName="__rep-
8041 cmis:contentStreamMimeType__"
8042 propertyDefinitionId="__cmis:contentStreamMimeType__">
8043 <cmis:value>text/plain</cmis:value>
8044         </cmis:propertyString>
8045         <cmis:propertyString localName="__rep-
8046 cmis:contentStreamFileName__"
8047 propertyDefinitionId="__cmis:contentStreamFileName__">
8048 <cmis:value>text.txt</cmis:value>
8049         </cmis:propertyString>
8050         <cmis:propertyInteger localName="__rep-
8051 cmis:contentStreamLength__" propertyDefinitionId="__cmis:contentStreamLength__">
8052 <cmis:value>4234</cmis:value>
8053         </cmis:propertyInteger>
8054         <cmis:propertyString displayName="__Keywords for
8055 Document__" localName="__keywords__" propertyDefinitionId="__keywords__">
8056 <cmis:value>document</cmis:value>
8057 <cmis:value>example</cmis:value>
8058 <cmis:value>sample</cmis:value>
8059 <cmis:value>cmis</cmis:value>
8060         </cmis:propertyString>
8061     </cmis:properties>
8062 </cmisra:object>
8063     <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
8064 </atom:entry>
8065 </atom:feed>
8066 </cmisra:children>
8067 </atom:entry>
8068 </atom:feed>

```

8069

8070 Please also see the example documents included with the schema.

8071 **3.9.3.1 GET**

8072 The following arguments may be supplied. Please see the domain model for more information:

- 8073 • filter
- 8074 • depth
- 8075 • includeAllowableActions
- 8076 • includeRelationships
- 8077 • renditionFilter
- 8078 • includePathSegment

8079 **3.9.3.2 DELETE**

8080 This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the
8081 domain model for more information:

- 8082 • continueOnFailure
- 8083 • unfileObjects

8084

8085 Status Code:

- 8086 • 200 OK if successful. Body contains entity describing the status
- 8087 • 202 Accepted, if accepted but deletion not yet taking place
- 8088 • 204 No Content, if successful with no content
- 8089 • 403 Forbidden, if permission is denied
- 8090 • 401 Unauthorized, if not authenticated
- 8091 • 500 Internal Server Error. The body SHOULD contain an entity describing the status

8092

8093 If the delete method does not delete all items, invoking GET with infinite depth on this URI will return the
8094 items not deleted. Subsequent DELETE methods can be invoked on this URI.

8095 Note: If the repository does not implement get on this resource, or the canGetDescendants is false, there
8096 is no mechanism to identify the resources that were not removed.

8097 3.9.4 Folder Tree

8098 This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link
8099 relation foldertree with media type application/atom+xml;type=feed. Please see the Hierarchical Atom
8100 Entries for more information on format.

8101

8102 CMIS Services:

8103 GET: getFolderTree

8104 DELETE: deleteTree

8105 Media Type: application/atom+xml;type=feed

8106

8107 Link Relations:

- 8108 • service: Points to service document containing the CMIS repository. The service document
8109 MUST contain only one workspace element.
 - 8110 ○ Media Type: application/atomsvc+xml
- 8111 • via: points to the atom entry of the folder generating this collection
- 8112 • up: points to the atom entry document of this folder's parent
 - 8113 ○ If the root folder, this link relation MUST not be included.
 - 8114 ○ Media Type: application/atom+xml;type=entry
- 8115 • down:
 - 8116 ○ application/atom+xml : Points to the atom feed document representing the children feed
8117 for this same folder
 - 8118 ○ application/cmistree+xml: Points to the descendants feed of the same folder. If a
8119 repository does not support capabilityGetDescendants, then this link SHOULD NOT be
8120 included.
- 8121 • paging link relations MAY be included as appropriate: first, next, previous, last
 - 8122 ○ Repositories may support these paging link relations on a particular cmisra:children
8123 element.

8124

8125 This feed contains a set of atom entries for each sub-folder in the folder.

8126

8127 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8128 • cmisra:numItems

8129

8130 The following CMIS Atom extension element MUST be included inside the atom entries:

- 8131 • cmisra:object inside atom:entry
- 8132 • cmisra:pathSegment inside atom:entry
- 8133 • cmisra:children inside atom:entry

8134

8135 Example:

```
8136 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8137 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
8138 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
8139 xmlns:atom="http://www.w3.org/2005/Atom"
8140 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8141 open.org/ns/cmisis/restatom/200908/">
8142   <atom:title type="text">FolderTree Feed of Folder1</atom:title>
8143   <atom:author>
8144     <atom:name>Al Brown</atom:name>
8145     <atom:uri>http://www.ibm.com/</atom:uri>
8146     <atom:email>albertcbrown@us.ibm.com</atom:email>
8147   </atom:author>
8148   <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
8149   <atom:id>urn:uuid:f87e5678-dd24-4214-9e71-635f060beb7d</atom:id>
8150   <atom:link type="application/atom+xml;type=feed" rel="self"
8151 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8152 b65edc3e1854/3"/>
8153   <atom:link type="application/atomsvc+xml" rel="service"
8154 href="http://cmisexample.oasis-open.org/rep1//service"/>
8155   <atom:link type="application/atom+xml;type=entry" rel="via"
8156 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8157 b65edc3e1854"/>
8158   <atom:link type="application/cmistree+xml" rel="down"
8159 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8160 b65edc3e1854/tree"/>
8161   <atom:link type="application/atom+xml;type=feed" rel="down"
8162 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8163 b65edc3e1854/children"/>
8164   <atom:link type="application/atom+xml;type=entry" rel="up"
8165 href="http://cmisexample.oasis-open.org/rep1/3056c4d7-4e16-49cb-a750-
8166 ad7a3844alaa"/>
8167   <cmisra:numItems>1</cmisra:numItems>
8168   <atom:entry>
8169     <atom:author>
8170       <atom:name>Al Brown</atom:name>
8171       <atom:uri>http://www.ibm.com/</atom:uri>
8172       <atom:email>albertcbrown@us.ibm.com</atom:email>
8173     </atom:author>
8174     <atom:content src="http://cmisexample.oasis-open.org/rep1/c7b5a83e-
8175 37b6-4f5a-b646-50892252a180"/>
8176     <atom:id>urn:uuid:c7b5a83e-37b6-4f5a-b646-50892252a180</atom:id>
8177     <atom:title type="text">Customer Folder</atom:title>
8178     <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
8179     <atom:link rel="self" href="http://cmisexample.oasis-
8180 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
8181     <atom:link rel="edit" href="http://cmisexample.oasis-
8182 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
```

```

8183 |         <atom:link type=""application/cmism+xml;type=allowableActions""
8184 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions""
8185 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8186 | 50892252a180/allowableactions""/>
8187 |         <atom:link type=""application/atom+xml;type=entry""
8188 | rel=""describedby"" href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-
8189 | 37b6-4f5a-b646-50892252a180/type""/>
8190 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
8191 | href=""http://cmisexample.oasis-open.org/rep1//service""/>
8192 |         <atom:published>2010-01-25T10:20:59.521-08:00</atom:published>
8193 |         <atom:summary type=""html"">HTML summary of Entry c7b5a83e-37b6-4f5a-
8194 | b646-50892252a180</atom:summary>
8195 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
8196 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8197 | 50892252a180/up""/>
8198 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
8199 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8200 | 50892252a180/children""/>
8201 |         <atom:link type=""application/cmistree+xml"" rel=""down""
8202 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8203 | 50892252a180/tree""/>
8204 |         <atom:link type=""application/atom+xml;type=feed""
8205 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/foldertree""
8206 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8207 | 50892252a180/foldertree""/>
8208 |         <atom:link type=""application/atom+xml;type=feed""
8209 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/relationships""
8210 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8211 | 50892252a180/relationships""/>
8212 |         <atom:link type=""application/atom+xml;type=feed""
8213 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/policies""
8214 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8215 | 50892252a180/policies""/>
8216 |         <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
8217 | open.org/ns/cmism/link/200908/acl"" href=""http://cmisexample.oasis-
8218 | open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180/acl""/>
8219 |         <cmisra:object>
8220 |             <cmis:properties>
8221 |                 <cmis:propertyId localName=""rep-cmis:objectId""
8222 | propertyDefinitionId=""cmis:objectId"">
8223 |                     <cmis:value>c7b5a83e-37b6-4f5a-b646-
8224 | 50892252a180</cmis:value>
8225 |                 </cmis:propertyId>
8226 |                 <cmis:propertyId localName=""rep-cmis:objectTypeId""
8227 | propertyDefinitionId=""cmis:objectTypeId"">
8228 |                     <cmis:value>customer</cmis:value>
8229 |                 </cmis:propertyId>
8230 |                 <cmis:propertyString localName=""rep-cmis:name""
8231 | propertyDefinitionId=""cmis:name"">
8232 |                     <cmis:value>Customer Folder</cmis:value>
8233 |                 </cmis:propertyString>
8234 |                 <cmis:propertyDateTime localName=""rep-cmis:creationDate""
8235 | propertyDefinitionId=""cmis:creationDate"">
8236 |                     <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
8237 |                 </cmis:propertyDateTime>
8238 |                 <cmis:propertyDateTime localName=""rep-
8239 | cmis:lastModificationDate""
8240 | propertyDefinitionId=""cmis:lastModificationDate"">
8241 |                     <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
8242 |                 </cmis:propertyDateTime>
8243 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
8244 | propertyDefinitionId=""cmis:baseTypeId"">
8245 |                     <cmis:value>cmis:folder</cmis:value>
8246 |                 </cmis:propertyId>

```

```

8247 |         <cmis:propertyString localName="__rep-cmis:lastModifiedBy__"
8248 | propertyDefinitionId="__cmis:lastModifiedBy__">
8249 |             <cmis:value>Al Brown</cmis:value>
8250 |         </cmis:propertyString>
8251 |         <cmis:propertyString localName="__rep-cmis:createdBy__"
8252 | propertyDefinitionId="__cmis:createdBy__">
8253 |             <cmis:value>Al Brown</cmis:value>
8254 |         </cmis:propertyString>
8255 |         <cmis:propertyId localName="__rep-cmis:parentId__"
8256 | propertyDefinitionId="__cmis:parentId__">
8257 |             <cmis:value>c7b5a83e-37b6-4f5a-b646-
8258 | 50892252a180up</cmis:value>
8259 |         </cmis:propertyId>
8260 |     </cmis:properties>
8261 | </cmisra:object>
8262 | <cmisra:pathSegment>customer</cmisra:pathSegment>
8263 | </atom:entry>
8264 | </atom:feed>

```

8265

8266 Please also see the example documents included with the schema.

8267 **3.9.4.1 GET**

8268 The following arguments may be supplied. Please see the domain model for more information:

- 8269 • filter
- 8270 • depth
- 8271 • includeAllowableActions
- 8272 • includeRelationships
- 8273 • renditionFilter

8274 **3.9.4.2 DELETE**

8275 This is the same as DELETE on Folder Descendants. Please see that section.

8276 **3.9.5 AllVersions Feed**

8277 This is a feed comprised of all the versions of the given document.

8278 CMIS Services:

8279 GET: getAllVersions

8280 DELETE: deleteAllVersions

8281 Media Type: application/atom+xml;type=feed

8282

8283 The feed SHOULD contain the newest versions at the beginning of the feed.

8284

8285 Link Relations:

- 8286 • service: Points to service document containing the CMIS repository. The service document
 - 8287 MUST contain only one workspace element.
 - 8288 ○ Media Type: application/atomsvc+xml
- 8289 • via: points to the atom entry of the resource generating this collection
- 8290 • paging link relations as appropriate: first, next, previous, last

8291

8292 This feed contains a set of atom entries for each version in the version series

- 8293 • cmisra:object inside atom:entry
- 8294 • cmisra:children inside atom:entry if atom:entry represents a CMIS Folder

8295

8296 Example:

```

8297 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8298 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
8299 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
8300 xmlns:atom="http://www.w3.org/2005/Atom"
8301 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8302 open.org/ns/cmisis/restatom/200908/"
8303 <atom:title type="text">AllVersions for Document e8abd7cd-b9ec-4dba-
8304 9eaa-1bce2ae9977f</atom:title>
8305 <atom:author>
8306 <atom:name>Al Brown</atom:name>
8307 <atom:uri>http://www.ibm.com/</atom:uri>
8308 <atom:email>albertcbrown@us.ibm.com</atom:email>
8309 </atom:author>
8310 <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
8311 <atom:id>urn:uuid:5dc3d1c1-3e85-4720-acf8-cf98c96a5830</atom:id>
8312 <atom:link type="application/atom+xml;type=feed" rel="self"
8313 href="http://cmisexample.oasis-open.org/rep1/e8abd7cd-b9ec-4dba-9eaa-
8314 1bce2ae9977f/3"/>
8315 <atom:link type="application/atomsvc+xml" rel="service"
8316 href="http://cmisexample.oasis-open.org/rep1//service"/>
8317 <atom:link type="application/atom+xml;type=entry" rel="via"
8318 href="http://cmisexample.oasis-open.org/rep1/e8abd7cd-b9ec-4dba-9eaa-
8319 1bce2ae9977f"/>
8320 <cmisra:numItems>1</cmisra:numItems>
8321 <atom:entry>
8322 <atom:author>
8323 <atom:name>Al Brown</atom:name>
8324 <atom:uri>http://www.ibm.com/</atom:uri>
8325 <atom:email>albertcbrown@us.ibm.com</atom:email>
8326 </atom:author>
8327 <atom:content src="http://cmisexample.oasis-open.org/rep1/197033f2-
8328 ac11-4911-b5a3-60781fa5c281"/>
8329 <atom:id>urn:uuid:197033f2-ac11-4911-b5a3-60781fa5c281</atom:id>
8330 <atom:title type="text">Invoice (Version1)</atom:title>
8331 <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
8332 <atom:link rel="self" href="http://cmisexample.oasis-
8333 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281"/>
8334 <atom:link rel="edit" href="http://cmisexample.oasis-
8335 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281"/>
8336 <atom:link type="application/cmisis+xml;type=allowableActions"
8337 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
8338 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8339 60781fa5c281/allowableactions"/>
8340 <atom:link type="application/atom+xml;type=entry"
8341 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/197033f2-
8342 ac11-4911-b5a3-60781fa5c281/type"/>
8343 <atom:link type="application/atomsvc+xml" rel="service"
8344 href="http://cmisexample.oasis-open.org/rep1//service"/>
8345 <atom:published>2010-01-25T10:20:58.896-08:00</atom:published>
8346 <atom:summary type="html">HTML summary of Entry 197033f2-ac11-4911-
8347 b5a3-60781fa5c281</atom:summary>
8348 <atom:link rel="edit-media" href="http://cmisexample.oasis-
8349 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/edit-media"/>
8350 <atom:link rel="alternate" href="http://cmisexample.oasis-
8351 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/alternate"/>
8352 <atom:link type="application/atom+xml;type=feed" rel="up"
8353 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8354 60781fa5c281/parents"/>

```



```

8355     <atom:link type=""application/atom+xml;type=feed"" rel=""version-
8356 history"" href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-
8357 b5a3-60781fa5c281/allversions"" />
8358     <atom:link type=""application/atom+xml;type=entry"" rel=""current-
8359 version"" href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-
8360 b5a3-60781fa5c281/latest"" />
8361     <atom:link type=""application/atom+xml;type=feed""
8362 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships""
8363 href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8364 60781fa5c281/relationships"" />
8365     <atom:link type=""application/atom+xml;type=feed""
8366 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies""
8367 href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8368 60781fa5c281/policies"" />
8369     <atom:link type=""application/cmisisacl+xml"" rel=""http://docs.oasis-
8370 open.org/ns/cmisis/link/200908/acl"" href=""http://cmisexample.oasis-
8371 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/acl"" />
8372     <cmisisra:object>
8373       <cmisisra:properties>
8374         <cmisisra:propertyId localName=""rep-cmisis:objectId""
8375 propertyDefinitionId=""cmisis:objectId"">
8376           <cmisisra:value>197033f2-ac11-4911-b5a3-
8377 60781fa5c281</cmisisra:value>
8378         </cmisisra:propertyId>
8379       </cmisisra:properties>
8380     </cmisisra:object>
8381   </atom:entry>
8382 </atom:feed>

```

8383

8384 Please also see the example documents included with the schema.

8385 **3.9.5.1 GET**

8386 The following arguments may be supplied. Please see the domain model for more information:

- 8387 • filter
- 8388 • includeAllowableActions

8389 **3.9.5.2 DELETE**

8390 This removes the entire version history of the document.

8391

8392 Success HTTP code: 204

8393 **3.9.6 Type Descendants Feed**

8394 This is a feed described in the service document that contains all the types under a specific type in the
8395 repository to a specific depth. If no parent type is specified, then the base types and their descendants
8396 are returned in the feed which is equivalent to all types in the repository if depth is infinite. The link
8397 relation is <http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>.

8398

8399 Types are nested using the CMIS hierarchy extension. Please see section 3.4.3.2 [Hierarchy Navigation](#)
8400 [Internet Draft Link Relations](#)~~Hierarchy Navigation Internet Draft Link Relations~~.

8401

8402 CMIS Services:

8403 GET: getTypeDescendants

8404

8405 Media Type: application/atom+xml;type=feed

8406

8407 Link Relations:

- 8408 • service: Points to service document containing the CMIS repository. The service document
- 8409 MUST contain only one workspace element.
 - 8410 ○ Media Type: application/atomsvc+xml
- 8411 • via: points to the type definition whose descendents represent this feed. This link is not present if
- 8412 no parent type is specified.
- 8413 • down: points to the children feed for the same type
- 8414 • up: points to the parent type definition
 - 8415 ○ If this is a descendants feed for a base object type, this link is not present.

8416

8417 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8418 • cmisra:numItems

8419

8420 Example:

```
8421 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8422 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
8423 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
8424 xmlns:atom="http://www.w3.org/2005/Atom"
8425 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8426 open.org/ns/cmisis/restatom/200908/"
8427 <atom:title type="text">Base Types</atom:title>
8428 <atom:author>
8429 <atom:name>Al Brown</atom:name>
8430 <atom:uri>http://www.ibm.com/</atom:uri>
8431 <atom:email>albertcbrown@us.ibm.com</atom:email>
8432 </atom:author>
8433 <atom:updated>2010-01-25T10:20:59.911-08:00</atom:updated>
8434 <atom:id>urn:uuid:c5d3d357-33ec-47c1-8436-563e0d94d2e5</atom:id>
8435 <atom:link type="application/atom+xml;type=feed" rel="self"
8436 href="http://cmisexample.oasis-open.org/rep1//3"/>
8437 <atom:link type="application/atomsvc+xml" rel="service"
8438 href="http://cmisexample.oasis-open.org/rep1//service"/>
8439 <atom:link type="application/atom+xml;type=entry" rel="via"
8440 href="http://cmisexample.oasis-open.org/rep1/">
8441 <atom:link type="application/atom+xml;type=feed" rel="down"
8442 href="http://cmisexample.oasis-open.org/rep1//children"/>
8443 <cmisra:numItems>1</cmisra:numItems>
8444 <atom:entry>
8445 <atom:author>
8446 <atom:name>Al Brown</atom:name>
8447 <atom:uri>http://www.ibm.com/</atom:uri>
8448 <atom:email>albertcbrown@us.ibm.com</atom:email>
8449 </atom:author>
8450 <atom:content>Type Definition for cmis:document</atom:content>
8451 <atom:id>http://cmisexample.oasis-
8452 open.org/rep1/type/cmisis:document</atom:id>
8453 <atom:link type="application/atom+xml;type=entry" rel="self"
8454 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document"/>
8455 <atom:link type="application/atomsvc+xml" rel="service"
8456 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document"/>
8457 <atom:link type="application/atom+xml;type=entry"
8458 rel="describedby" href="http://cmisexample.oasis-
8459 open.org/rep1/type/cmisis:document"/>
8460 <atom:link type="application/atom+xml;type=entry" rel="up"
8461 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document/parent"/>
```

```

8462 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
8463 | href=""http://cmisexample.oasis-
8464 | open.org/rep1/type/cmism:document/children/flat"">
8465 |         <atom:link type=""application/cmistree+xml"" rel=""down""
8466 | href=""http://cmisexample.oasis-
8467 | open.org/rep1/type/cmism:document/children/tree"">
8468 |         <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
8469 |         <atom:summary type=""html"">HTML summary of Type Definition
8470 | cmism:document</atom:summary>
8471 |         <atom:title type=""text"">Type Definition - cmism:document</atom:title>
8472 | <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8473 | <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8474 | <cmisra:type xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
8475 | xsi:type=""cmism:cmisTypeDocumentDefinitionType"" cmisra:id=""cmism:document"">
8476 |     <cmis:id>dtcmism:document</cmis:id>
8477 |     <cmis:localName>myrepname-cmism:document</cmis:localName>
8478 |     <cmis:localNamespace xsi:nil=""true"">
8479 |     <cmis:displayName>cmism:document</cmis:displayName>
8480 |     <cmis:queryName>cmism:document</cmis:queryName>
8481 |     <cmis:description>Description for type definition
8482 | cmism:document</cmis:description>
8483 |     <cmis:baseId>cmism:document</cmis:baseId>
8484 |     <cmis:parentId>parent</cmis:parentId>
8485 |     <cmis:creatable>true</cmis:creatable>
8486 |     <cmis:fileable>true</cmis:fileable>
8487 |     <cmis:queryable>false</cmis:queryable>
8488 |     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8489 |
8490 | <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8491 |     <cmis:controllablePolicy>true</cmis:controllablePolicy>
8492 |     <cmis:controllableACL>true</cmis:controllableACL>
8493 |     <cmis:versionable>true</cmis:versionable>
8494 |     <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8495 | </cmisra:type>
8496 | <cmisra:children>
8497 |     <atom:feed>
8498 |         <atom:title type=""text"">Children for Document</atom:title>
8499 |         <atom:author>
8500 |             <atom:name>Al Brown</atom:name>
8501 |             <atom:uri>http://www.ibm.com/</atom:uri>
8502 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
8503 |         </atom:author>
8504 |         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8505 |         <atom:id>urn:uuid:6f1cdc44-bd89-41c0-8fad-
8506 | 89f3ad0f8f30</atom:id>
8507 |         <atom:link type=""application/atom+xml;type=feed""
8508 | rel=""self"" href=""http://cmisexample.oasis-open.org/rep1/cmism:document/3"">
8509 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
8510 | href=""http://cmisexample.oasis-open.org/rep1//service"">
8511 |         <atom:link type=""application/atom+xml;type=entry""
8512 | rel=""via"" href=""http://cmisexample.oasis-open.org/rep1/cmism:document"">
8513 |         <atom:link type=""application/atom+xml;type=feed""
8514 | rel=""down"" href=""http://cmisexample.oasis-
8515 | open.org/rep1/cmism:document/children"">
8516 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
8517 | href=""http://cmisexample.oasis-open.org/rep1/document"">
8518 |         <cmisra:numItems>1</cmisra:numItems>
8519 |         <atom:entry>
8520 |             <atom:author>
8521 |                 <atom:name>Al Brown</atom:name>
8522 |                 <atom:uri>http://www.ibm.com/</atom:uri>
8523 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8524 |             </atom:author>

```

```

8525         <atom:content>Type Definition for invoice-
8526 document</atom:content>
8527         <atom:id>http://cmisexample.oasis-
8528 open.org/repl/type/invoice-document</atom:id>
8529         <atom:link type=""application/atom+xml;type=entry""
8530 rel=""self"" href=""http://cmisexample.oasis-open.org/repl/type/invoice-
8531 document""/>
8532         <atom:link type=""application/atomsvc+xml""
8533 rel=""service"" href=""http://cmisexample.oasis-open.org/repl/type/invoice-
8534 document""/>
8535         <atom:link type=""application/atom+xml;type=entry""
8536 rel=""describedby"" href=""http://cmisexample.oasis-
8537 open.org/repl/type/cmis:document""/>
8538         <atom:link type=""application/atom+xml;type=entry""
8539 rel=""up"" href=""http://cmisexample.oasis-open.org/repl/type/invoice-
8540 document/parent""/>
8541         <atom:link type=""application/atom+xml;type=feed""
8542 rel=""down"" href=""http://cmisexample.oasis-open.org/repl/type/invoice-
8543 document/children/flat""/>
8544         <atom:link type=""application/cmistree+xml"" rel=""down""
8545 href=""http://cmisexample.oasis-open.org/repl/type/invoice-
8546 document/children/tree""/>
8547         <atom:published>2010-01-25T10:20:59.927-
8548 08:00</atom:published>
8549         <atom:summary type=""html"">HTML summary of Type
8550 Definition invoice-document</atom:summary>
8551         <atom:title type=""text"">Type Definition - invoice-
8552 document</atom:title>
8553         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8554         <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8555         <cmisra:type xmlns:xsi=""http://www.w3.org/2001/XMLSchema-
8556 instance"" xsi:type=""cmis:cmisTypeDocumentDefinitionType""
8557 cmisra:id=""invoice-document"">
8558         <cmis:id>dtinvoice-document</cmis:id>
8559         <cmis:localName>myrepname-invoice-
8560 document</cmis:localName>
8561         <cmis:localNamespace xsi:nil=""true""/>
8562         <cmis:displayName>invoice-document</cmis:displayName>
8563         <cmis:queryName>invoice-document</cmis:queryName>
8564         <cmis:description>Description for type definition
8565 invoice-document</cmis:description>
8566         <cmis:baseId>cmis:document</cmis:baseId>
8567         <cmis:parentId>parent</cmis:parentId>
8568         <cmis:creatable>true</cmis:creatable>
8569         <cmis:fileable>true</cmis:fileable>
8570         <cmis:queryable>false</cmis:queryable>
8571         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8572
8573 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8574
8575 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8576         <cmis:controllableACL>true</cmis:controllableACL>
8577         <cmis:versionable>true</cmis:versionable>
8578
8579 <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8580 </cmisra:type>
8581 </atom:entry>
8582 </atom:feed>
8583 </cmisra:children>
8584 </atom:entry>
8585 <atom:entry>
8586 <atom:author>
8587 <atom:name>Al Brown</atom:name>
8588 <atom:uri>http://www.ibm.com/</atom:uri>

```

```

8589         <atom:email>albertcbrown@us.ibm.com</atom:email>
8590     </atom:author>
8591     <atom:content>Type Definition for cmis:folder</atom:content>
8592     <atom:id>http://cmisexample.oasis-
8593 open.org/repl/type/cmis:folder</atom:id>
8594     <atom:link type="application/atom+xml;type=entry" rel="self"
8595 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8596     <atom:link type="application/atomsvc+xml" rel="service"
8597 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8598     <atom:link type="application/atom+xml;type=entry"
8599 rel="describedby" href="http://cmisexample.oasis-
8600 open.org/repl/type/cmis:folder"/>
8601     <atom:link type="application/atom+xml;type=entry" rel="up"
8602 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/parent"/>
8603     <atom:link type="application/atom+xml;type=feed" rel="down"
8604 href="http://cmisexample.oasis-
8605 open.org/repl/type/cmis:folder/children/flat"/>
8606     <atom:link type="application/cmistree+xml" rel="down"
8607 href="http://cmisexample.oasis-
8608 open.org/repl/type/cmis:folder/children/tree"/>
8609     <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
8610     <atom:summary type="html">HTML summary of Type Definition
8611 cmis:folder</atom:summary>
8612     <atom:title type="text">Type Definition - cmis:folder</atom:title>
8613     <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8614     <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8615     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8616 xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="cmis:folder">
8617         <cmis:id>dtcmis:folder</cmis:id>
8618         <cmis:localName>myrepname-cmis:folder</cmis:localName>
8619         <cmis:localNamespace xsi:nil="true"/>
8620         <cmis:displayName>cmis:folder</cmis:displayName>
8621         <cmis:queryName>cmis:folder</cmis:queryName>
8622         <cmis:description>Description for type definition
8623 cmis:folder</cmis:description>
8624         <cmis:baseId>cmis:folder</cmis:baseId>
8625         <cmis:parentId>parent</cmis:parentId>
8626         <cmis:creatable>true</cmis:creatable>
8627         <cmis:fileable>true</cmis:fileable>
8628         <cmis:queryable>false</cmis:queryable>
8629         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8630
8631 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8632 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8633 <cmis:controllableACL>true</cmis:controllableACL>
8634 </cmisra:type>
8635 <cmisra:children>
8636     <atom:feed>
8637         <atom:title type="text">Children for Folder</atom:title>
8638         <atom:author>
8639             <atom:name>Al Brown</atom:name>
8640             <atom:uri>http://www.ibm.com/</atom:uri>
8641             <atom:email>albertcbrown@us.ibm.com</atom:email>
8642         </atom:author>
8643         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8644         <atom:id>urn:uuid:361a3ac1-f7f7-47cb-b941-
8645 ae1200213fe0</atom:id>
8646         <atom:link type="application/atom+xml;type=feed"
8647 rel="self" href="http://cmisexample.oasis-open.org/repl/cmis:folder/3"/>
8648         <atom:link type="application/atomsvc+xml" rel="service"
8649 href="http://cmisexample.oasis-open.org/repl//service"/>
8650         <atom:link type="application/atom+xml;type=entry"
8651 rel="via" href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>

```

```

8652         <atom:link type="application/atom+xml;type=feed"
8653 rel="down" href="http://cmisexample.oasis-
8654 open.org/repl/cmisis:folder/children"/>
8655         <atom:link type="application/atom+xml;type=entry" rel="up"
8656 href="http://cmisexample.oasis-open.org/repl/cmisis:folder"/>
8657         <cmisra:numItems>1</cmisra:numItems>
8658         <atom:entry>
8659           <atom:author>
8660             <atom:name>Al Brown</atom:name>
8661             <atom:uri>http://www.ibm.com/</atom:uri>
8662             <atom:email>albertcbrown@us.ibm.com</atom:email>
8663           </atom:author>
8664           <atom:content>Type Definition for customer-
8665 folder</atom:content>
8666           <atom:id>http://cmisexample.oasis-
8667 open.org/repl/type/customer-folder</atom:id>
8668           <atom:link type="application/atom+xml;type=entry"
8669 rel="self" href="http://cmisexample.oasis-open.org/repl/type/customer-
8670 folder"/>
8671           <atom:link type="application/atomsvc+xml"
8672 rel="service" href="http://cmisexample.oasis-open.org/repl/type/customer-
8673 folder"/>
8674           <atom:link type="application/atom+xml;type=entry"
8675 rel="describedby" href="http://cmisexample.oasis-
8676 open.org/repl/type/cmisis:folder"/>
8677           <atom:link type="application/atom+xml;type=entry"
8678 rel="up" href="http://cmisexample.oasis-open.org/repl/type/customer-
8679 folder/parent"/>
8680           <atom:link type="application/atom+xml;type=feed"
8681 rel="down" href="http://cmisexample.oasis-open.org/repl/type/customer-
8682 folder/children/flat"/>
8683           <atom:link type="application/cmistree+xml" rel="down"
8684 href="http://cmisexample.oasis-open.org/repl/type/customer-
8685 folder/children/tree"/>
8686           <atom:published>2010-01-25T10:20:59.927-
8687 08:00</atom:published>
8688           <atom:summary type="html">HTML summary of Type
8689 Definition customer-folder</atom:summary>
8690           <atom:title type="text">Type Definition - customer-
8691 folder</atom:title>
8692           <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8693           <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8694           <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8695 instance" xsi:type="cmis:cmisTypeFolderDefinitionType"
8696 cmisra:id="customer-folder">
8697             <cmis:id>dtcustomer-folder</cmis:id>
8698             <cmis:localName>myrepname-customer-
8699 folder</cmis:localName>
8700             <cmis:localNamespace xsi:nil="true"/>
8701             <cmis:displayName>customer-folder</cmis:displayName>
8702             <cmis:queryName>customer-folder</cmis:queryName>
8703             <cmis:description>Description for type definition
8704 customer-folder</cmis:description>
8705             <cmis:baseId>cmis:folder</cmis:baseId>
8706             <cmis:parentId>parent</cmis:parentId>
8707             <cmis:creatable>true</cmis:creatable>
8708             <cmis:fileable>true</cmis:fileable>
8709             <cmis:queryable>false</cmis:queryable>
8710             <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8711
8712 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8713
8714 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8715 <cmis:controllableACL>true</cmis:controllableACL>

```

```

8716         </cmisra:type>
8717     </atom:entry>
8718 </atom:feed>
8719 </cmisra:children>
8720 </atom:entry>
8721 <atom:entry>
8722     <atom:author>
8723         <atom:name>Al Brown</atom:name>
8724         <atom:uri>http://www.ibm.com/</atom:uri>
8725         <atom:email>albertcbrown@us.ibm.com</atom:email>
8726     </atom:author>
8727     <atom:content>Type Definition for cmis:relationship</atom:content>
8728     <atom:id>http://cmisexample.oasis-
8729 open.org/repl/type/cmis:relationship</atom:id>
8730     <atom:link type="application/atom+xml;type=entry" rel="self"
8731 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8732     <atom:link type="application/atomsvc+xml" rel="service"
8733 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8734     <atom:link type="application/atom+xml;type=entry"
8735 rel="describedby" href="http://cmisexample.oasis-
8736 open.org/repl/type/cmis:relationship"/>
8737     <atom:link type="application/atom+xml;type=entry" rel="up"
8738 href="http://cmisexample.oasis-
8739 open.org/repl/type/cmis:relationship/parent"/>
8740     <atom:link type="application/atom+xml;type=feed" rel="down"
8741 href="http://cmisexample.oasis-
8742 open.org/repl/type/cmis:relationship/children/flat"/>
8743     <atom:link type="application/cmistree+xml" rel="down"
8744 href="http://cmisexample.oasis-
8745 open.org/repl/type/cmis:relationship/children/tree"/>
8746     <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
8747     <atom:summary type="html">HTML summary of Type Definition
8748 cmis:relationship</atom:summary>
8749     <atom:title type="text">Type Definition -
8750 cmis:relationship</atom:title>
8751     <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8752     <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8753     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8754 xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8755 cmisra:id="cmis:relationship">
8756         <cmis:id>dtcmis:relationship</cmis:id>
8757         <cmis:localName>myrepname-cmis:relationship</cmis:localName>
8758         <cmis:localNamespace xsi:nil="true"/>
8759         <cmis:displayName>cmis:relationship</cmis:displayName>
8760         <cmis:queryName>cmis:relationship</cmis:queryName>
8761         <cmis:description>Description for type definition
8762 cmis:relationship</cmis:description>
8763         <cmis:baseId>cmis:relationship</cmis:baseId>
8764         <cmis:parentId>parent</cmis:parentId>
8765         <cmis:creatable>true</cmis:creatable>
8766         <cmis:fileable>false</cmis:fileable>
8767         <cmis:queryable>false</cmis:queryable>
8768         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8769
8770 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8771     <cmis:controllablePolicy>true</cmis:controllablePolicy>
8772     <cmis:controllableACL>true</cmis:controllableACL>
8773     <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8774     <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8775     <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8776 </cmisra:type>
8777 </cmisra:children>
8778 </atom:feed>

```

```

8779 |         <atom:title type="text">Children for
8780 | Relationship</atom:title>
8781 |         <atom:author>
8782 |             <atom:name>Al Brown</atom:name>
8783 |             <atom:uri>http://www.ibm.com/</atom:uri>
8784 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
8785 |         </atom:author>
8786 |         <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8787 |         <atom:id>urn:uuid:9394ff3d-87c4-48c9-a951-
8788 | ba725560faac</atom:id>
8789 |         <atom:link type="application/atom+xml;type=feed"
8790 | rel="self" href="http://cmisexample.oasis-
8791 | open.org/repl/cmisis:relationship/3"/>
8792 |         <atom:link type="application/atomsvc+xml" rel="service"
8793 | href="http://cmisexample.oasis-open.org/repl//service"/>
8794 |         <atom:link type="application/atom+xml;type=entry"
8795 | rel="via" href="http://cmisexample.oasis-
8796 | open.org/repl/cmisis:relationship"/>
8797 |         <atom:link type="application/atom+xml;type=feed"
8798 | rel="down" href="http://cmisexample.oasis-
8799 | open.org/repl/cmisis:relationship/children"/>
8800 |         <atom:link type="application/atom+xml;type=entry" rel="up"
8801 | href="http://cmisexample.oasis-open.org/repl/cmisis:folder"/>
8802 |         <cmisra:numItems>1</cmisra:numItems>
8803 |         <atom:entry>
8804 |             <atom:author>
8805 |                 <atom:name>Al Brown</atom:name>
8806 |                 <atom:uri>http://www.ibm.com/</atom:uri>
8807 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8808 |             </atom:author>
8809 |             <atom:content>Type Definition for customer-
8810 | relationship</atom:content>
8811 |             <atom:id>http://cmisexample.oasis-
8812 | open.org/repl/type/customer-relationship</atom:id>
8813 |             <atom:link type="application/atom+xml;type=entry"
8814 | rel="self" href="http://cmisexample.oasis-open.org/repl/type/customer-
8815 | relationship"/>
8816 |             <atom:link type="application/atomsvc+xml"
8817 | rel="service" href="http://cmisexample.oasis-open.org/repl/type/customer-
8818 | relationship"/>
8819 |             <atom:link type="application/atom+xml;type=entry"
8820 | rel="describedby" href="http://cmisexample.oasis-
8821 | open.org/repl/type/cmisis:relationship"/>
8822 |             <atom:link type="application/atom+xml;type=entry"
8823 | rel="up" href="http://cmisexample.oasis-open.org/repl/type/customer-
8824 | relationship/parent"/>
8825 |             <atom:link type="application/atom+xml;type=feed"
8826 | rel="down" href="http://cmisexample.oasis-open.org/repl/type/customer-
8827 | relationship/children/flat"/>
8828 |             <atom:link type="application/cmistree+xml" rel="down"
8829 | href="http://cmisexample.oasis-open.org/repl/type/customer-
8830 | relationship/children/tree"/>
8831 |             <atom:published>2010-01-25T10:20:59.943-
8832 | 08:00</atom:published>
8833 |             <atom:summary type="html">HTML summary of Type
8834 | Definition customer-relationship</atom:summary>
8835 |             <atom:title type="text">Type Definition - customer-
8836 | relationship</atom:title>
8837 |             <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8838 |             <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8839 |             <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8840 | instance" xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8841 | cmisra:id="customer-relationship">
8842 |                 <cmis:id>dtcustomer-relationship</cmis:id>

```



```

8843         <cmis:localName>myrepname-customer-
8844 relationship</cmis:localName>
8845         <cmis:localNamespace xsi:nil="true"/>
8846         <cmis:displayName>customer-
8847 relationship</cmis:displayName>
8848         <cmis:queryName>customer-relationship</cmis:queryName>
8849         <cmis:description>Description for type definition
8850 customer-relationship</cmis:description>
8851         <cmis:baseId>cmis:relationship</cmis:baseId>
8852         <cmis:parentId>parent</cmis:parentId>
8853         <cmis:creatable>true</cmis:creatable>
8854         <cmis:fileable>false</cmis:fileable>
8855         <cmis:queryable>false</cmis:queryable>
8856         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8857
8858 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8859
8860 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8861         <cmis:controllableACL>true</cmis:controllableACL>
8862
8863 <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8864
8865 <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8866
8867 <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8868         </cmisra:type>
8869     </atom:entry>
8870 </atom:feed>
8871 </cmisra:children>
8872 </atom:entry>
8873 <atom:entry>
8874     <atom:author>
8875         <atom:name>Al Brown</atom:name>
8876         <atom:uri>http://www.ibm.com/</atom:uri>
8877         <atom:email>albertcbrown@us.ibm.com</atom:email>
8878     </atom:author>
8879     <atom:content>Type Definition for cmis:policy</atom:content>
8880     <atom:id>http://cmisexample.oasis-
8881 open.org/rep1/type/cmis:policy</atom:id>
8882     <atom:link type="application/atom+xml;type=entry" rel="self"
8883 href="http://cmisexample.oasis-open.org/rep1/type/cmis:policy"/>
8884     <atom:link type="application/atomsvc+xml" rel="service"
8885 href="http://cmisexample.oasis-open.org/rep1/type/cmis:policy"/>
8886     <atom:link type="application/atom+xml;type=entry"
8887 rel="describedby" href="http://cmisexample.oasis-
8888 open.org/rep1/type/cmis:policy"/>
8889     <atom:link type="application/atom+xml;type=entry" rel="up"
8890 href="http://cmisexample.oasis-open.org/rep1/type/cmis:policy/parent"/>
8891     <atom:link type="application/atom+xml;type=feed" rel="down"
8892 href="http://cmisexample.oasis-
8893 open.org/rep1/type/cmis:policy/children/flat"/>
8894     <atom:link type="application/cmistree+xml" rel="down"
8895 href="http://cmisexample.oasis-
8896 open.org/rep1/type/cmis:policy/children/tree"/>
8897     <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
8898     <atom:summary type="html">HTML summary of Type Definition
8899 cmis:policy</atom:summary>
8900     <atom:title type="text">Type Definition - cmis:policy</atom:title>
8901     <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8902     <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8903     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8904 xsi:type="cmis:cmisTypePolicyDefinitionType" cmisra:id="cmis:policy">
8905         <cmis:id>dtcmis:policy</cmis:id>
8906         <cmis:localName>myrepname-cmis:policy</cmis:localName>

```

```

8907 | <cmis:localNamespace xsi:nil="true"/>
8908 | <cmis:displayName>cmis:policy</cmis:displayName>
8909 | <cmis:queryName>cmis:policy</cmis:queryName>
8910 | <cmis:description>Description for type definition
8911 | cmis:policy</cmis:description>
8912 | <cmis:baseId>cmis:policy</cmis:baseId>
8913 | <cmis:parentId>parent</cmis:parentId>
8914 | <cmis:creatable>true</cmis:creatable>
8915 | <cmis:fileable>false</cmis:fileable>
8916 | <cmis:queryable>false</cmis:queryable>
8917 | <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8918 |
8919 | <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8920 | <cmis:controllablePolicy>true</cmis:controllablePolicy>
8921 | <cmis:controllableACL>true</cmis:controllableACL>
8922 | </cmisra:type>
8923 | </atom:entry>
8924 | </atom:feed>

```

8925

8926 Please also see the example documents included with the schema.

8927 **3.9.6.1 GET**

8928 The following arguments may be supplied. Please see the domain model for more information:

- 8929 • includePropertyDefinitions
- 8930 • depth

8931 **3.10 Resources**

8932 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
8933 verb in a repository-specific manner.

8934 **3.10.1 Type Entry**

8935 This represents a type definition in the repository. This is enclosed as an atom entry

8936

8937 CMIS Services:

8938 GET: getTypeDefinition

8939 Media Type: application/atom+xml;type=entry

8940

8941 Link Relations:

- 8942 • service: Points to service document containing the CMIS repository. The service document
8943 MUST contain only one workspace element.
 - 8944 ○ Media Type: application/atomsvc+xml
- 8945 • up: Points to the parent type as atom entry if applicable
- 8946 • down: Points to the children of this type as atom feed if applicable
 - 8947 ○ (Children) Media Type: application/atom +xml;type=feed points to the atom feed
8948 document representing the children feed for this same type
 - 8949 ○ (Descendants) Media Type: application/cmistree+xml points to the atom feed document
8950 representing the descendants feed for this same type
- 8951 • describedby: Points to the type definition atom entry of the base type of this type definition.

8952

8953 The following CMIS Atom extension element MUST be included inside the atom entry:

[cmis-v1.0-errata-1](#)cmis-spec-v1.0
Copyright © OASIS® 2010. All Rights Reserved.

8954 • cmisra:type

8955 3.10.1.1 GET

8956 There are no optional arguments for this resource.

8957

8958 Request:

```
8959 GET /obj/5070f89a-6f00-4acf-84e9-d8836a6c7d23 HTTP/1.1
8960 Host: example.org
8961
```

8962

8963 Response:

```
8964 HTTP/1.1 200 Ok
8965 Date: Mon, 25 Jan 2010 10:21:00 -0800
8966 Content-Length: 2995
8967 Content-Type: application/atom+xml;type=entry
8968 Location: http://cmisexample.oasis-open.org/repl/cmis:document
8969
8970
8971 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8972 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8973 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8974 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8975 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8976 open.org/ns/cmisis/restatom/200908/">
8977   <atom:author>
8978     <atom:name>Al Brown</atom:name>
8979     <atom:uri>http://www.ibm.com/</atom:uri>
8980     <atom:email>albertcbrown@us.ibm.com</atom:email>
8981   </atom:author>
8982   <atom:content>Type Definition for cmis:document</atom:content>
8983   <atom:id>http://cmisexample.oasis-
8984 open.org/repl/type/cmis:document</atom:id>
8985   <atom:link type="application/atom+xml;type=entry" rel="self"
8986 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8987   <atom:link type="application/atomsvc+xml" rel="service"
8988 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8989   <atom:link type="application/atom+xml;type=entry" rel="describedby"
8990 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8991   <atom:link type="application/atom+xml;type=entry" rel="up"
8992 href="http://cmisexample.oasis-open.org/repl/type/cmis:document/parent"/>
8993   <atom:link type="application/atom+xml;type=feed" rel="down"
8994 href="http://cmisexample.oasis-
8995 open.org/repl/type/cmis:document/children/flat"/>
8996   <atom:link type="application/cmistree+xml" rel="down"
8997 href="http://cmisexample.oasis-
8998 open.org/repl/type/cmis:document/children/tree"/>
8999   <atom:published>2010-01-25T10:21:00.380-08:00</atom:published>
9000   <atom:summary type="html">HTML summary of Type Definition
9001 cmis:document</atom:summary>
9002   <atom:title type="text">Type Definition - cmis:document</atom:title>
9003   <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
9004   <app:edited>2010-01-25T10:21:00.380-08:00</app:edited>
9005   <cmisra:type xsi:type="cmis:cmisTypeDocumentDefinitionType"
9006 cmisra:id="cmis:document" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
9007 instance">
9008     <cmis:id>dtcmis:document</cmis:id>
9009     <cmis:localName>myrepname-cmis:document</cmis:localName>
9010     <cmis:localNamespace xsi:nil="true"/>
9011     <cmis:displayName>cmis:document</cmis:displayName>
9012     <cmis:queryName>cmis:document</cmis:queryName>
```

```

9013     <cmis:description>Description for type definition
9014 cmis:document</cmis:description>
9015     <cmis:baseId>cmis:document</cmis:baseId>
9016     <cmis:parentId>parent</cmis:parentId>
9017     <cmis:creatable>true</cmis:creatable>
9018     <cmis:fileable>true</cmis:fileable>
9019     <cmis:queryable>false</cmis:queryable>
9020     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
9021     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
9022     <cmis:controllablePolicy>true</cmis:controllablePolicy>
9023     <cmis:controllableACL>true</cmis:controllableACL>
9024     <cmis:versionable>true</cmis:versionable>
9025     <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
9026   </cmisra:type>
9027 </atom:entry>
9028

```

9029

9030 Please also see the example documents included with the schema.

9031

9032 **3.10.2 Document Entry**

9033 This is a CMIS Document instance.

9034

9035 CMIS Services:

9036 GET: getObject, getObjectOfLatestVersion (getObject)

9037 PUT: updateProperties

9038 DELETE: deleteObject

9039 Media Type: application/atom+xml;type=entry

9040

9041 Link Relations:

- 9042 • self: Points to an URI that returns the atom entry for this document. Please see Atom for more
- 9043 information.
- 9044 • edit: Points to an URI that accepts PUT of atom entry. Please see AtomPub for more information.
- 9045 • service: Points to service document containing the CMIS repository. The service document
- 9046 MUST contain only one workspace element.
 - 9047 ○ Media Type: application/atomsvc+xml
- 9048 • up: Points to the atom feed containing the set of parents. If there is only one parent, the
- 9049 repository MAY point this link relation directly to the atom entry of the parent.
- 9050 • version-history: Points to atom feed containing the versions of this document
 - 9051 ○ If the document is not versionable, this link relation may not be on the resource
- 9052 • current-version: Points to the latest version of the document
 - 9053 ○ Uses query parameter "`returnVersion`" and `enumReturnVersion`
 - 9054 ○ If this version is the current-version, this link relation may not be on the resource
- 9055 • edit-media:
 - 9056 ○ Same as `setContentStream`. Allows updating the content stream on this document
 - 9057 ○ Please see AtomPub for more information
- 9058 • working-copy: Points to the private working copy if it exists.
- 9059 • describedby: Points to the type definition as an atom entry for the type of this document entry.

- 9060 • alternate: this is used to identify the renditions available for the specified object. Please see the
9061 Renditions section.
- 9062 • <http://docs.oasis-open.org/ns/cmismessaging/200908/allowableactions>: Points to the allowable actions
9063 document for this object.
- 9064 • <http://docs.oasis-open.org/ns/cmismessaging/200908/relationships>: Points to the relationships feed for
9065 this object
- 9066 • <http://docs.oasis-open.org/ns/cmismessaging/200908/policies>: Points to the policy feed for this object.
- 9067 • <http://docs.oasis-open.org/ns/cmismessaging/200908/acl>: Points to ACL document for this object
9068

9069 The following CMIS Atom extension element MUST be included inside the atom entry:

- 9070 • cmisra:object
9071

9072 3.10.2.1 GET

9073 The following arguments may be supplied. Please see the domain model for more information:

- 9074 • returnVersion
 - 9075 ○ Used to differentiate between getObject() and getObjectOfLatestVersion().
 - 9076 ○ valid values are described by the schema element cmisra:enumReturnVersion
 - 9077 ○ If not specified, return the version specified by the URI
- 9078 • includeAllowableActions
- 9079 • includeRelationships
- 9080 • includePolicyIds
- 9081 • includeACL
- 9082 • filter
- 9083 • renditionFilter
 - 9084 ○ If not specified, renditions will not be included.
9085
9086

9087 Request:

```
9088 GET /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId HTTP/1.1
9089 Host: example.org
9090
```

9092 Response:

```
9093 HTTP/1.1 200 Ok
9094 Date: Mon, 25 Jan 2010 10:21:00 -0800
9095 Content-Length: 3403
9096 Content-Type: application/atom+xml;type=entry
9097 Location: /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId
9098
```

```
9099 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9100 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9101 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9102 open.org/ns/cmismessaging/200908/" xmlns:cmism="http://docs.oasis-
9103 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
9104 open.org/ns/cmismessaging/200908/">
9105 <atom:author>
9106 <atom:name>Al Brown</atom:name>
9107
```

```

9108     <atom:uri>http://www.ibm.com/</atom:uri>
9109     <atom:email>albertcbrown@us.ibm.com</atom:email>
9110   </atom:author>
9111   <atom:content src="http://cmisexample.oasis-open.org/repl/7c088887-5991-
9112 4b3a-9ad3-16379127e647"/>
9113   <atom:id>urn:uuid:7c088887-5991-4b3a-9ad3-16379127e647</atom:id>
9114   <atom:title type="text">Invoice</atom:title>
9115   <atom:updated>2010-01-25T10:21:00.193-08:00</atom:updated>
9116   <atom:link rel="self" href="http://cmisexample.oasis-
9117 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647"/>
9118   <atom:link rel="edit" href="http://cmisexample.oasis-
9119 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647"/>
9120   <atom:link type="application/cmism+xml;type=allowableActions"
9121 rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
9122 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9123 16379127e647/allowableactions"/>
9124   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9125 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9126 16379127e647/type"/>
9127   <atom:link type="application/atomsvc+xml" rel="service"
9128 href="http://cmisexample.oasis-open.org/repl//service"/>
9129   <atom:published>2010-01-25T10:21:00.193-08:00</atom:published>
9130   <atom:summary type="html">HTML summary of Entry 7c088887-5991-4b3a-
9131 9ad3-16379127e647</atom:summary>
9132   <atom:link rel="edit-media" href="http://cmisexample.oasis-
9133 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/edit-media"/>
9134   <atom:link rel="alternate" href="http://cmisexample.oasis-
9135 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/alternate"/>
9136   <atom:link type="application/atom+xml;type=feed" rel="up"
9137 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9138 16379127e647/parents"/>
9139   <atom:link type="application/atom+xml;type=feed" rel="version-history"
9140 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9141 16379127e647/allversions"/>
9142   <atom:link type="application/atom+xml;type=entry" rel="current-
9143 version" href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-
9144 9ad3-16379127e647/latest"/>
9145   <atom:link type="application/atom+xml;type=feed"
9146 rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
9147 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9148 16379127e647/relationships"/>
9149   <atom:link type="application/atom+xml;type=feed"
9150 rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
9151 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9152 16379127e647/policies"/>
9153   <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9154 open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
9155 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/acl"/>
9156   <cmisra:object>
9157     <cmis:properties>
9158       <cmis:propertyId localName="rep-cmis:objectId"
9159 propertyDefinitionId="cmis:objectId">
9160         <cmis:value>7c088887-5991-4b3a-9ad3-16379127e647</cmis:value>
9161       </cmis:propertyId>
9162     </cmis:properties>
9163   </cmisra:object>
9164 </atom:entry>
9165

```

9166

9167 Please also see the example documents included with the schema.

9168

9169 **3.10.2.2 PUT**

9170 This does a replacement of the atom entry with the atom entry document specified. If readwrite
9171 properties are not included, the repository SHOULD NOT modify them.

9172

9173 The server SHOULD respond with:

- 9174 • HTTP Status Code 200
- 9175 • Response Body containing the updated atom entry

9176

9177 **3.10.2.3 DELETE**

9178 This removes the document.

9179 Success HTTP code: 204

9180 **3.10.3 Document Private Working Copy (PWC) Entry**

9181 This is the private working copy of the document (checkedout version of document)

9182 CMIS Services:

- 9183 GET: getObject
- 9184 PUT: updateProperties or checkIn
- 9185 DELETE: cancelCheckOut

9186 Media Type: application/atom+xml;type=entry

9187

9188 Link relations:

- 9189 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9190 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more
9191 information.
- 9192 • service: Points to service document containing the CMIS repository. The service document
9193 MUST contain only one workspace element.
 - 9194 ○ Media Type: application/atomsvc+xml
- 9195 • up: Points to the atom feed containing the set of parents. If there is only one parent, the
9196 repository MAY point this link relation directly to the atom entry of the parent.
- 9197 • version-history
 - 9198 ○ Points to an URI that returns the feed associated with the version history
- 9199 • edit-media
 - 9200 ○ Same as setContentStream. Allows updating the content stream on this document
 - 9201 ○ Please see AtomPub for more information
- 9202 • via: atom entry that created this private working copy
- 9203 • describedby: Points to the type definition as an atom entry for the type of this PWC entry.
- 9204 • alternate: this is used to identify the renditions available for the specified object. Please see the
9205 Renditions section.
- 9206 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
9207 document for this object.
- 9208 • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for
9209 this object
- 9210 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.

- 9211 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

9212

9213 The following element MUST be included inside the atom entry:

- 9214 • cmisra:object

9215

9216 3.10.3.1 GET

9217 The following arguments may be supplied. Please see the domain model for more information:

- 9218 • filter
- 9219 • includeAllowableActions
- 9220 • includeRelationships
- 9221 • renditionFilter
- 9222 ○ If not specified, renditions will not be included.

9223

9224 Request:

```
9225 GET /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId HTTP/1.1
9226 Host: example.org
9227
```

9228

9229 Response:

```
9230 HTTP/1.1 200 Ok
9231 Date: Mon, 25 Jan 2010 10:21:00 -0800
9232 Content-Length: 3564
9233 Content-Type: application/atom+xml;type=entry
9234 Location: /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId
9235
9236
9237 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9238 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9239 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9240 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9241 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9242 open.org/ns/cmisis/restatom/200908/">
9243   <atom:author>
9244     <atom:name>Al Brown</atom:name>
9245     <atom:uri>http://www.ibm.com/</atom:uri>
9246     <atom:email>albertcbrown@us.ibm.com</atom:email>
9247   </atom:author>
9248   <atom:content src="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-
9249 4ab2-978d-85ca2f4f3206"/>
9250   <atom:id>urn:uuid:3240a476-6de6-4ab2-978d-85ca2f4f3206</atom:id>
9251   <atom:title type="text">Invoice</atom:title>
9252   <atom:updated>2010-01-25T10:21:00.333-08:00</atom:updated>
9253   <atom:link rel="self" href="http://cmisexample.oasis-
9254 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>
9255   <atom:link rel="edit" href="http://cmisexample.oasis-
9256 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>
9257   <atom:link type="application/cmisis+xml;type=allowableActions"
9258 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9259 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9260 85ca2f4f3206/allowableactions"/>
9261   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9262 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9263 85ca2f4f3206/type"/>
```



```

9264 | <atom:link type="application/atomsvc+xml" rel="service"
9265 | href="http://cmisexample.oasis-open.org/rep1//service"/>
9266 | <atom:published>2010-01-25T10:21:00.333-08:00</atom:published>
9267 | <atom:summary type="html">HTML summary of Entry 3240a476-6de6-4ab2-
9268 | 978d-85ca2f4f3206</atom:summary>
9269 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
9270 | open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/edit-media"/>
9271 | <atom:link rel="alternate" href="http://cmisexample.oasis-
9272 | open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/alternate"/>
9273 | <atom:link type="application/atom+xml;type=feed" rel="up"
9274 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9275 | 85ca2f4f3206/parents"/>
9276 | <atom:link type="application/atom+xml;type=feed" rel="version-history"
9277 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9278 | 85ca2f4f3206/allversions"/>
9279 | <atom:link type="application/atom+xml;type=entry" rel="current-
9280 | version" href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-
9281 | 978d-85ca2f4f3206/latest"/>
9282 | <atom:link type="application/atom+xml;type=feed"
9283 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
9284 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9285 | 85ca2f4f3206/relationships"/>
9286 | <atom:link type="application/atom+xml;type=feed"
9287 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
9288 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9289 | 85ca2f4f3206/policies"/>
9290 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9291 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9292 | open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/acl"/>
9293 | <atom:link type="application/atom+xml;type=feed" rel="working-copy"
9294 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9295 | 85ca2f4f3206/pwc"/>
9296 | <cmisra:object>
9297 |   <cmis:properties>
9298 |     <cmis:propertyId localName="rep-cmis:objectId"
9299 |     propertyDefinitionId="cmis:objectId">
9300 |       <cmis:value>3240a476-6de6-4ab2-978d-85ca2f4f3206</cmis:value>
9301 |     </cmis:propertyId>
9302 |   </cmis:properties>
9303 | </cmisra:object>
9304 | </atom:entry>
9305 |

```

9306

9307 Please also see the example documents included with the schema.

9308

9309 **3.10.3.2 PUT**

9310 This does a replacement of the atom entry with the atom entry document specified. If modifiable
9311 properties (whencheckedout or readwrite) are not included, the repository SHOULD NOT modify them.

9312

9313 The following arguments may be supplied. Please see the domain model for more information:

9314 • checkinComment

9315 • major

9316 • checkin

9317 ○ Used to differentiate between updateProperties() or checkin() services. If TRUE, execute
9318 checkin service.

- 9319 ○ If this argument is specified as TRUE, then the body to PUT MAY be omitted if there are
9320 no modifications to be made during checkin

9321

9322 The server SHOULD respond with:

- 9323 • HTTP Status Code 200
- 9324 • Location header of the resource (if changed via checkin)
- 9325 • Response Body containing the updated atom entry

9326 **3.10.3.3 DELETE**

9327 This removes the document entry, in this case, cancels the check out. The PWC will be removed.

9328

9329 Success HTTP code: 204

9330 **3.10.4 Folder Entry**

9331 This is a CMIS Folder instance. The properties of a folder map onto the feed tag.

9332 CMIS Services:

- 9333 GET: getObject
- 9334 PUT: updateProperties
- 9335 DELETE: deleteObject (this is deletion of the folder only and not any contained objects)

9336 Media Type: application/atom+xml;type=entry

9337

9338 Link Relations:

- 9339 • self: Points to the URI to retrieve this atom entry. Please see Atom for more informationedit:
9340 Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
- 9341 • service: Points to service document containing the CMIS repository. The service document
9342 MUST contain only one workspace element.
 - 9343 ○ Media Type: application/atomsvc+xml
- 9344 • describedby: Points to the type definition as an atom entry for the type of this folder entry.
- 9345 • down: Points to the children of this folder
 - 9346 ○ application/atom+xml : Points to the atom feed document representing the children feed
9347 for this same folder
 - 9348 ○ application/cmistree+xml: Points to the descendants feed of the same folder
- 9349 • up: Points to the atom entry for the parent
 - 9350 ○ If the root folder, this link will not be present
- 9351 • alternate: this is used to identify the renditions available for the specified object. Please see the
9352 Renditions section.
- 9353 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
9354 document for this object.
- 9355 • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for
9356 this object
- 9357 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 9358 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object
- 9359 • <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>: Points to the folder tree for this folder

9360

9361 The following CMIS Atom extension element MUST be included inside the atom entry:

- 9362 • cmisra:object

9363

9364 3.10.4.1 GET

9365 The following arguments may be supplied. Please see the domain model for more information:

- 9366 • filter
- 9367 • includeAllowableActions
- 9368 • includeRelationships
- 9369 • renditionFilter
 - 9370 ○ If not specified, renditions will not be included.

9371

9372 Request:

```
9373 GET /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId HTTP/1.1
9374 Host: example.org
9375
```

9376

9377 Response:

```
9378 HTTP/1.1 200 Ok
9379 Date: Mon, 25 Jan 2010 10:21:00 -0800
9380 Content-Length: 3332
9381 Content-Type: application/atom+xml;type=entry
9382 Location: /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId
9383
9384
9385 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9386 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9387 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9388 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9389 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9390 open.org/ns/cmisis/restatom/200908/">
9391   <atom:author>
9392     <atom:name>Al Brown</atom:name>
9393     <atom:uri>http://www.ibm.com/</atom:uri>
9394     <atom:email>albertcbrown@us.ibm.com</atom:email>
9395   </atom:author>
9396   <atom:content src="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-
9397 471d-b4ba-6d8756cd5093"/>
9398   <atom:id>urn:uuid:cfc03a28-8240-471d-b4ba-6d8756cd5093</atom:id>
9399   <atom:title type="text">Customer Folder</atom:title>
9400   <atom:updated>2010-01-25T10:21:00.208-08:00</atom:updated>
9401   <atom:link rel="self" href="http://cmisexample.oasis-
9402 open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
9403   <atom:link rel="edit" href="http://cmisexample.oasis-
9404 open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
9405   <atom:link type="application/cmisis+xml;type=allowableActions"
9406 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9407 href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9408 6d8756cd5093/allowableactions"/>
9409   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9410 href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9411 6d8756cd5093/type"/>
9412   <atom:link type="application/atomsvc+xml" rel="service"
9413 href="http://cmisexample.oasis-open.org/rep1//service"/>
9414   <atom:published>2010-01-25T10:21:00.208-08:00</atom:published>
```

```

9415 | <atom:summary type="html">HTML summary of Entry cfc03a28-8240-471d-
9416 | b4ba-6d8756cd5093</atom:summary>
9417 | <atom:link type="application/atom+xml;type=entry" rel="up"
9418 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9419 | 6d8756cd5093/up"/>
9420 | <atom:link type="application/atom+xml;type=feed" rel="down"
9421 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9422 | 6d8756cd5093/children"/>
9423 | <atom:link type="application/cmistree+xml" rel="down"
9424 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9425 | 6d8756cd5093/tree"/>
9426 | <atom:link type="application/atom+xml;type=feed"
9427 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
9428 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9429 | 6d8756cd5093/foldertree"/>
9430 | <atom:link type="application/atom+xml;type=feed"
9431 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
9432 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9433 | 6d8756cd5093/relationships"/>
9434 | <atom:link type="application/atom+xml;type=feed"
9435 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
9436 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
9437 | 6d8756cd5093/policies"/>
9438 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9439 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9440 | open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093/acl"/>
9441 | <cmisra:object>
9442 |   <cmis:properties>
9443 |     <cmis:propertyId localName="rep-cmis:objectId"
9444 | propertyDefinitionId="cmis:objectId">
9445 |       <cmis:value>cfc03a28-8240-471d-b4ba-6d8756cd5093</cmis:value>
9446 |     </cmis:propertyId>
9447 |   </cmis:properties>
9448 | </cmisra:object>
9449 | </atom:entry>
9450 |

```

9451

9452 Please also see the example documents included with the schema.

9453

9454 3.10.4.2 PUT

9455 This does a replacement of the atom entry with the atom entry document specified. If readwrite
9456 properties are not included, the repository SHOULD NOT modify them.

9457

9458 The server SHOULD respond with:

- 9459 • HTTP Status Code 200
- 9460 • Response Body containing the updated atom entry

9461

9462 3.10.4.3 DELETE

9463 This removes the object (folder) from the repository.

9464 Success HTTP code: 204

9465 3.10.5 Relationship Entry

9466 This is a CMIS relationship instance. These objects are exposed via "relationships" link type.

9467 CMIS Services:
9468 GET: getObject
9469 PUT: updateProperties
9470 DELETE: deleteObject
9471 Media Type: application/atom+xml;type=entry
9472

9473 Link Relations:

- 9474 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9475 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more
9476 information.
- 9477 • service: Points to service document containing the CMIS repository. The service document
9478 MUST contain only one workspace element.
 - 9479 ○ Media Type: application/atomsvc+xml
- 9480 • describedby: Points to the type definition as an atom entry for the type of this relationship entry.
- 9481 • <http://docs.oasis-open.org/ns/cmisis/link/200908/target>
- 9482 • <http://docs.oasis-open.org/ns/cmisis/link/200908/source>
- 9483 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
9484 document for this object.
- 9485 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 9486 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

9487
9488 The following element MUST be included inside the atom entry:

- 9489 • cmisra:object
- 9490

9491 3.10.5.1 GET

9492 The following arguments may be supplied. Please see the domain model for more information:

- 9493 • filter
- 9494 • includeAllowableActions

9495

9496 Request:

```
9497 GET /obj/ad443afd-aala-4071-9735-1a49aac4e439?filter=cmis:objectId HTTP/1.1  
9498 Host: example.org  
9499
```

9500

9501 Response:

```
9502 HTTP/1.1 200 Ok  
9503 Date: Mon, 25 Jan 2010 10:21:00 -0800  
9504 Content-Length: 2861  
9505 Content-Type: application/atom+xml;type=entry  
9506 Location: /obj/ad443afd-aala-4071-9735-1a49aac4e439?filter=cmis:objectId  
9507  
9508  
9509 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

9510 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
9511 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9512 | open.org/ns/cmismessaging/200908/" xmlns:cmism="http://docs.oasis-
9513 | open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
9514 | open.org/ns/cmismessaging/200908/">
9515 |   <atom:author>
9516 |     <atom:name>Al Brown</atom:name>
9517 |     <atom:uri>http://www.ibm.com/</atom:uri>
9518 |     <atom:email>albertcbrown@us.ibm.com</atom:email>
9519 |   </atom:author>
9520 |   <atom:content src="http://cmisexample.oasis-open.org/repl/ad443afd-aala-
9521 | 4071-9735-1a49aac4e439">
9522 |     <atom:id>urn:uuid:ad443afd-aala-4071-9735-1a49aac4e439</atom:id>
9523 |     <atom:title type="text">Customer Relationship</atom:title>
9524 |     <atom:updated>2010-01-25T10:21:00.349-08:00</atom:updated>
9525 |     <atom:link rel="self" href="http://cmisexample.oasis-
9526 | open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439">
9527 |     <atom:link rel="edit" href="http://cmisexample.oasis-
9528 | open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439">
9529 |     <atom:link type="application/cmismessaging+xml;type=allowableActions"
9530 | rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/allowableactions"
9531 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9532 | 1a49aac4e439/allowableactions">
9533 |     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9534 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9535 | 1a49aac4e439/type">
9536 |     <atom:link type="application/atomsvc+xml" rel="service"
9537 | href="http://cmisexample.oasis-open.org/repl//service">
9538 |     <atom:published>2010-01-25T10:21:00.365-08:00</atom:published>
9539 |     <atom:summary type="html">HTML summary of Entry ad443afd-aala-4071-
9540 | 9735-1a49aac4e439</atom:summary>
9541 |     <atom:link type="application/atom+xml;type=entry"
9542 | rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/source"
9543 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9544 | 1a49aac4e439/source">
9545 |     <atom:link type="application/atom+xml;type=entry"
9546 | rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/target"
9547 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9548 | 1a49aac4e439/target">
9549 |     <atom:link type="application/atom+xml;type=feed"
9550 | rel="http://docs.oasis-open.org/ns/cmismessaging/link/200908/policies"
9551 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9552 | 1a49aac4e439/policies">
9553 |     <atom:link type="application/cmismessaging+xml" rel="http://docs.oasis-
9554 | open.org/ns/cmismessaging/link/200908/acl" href="http://cmisexample.oasis-
9555 | open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/acl">
9556 |     <cmismessaging:object>
9557 |       <cmismessaging:properties>
9558 |         <cmismessaging:propertyId localName="rep-cmis:objectId"
9559 | propertyDefinitionId="cmismessaging:objectId">
9560 |           <cmismessaging:value>ad443afd-aala-4071-9735-1a49aac4e439</cmismessaging:value>
9561 |         </cmismessaging:propertyId>
9562 |       </cmismessaging:properties>
9563 |     </cmismessaging:object>
9564 |   </atom:entry>
9565 |

```

9566

9567 | Please also see the example documents included with the schema.

9568

9569 **3.10.5.2 PUT**

9570 This does a replacement of the atom entry with the atom entry document specified. If readwrite
9571 properties are not included, the repository SHOULD NOT modify them.

9572

9573 The server SHOULD respond with:

- 9574 • HTTP Status Code 200
- 9575 • Response Body containing the updated atom entry

9576

9577 **3.10.5.3 DELETE**

9578 This removes the relationship entry.

9579 Successful HTTP code: 204

9580 **3.10.6 Policy Entry**

9581 This is a CMIS policy instance.

9582 CMIS Services:

- 9583 GET: getObject
- 9584 PUT: updateProperties
- 9585 DELETE: deleteObject or removePolicy

9586 Media Type: application/atom+xml;type=entry

9587

9588 Link Relations:

- 9589 • self
- 9590 • edit
- 9591 • service: Points to service document containing the CMIS repository. The service document
9592 MUST contain only one workspace element.
 - 9593 ○ Media Type: application/atomsvc+xml
- 9594 • describedby: Points to the type definition as an atom entry for the type of this policy entry.
- 9595 • alternate: this is used to identify the renditions available for the specified object. Please see the
9596 Renditions section.
- 9597 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
9598 document for this object.
- 9599 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 9600 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

9601

9602 The following element MUST be included inside the atom entry:

- 9603 • `cmisra:object`

9604

9605 **3.10.6.1 GET**

9606 The following arguments may be supplied. Please see the domain model for more information:

- 9607 • filter
- 9608 • includeAllowableActions

- 9609 • includeRelationships
- 9610 • renditionFilter
 - 9611 ○ If not specified, renditions will not be included.

9612

9613 Request:

```
9614 GET /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId HTTP/1.1
9615 Host: example.org
9616
```

9617

9618 Response:

```
9619 HTTP/1.1 200 Ok
9620 Date: Mon, 25 Jan 2010 10:21:00 -0800
9621 Content-Length: 2608
9622 Content-Type: application/atom+xml;type=entry
9623 Location: /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId
9624
9625
9626 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9627 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9628 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9629 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9630 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9631 open.org/ns/cmisis/restatom/200908/">
9632   <atom:author>
9633     <atom:name>Al Brown</atom:name>
9634     <atom:uri>http://www.ibm.com/</atom:uri>
9635     <atom:email>albertcbrown@us.ibm.com</atom:email>
9636   </atom:author>
9637   <atom:content src="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-
9638 4940-b2f0-4e4cd4631bf0"/>
9639   <atom:id>urn:uuid:a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</atom:id>
9640   <atom:title type="text">Security Policy</atom:title>
9641   <atom:updated>2010-01-25T10:21:00.318-08:00</atom:updated>
9642   <atom:link rel="self" href="http://cmisexample.oasis-
9643 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
9644   <atom:link rel="edit" href="http://cmisexample.oasis-
9645 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
9646   <atom:link type="application/cmisis+xml;type=allowableActions"
9647 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9648 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9649 4e4cd4631bf0/allowableactions"/>
9650   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9651 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9652 4e4cd4631bf0/type"/>
9653   <atom:link type="application/atomsvc+xml" rel="service"
9654 href="http://cmisexample.oasis-open.org/rep1//service"/>
9655   <atom:published>2010-01-25T10:21:00.318-08:00</atom:published>
9656   <atom:summary type="html">HTML summary of Entry a09ed524-5f1b-4940-
9657 b2f0-4e4cd4631bf0</atom:summary>
9658   <atom:link type="application/atom+xml;type=feed" rel="up"
9659 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9660 4e4cd4631bf0/parents"/>
9661   <atom:link type="application/atom+xml;type=feed"
9662 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
9663 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9664 4e4cd4631bf0/relationships"/>
9665   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
9666 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9667 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0/acl"/>
9668   <cmisra:object>
```



```
9669     <cmis:properties>
9670         <cmis:propertyId localName=""_rep-cmis:objectId""
9671         propertyDefinitionId=""cmis:objectId"">
9672             <cmis:value>a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</cmis:value>
9673         </cmis:propertyId>
9674     </cmis:properties>
9675 </cmisra:object>
9676 </atom:entry>
9677
```

9678

9679 Please also see the example documents included with the schema.

9680

9681 **3.10.6.2 PUT**

9682 This does a replacement of the atom entry with the atom entry document specified. If read/write
9683 properties are not included, the repository SHOULD NOT modify them.

9684

9685 The server SHOULD respond with:

- 9686 • HTTP Status Code 200
- 9687 • Response Body containing the updated atom entry

9688

9689 **3.10.6.3 DELETE**

9690 This removes the policy entry. If this policy entry was discovered through a policy collection on an object,
9691 then removePolicy() is performed rather than deleteObject() on the policy itself.

9692

9693 Success HTTP code: 204

9694 **3.10.7 Content Stream**

9695 This is the content stream portion of the document object.

9696 CMIS Services:

9697 GET: getContentStream

9698 PUT: setContentStream

9699 DELETE: deleteContentStream

9700 Media Type: Mime/Type of resource (mime type of content stream on document)

9701 **3.10.7.1 GET**

9702 This returns the content stream.

9703

9704 It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED
9705 that HTTP compression is also supported.

9706

9707 Please see RFC2616 for more information on HTTP Range requests.

9708 **3.10.7.2 PUT**

9709 This does a replacement of the content stream.

9710

9711 The following optional arguments may be supplied. Please see the domain model for more information:

- 9712 • overwriteFlag.
 - 9713 ○ If not specified, this defaults to `“true”` in this binding and behaves consistent with
 - 9714 AtomPub.

9715

9716 Success HTTP code: 200 (with content), 204 (without content) or 201 if a new resource is created.
9717 Please see the HTTP specification for more information.

9718

9719 Returns headers:

- 9720 • Content-Location: URI for content stream
- 9721 • Location: URI for content stream

9722 **3.10.7.3 DELETE**

9723 This removes the content stream.

9724 **3.10.8 ACL Resource**

9725 CMIS Services:

9726 GET: getACL

9727 PUT: applyACL

9728

9729 Media Type: application/cmisacl+xml

9730

9731 **3.10.8.1 GET**

9732 This returns the CMIS ACL for a specified object. The client will follow the link on the atom entry to get
9733 the CMIS ACL for that object.

9734

9735 Request:

```
9736 GET /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef HTTP/1.1  
9737 Host: example.org  
9738
```

9739

9740 Response:

```
9741 HTTP/1.1 200 Ok  
9742 Date: Mon, 25 Jan 2010 10:21:00 -0800  
9743 Content-Length: 758  
9744 Content-Type: application/cmisacl+xml  
9745 Location: /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef  
9746  
9747  
9748 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
9749 <cmis:acl xmlns:app="http://www.w3.org/2007/app"  
9750 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
9751 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-  
9752 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-  
9753 open.org/ns/cmisis/restatom/200908/">  
9754 <cmis:permission>  
9755 <cmis:principal>  
9756 <cmis:principalId>Al Brown</cmis:principalId>  
9757 </cmis:principal>
```

```
9758 <cmis:permission>cmis:read</cmis:permission>
9759 <cmis:permission>cmis:write</cmis:permission>
9760 <cmis:permission>cmis:all</cmis:permission>
9761 <cmis:permission>publish</cmis:permission>
9762 <cmis:direct>true</cmis:direct>
9763 </cmis:permission>
9764 </cmis:acl>
9765
```

9766

9767 Please also see the example documents included with the schema.

9768 **4 Web Services Binding**

9769 **4.1 Overview**

9770 All services and operations defined in the Domain Model are presented in the Web Services binding.

9771 The WSDL for these services reference two XSD documents. One defines elements for the primary data
9772 types of documents, folders, relationships and policies as well as collections of these types of objects.

9773 The second XSD defines the message formats for each of the CMIS services; the messages often refer
9774 to the data types defined in the first XSD schema. The WSDL presents exactly the abstract services
9775 defined in the Services section.

9776 The normative CMIS Web Services binding is defined by the WSDL and XSD as well as the details given
9777 here in this part of the CMIS specification except the examples.

9778 **4.1.1 WS-I**

9779 A CMIS Web Services binding **MUST** comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

9780 **4.1.2 Authentication**

9781 A CMIS Web Services binding **SHOULD** support WS-Security 1.1 for Username Token Profile 1.1 and
9782 **MAY** also support other authentication mechanisms. A CMIS repository **MAY** grant access to all or a
9783 subset of the CMIS services to unauthenticated clients.

9784 **4.1.3 Content Transfer**

9785 All endpoints of the Web Services binding **MUST** be MTOM enabled.

9786 **4.1.4 Reporting Errors**

9787 Services **MUST** report errors via SOAP faults. The CMIS-Messaging.xsd defines a basic fault structure
9788 that includes an error code and an error message and the WSDL for each service defines specific
9789 messages that have the basic fault format.

9790 **4.2 Web Services Binding Mapping**

9791 The Domain Model defines all services, operations, parameters and objects of CMIS. The Web Services
9792 binding is an exact one-to-one mapping of this definition with small exceptions that are explained in the
9793 next section. Operations and parameters are named exactly after their counterparts in the Services
9794 section. All rules and exceptions defined there apply to the Web Services binding. Optional parameters
9795 and optional return values are not set if they are missing or their value is NULL.

9796 **4.3 Additions to the Services section**

9797 **4.3.1 updateProperties and checkIn Semantics**

9798 This binding supports partial properties updates. All properties passed to updateProperties or checkIn will
9799 be updated to their new values. Properties that are passed without a value will be set to their default
9800 value or un-set if no default value is defined. All others property values remain untouched.

9801 **4.3.2 Content Ranges**

9802 This binding supports the retrieval of content ranges. The operation getContentStream accepts two
9803 optional parameters:

- 9804
- **Integer offset:** The first byte of the content to retrieve. Default value is 0.
- 9805
- **Integer length:** The length of the range in bytes. Default value is the size of the content minus
- 9806 the offset.

9807

9808 If the **offset** value is greater than the size of the content the repository SHOULD throw a `constraint`

9809 exception.

9810 If **offset + length** is greater than the size of the content the repository should deliver the content from the

9811 offset to the end of the content.

9812

9813 **4.3.3 Extensions**

9814 On all input messages and some output messages exists an element called extension. This element is

9815 used to provide vendor or repository-specific information between client and server.

9816 All of the types referenced by the schema also support `xs:any` for vendor or repository-specific

9817 information.

9818 **4.3.4 Web Services Specific Structures**

9819 This binding requires specific structures that are not part of the general CMIS schema.

9820 Please also see the example request and response documents included with the schema.

9821 **4.3.4.1 cmisFaultType and cmisFault**

9822 `cmisFaultType` and `cmisFault` SHOULD be used to generate SOAP faults. See 0

9823 [Reporting Errors](#)

9824 [Reporting Errors](#).

9825 **4.3.4.2 cmisRepositoryEntryType**

9826 `cmisRepositoryEntryType` is the return structure of `getRepositoryes`. It contains the id and the name

9827 of a repository.

9828 **4.3.4.3 cmisTypeContainer**

9829 `cmisTypeContainer` is the return structure of `getTypeDescendants`. It holds a type hierarchy.

9830 **4.3.4.4 cmisTypeDefinitionListType**

9831 `cmisTypeDefinitionListType` is the return structure of `getTypeChildren`. It contains a list of types,

9832 the `hasMoreItems` flag and the `numItem` element.

9833 **4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and**

9834 **cmisObjectInFolderContainerType**

9835 `cmisObjectInFolderType` holds, in addition to a `cmisObjectType` object, a path segment string. It

9836 is used in all operations that support the `includePathSegments` parameter.

9837 `cmisObjectParentsType` is similar but has a relative path segment string instead of a path segment.

9838 For details about path segments and relative path segments see section 2.1.5.3 Paths.

9839 `cmisObjectInFolderContainerType` contains a folder hierarchy.

9840 **4.3.4.6 cmisObjectListType and cmisObjectInFolderListType**

9841 cmisObjectListType and cmisObjectInFolderListType hold lists of cmisObjectType and
9842 cmisObjectInFolderType structures. They also contain the hasMoreItems flag and the numItems
9843 element that are returned by operations that return these lists.

9844 **4.3.4.7 cmisContentStreamType**

9845 cmisContentStreamType wraps a content stream and additional information about the stream.

		Client to Repository	Repository to Client
length	Length of the content stream in bytes. If set it MUST be a positive number. If the length is unknown it MUST NOT be set.	SHOULD be set	SHOULD be set
mimeType	MIME Media Type of the content stream. For the primary content of a document it SHOULD match the value of the property cmis:contentStreamMimeType.	SHOULD be set	MUST be set
filename	Filename of the content stream. For the primary content of a document it SHOULD match the value of the property cmis:contentStreamFileName.	SHOULD be set	SHOULD be set
stream	The content stream. MUST be present even if the content stream has 0 bytes.	MUST be set	MUST be set

9846

9847 **4.3.4.8 cmisACLType**

9848 cmisACLType is the return structure of getACL and applyACL. It contains the current Access Control List
9849 (ACL) of the object and the exact flag that indicates if the ACL fully describes the permission of this
9850 object.

9851 **4.3.4.9 cmisExtensionType**

9852 cmisExtensionType is a placeholder for extensions. See 4.3.3 Extensions.

9853

9854 5 IANA Considerations

9855 5.1 Content-Type Registration

9856 5.1.1 CMIS Query

9857 A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

9858

9859 MIME media type name: application

9860 MIME subtype name: cmisquery+xml

9861 Mandatory parameters: None

9862 Optional parameters:

9863 | `charset`: This parameter has semantics identical to the `charset` parameter of the
9864 | `application/xml` media type as specified in [RFC3023].

9865 Encoding considerations:

9866 | Identical to those of `application/xml` as described in [RFC3023], Section 3.2.

9867 Security considerations: As defined in this specification.

9868 | In addition, as this media type uses the `+xml` convention, it shares the same security
9869 | considerations as described in [RFC3023], Section 10.

9870 Interoperability considerations:

9871 | There are no known interoperability issues.

9872 Published specification: This specification.

9873 Applications that use this media type:

9874 | No known applications currently use this media type.

9875 Additional information:

9876 Magic number(s):

9877 | As specified for `application/xml` in [RFC3023], Section 3.2.

9878 File extension: `.cmisquery`

9879 Fragment identifiers:

9880 | As specified for `application/xml` in [RFC3023], Section 5.

9881 Base URI:

9882 | As specified in [RFC3023], Section 6.

9883 Macintosh File Type code: TEXT

9884 Person and email address to contact for further information:

9885 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)

9886 Intended usage: COMMON

9887 Author/Change controller: IESG

9888 5.1.2 CMIS AllowableActions

9889 A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following
9890 media type:

9891

9892 MIME media type name: application
9893 MIME subtype name: cmisallowableactions+xml
9894 Mandatory parameters: None.
9895 Optional parameters:
9896 | "`charset`": This parameter has semantics identical to the `charset` parameter of the
9897 | "`application/xml`" media type as specified in [RFC3023].
9898 Encoding considerations:
9899 | Identical to those of "`application/xml`" as described in [RFC3023], Section 3.2.
9900 Security considerations: As defined in this specification.
9901 | In addition, as this media type uses the "`+xml`" convention, it shares the same security
9902 | considerations as described in [RFC3023], Section 10.
9903 Interoperability considerations:
9904 | There are no known interoperability issues.
9905 Published specification: This specification.
9906 Applications that use this media type:
9907 | No known applications currently use this media type.
9908 Additional information:
9909 Magic number(s):
9910 | As specified for "`application/xml`" in [RFC3023], Section 3.2.
9911 File extension: `.cmisallowableactions`
9912 Fragment identifiers:
9913 | As specified for "`application/xml`" in [RFC3023], Section 5.
9914 Base URI:
9915 | As specified in [RFC3023], Section 6.
9916 Macintosh File Type code: TEXT
9917 Person and email address to contact for further information:
9918 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
9919 Intended usage: COMMON
9920 Author/Change controller: IESG
9921

9922 **5.1.3 CMIS Tree**

9923 A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:

9924
9925 MIME media type name: application
9926 MIME subtype name: cmistree+xml
9927 Mandatory parameters: None.
9928 Optional parameters:
9929 | "`charset`": This parameter has semantics identical to the `charset` parameter of the "`application/xml`"
9930 | media type as specified in [RFC3023].
9931 Encoding considerations:
9932 | Identical to those of "`application/xml`" as described in [RFC3023], Section 3.2.
9933 Security considerations: As defined in this specification.

9934 | In addition, as this media type uses the `""+xml""` convention, it shares the same security considerations as
9935 described in [RFC3023], Section 10.

9936 Interoperability considerations:
9937 There are no known interoperability issues.

9938 Published specification: This specification.

9939 Applications that use this media type:
9940 No known applications currently use this media type.

9941 Additional information:
9942 Magic number(s):
9943 | As specified for `""application/xml""` in [RFC3023], Section 3.2.
9944 File extension: `.cmistree`
9945 Fragment identifiers:
9946 | As specified for `""application/xml""` in [RFC3023], Section 5.
9947 Base URI:
9948 As specified in [RFC3023], Section 6.
9949 Macintosh File Type code: `TEXT`
9950 Person and email address to contact for further information:
9951 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
9952 Intended usage: `COMMON`
9953 Author/Change controller: `IESG`
9954

9955 **5.1.4 CMIS Atom**

9956 A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:

9957

9958 MIME media type name: `application`
9959 MIME subtype name: `cmisatom+xml`
9960 Mandatory parameters: `None`.
9961 Optional parameters:
9962 | `""charset""`: This parameter has semantics identical to the `charset` parameter of the `""application/xml""`
9963 media type as specified in [RFC3023].
9964 | `""type""`: This parameter has semantics identical to the `type` parameter of the `""application/atom+xml""` as
9965 specified in [RFC4287]
9966 Encoding considerations:
9967 | Identical to those of `""application/xml""` as described in [RFC3023], Section 3.2.
9968 Security considerations: As defined in this specification.

9969 | In addition, as this media type uses the `""+xml""` convention, it shares the same security considerations as
9970 described in [RFC3023], Section 10.

9971 Interoperability considerations:
9972 There are no known interoperability issues.

9973 Published specification: This specification.

9974 Applications that use this media type:
9975 No known applications currently use this media type.

9976 Additional information:

9977 Magic number(s):
9978 | As specified for `application/xml` in [RFC3023], Section 3.2.
9979 File extension: `.cmisatom`
9980 Fragment identifiers:
9981 | As specified for `application/xml` in [RFC3023], Section 5.
9982 Base URI:
9983 As specified in [RFC3023], Section 6.
9984 Macintosh File Type code: `TEXT`
9985 Person and email address to contact for further information:
9986 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
9987 Intended usage: `COMMON`
9988 Author/Change controller: `IESG`
9989
9990 Please see section 3.1.1 on why this media type is needed above the Atom Media Type.

9991 **5.1.5 CMIS ACL**

9992 A CMIS ACL Document, when serialized as XML 1.0, can be identified with the following media type:

9993
9994 MIME media type name: `application`
9995 MIME subtype name: `cmisacl+xml`
9996 Mandatory parameters: `None`.
9997 Optional parameters:
9998 | `charset`: This parameter has semantics identical to the `charset` parameter of the `application/xml`
9999 media type as specified in [RFC3023].
10000 Encoding considerations:
10001 | Identical to those of `application/xml` as described in [RFC3023], Section 3.2.
10002 Security considerations: As defined in this specification.
10003 | In addition, as this media type uses the `+xml` convention, it shares the same security considerations as
10004 described in [RFC3023], Section 10.
10005 Interoperability considerations:
10006 There are no known interoperability issues.
10007 Published specification: This specification.
10008 Applications that use this media type:
10009 No known applications currently use this media type.
10010 Additional information:
10011 Magic number(s):
10012 | As specified for `application/xml` in [RFC3023], Section 3.2.
10013 File extension: `.cmisacl`
10014 Fragment identifiers:
10015 | As specified for `application/xml` in [RFC3023], Section 5.
10016 Base URI:
10017 As specified in [RFC3023], Section 6.
10018 Macintosh File Type code: `TEXT`

10019 Person and email address to contact for further information:
10020 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
10021 Intended usage: COMMON
10022 Author/Change controller: IESG
10023

6 Conformance

10024

10025 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level
10026 requirements defined within this specification.

10027 Specification:

10028 This specification references a number of other specifications (see the table above). In order to
10029 comply with this specification, an implementation MUST implement the portions of referenced
10030 specifications necessary to comply with the required provisions of this specification. Additionally,
10031 the implementation of the portions of the referenced specifications that are specifically cited in
10032 this specification MUST comply with the rules for those portions as established in the referenced
10033 specification.

10034

10035 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level
10036 requirements defined within this specification.

10037

10038

10039 Domain Model:

10040 Normative text within this specification takes precedence over the CMIS Core XML Schema.

10041 That is, the normative text in this specification further constrains the schemas and/or WSDL that
10042 are part of this specification; and this specification contains further constraints on the elements
10043 defined in referenced schemas.

10044

10045 Clients:

10046 Client implementations MAY implement either Restful AtomPub Binding or the Web
10047 Services Binding.

10048

10049 Repositories:

10050 Repositories MUST implement the following CMIS protocol bindings:

10051 [❏i.](#) Restful AtomPub Binding

10052 [❏ii.](#) Web Services Binding

10053

10054 Rest Binding:

10055 This specification references a number of other specifications. In order to comply with this
10056 specification, an implementation MUST implement the portions of referenced specifications
10057 necessary to comply with the required provisions of this specification. Additionally, the
10058 implementation of the portions of the referenced specifications that are specifically cited in this
10059 specification MUST comply with the rules for those portions as established in the referenced
10060 specification.

10061 Additionally normative text within this specification takes precedence over the CMIS RestAtom
10062 XML Schema. That is, the normative text in this specification further constrains the schemas
10063 and/or WSDL that are part of this specification; and this specification contains further constraints
10064 on the elements defined in referenced schemas.

10065 The CMIS RestAtom XML takes precedence over any examples or non-normative outlines
10066 included either in this document or as standalone examples.

10067

10068 Web Services Binding:

10069 Normative text within this specification takes precedence over the CMIS Messaging XML and
10070 CMIS WSDL. That is, the normative text in this specification further constrains the schemas and
10071 WSDL that are part of this specification; and this specification contains further constraints on the
10072 elements defined in referenced schemas.

10073 The CMIS Messaging XML and CMIS WSDL takes precedence over any examples or non-
10074 normative outlines included either in this document or as standalone examples.

10075

A. Acknowledgements

10076 The following individuals have participated in the creation of this specification and are gratefully
10077 acknowledged:

10078

10079 **Participants:**

10080 Philippe Allart, Adullact
10081 Florian Bartels, fme AG
10082 Fred Boiscuvier, Exalead, Inc.
10083 Al Brown, IBM
10084 Jay Brown, IBM
10085 Mark Carlson, Sun Microsystems
10086 Derek Carr, IBM
10087 David Caruana, Alfresco Software
10088 Eric Chan, Oracle Corporation
10089 Sameer Charles, Magnolia International AG
10090 Derek Chow, Genus Technologies, LLC
10091 David Choy, EMC Corporation
10092 Scott Conroy, Individual
10093 Cornelia Davis, EMC Corporation
10094 Doug Domeny, Ektron
10095 Kevin Dorr, Flatirons Solutions Corporation
10096 Jason Dubreuil, Fidelity Investments
10097 Michael Duerig, Day Software
10098 Randy Dufault, Genus Technologies, LLC
10099 Will Ezell, dotCMS
10100 Betsy Fanning, AIIM
10101 Steffen Frederiksen, Content Technologies ApS
10102 Stephan Friedl, Quark
10103 Dustin Friesenhahn, Microsoft Corporation
10104 Gary Gershon, Individual
10105 Paul Goetz, SAP AG
10106 Jens Goldhammer, fme AG
10107 Gregory Grefenstette, Exalead, Inc.
10108 Florent Guillaume, Nuxeo
10109 Ethan Gur-esh, Microsoft Corporation
10110 Alexander Haag, WeWebU Software AG
10111 Dennis Hamilton, Individual
10112 Martin Hermes, SAP AG
10113 Jens Huebel, Open Text Corporation
10114 David Izatt, Structured Software Systems Limited (3SL)
10115 Gershon Janssen, Individual
10116 Raphael Jean, Entropysoft
10117 Volker John, Saperion AG
10118 Shane Johnson, Citytech, Inc.
10119 Christophe Kijewska, Adullact
10120 Ijonas Kisselbach, Vamosa
10121 Mark Klamerus, Individual
10122 Stephan Klevenz, SAP AG
10123 Boris Kraft, Magnolia International AG
10124 Alison Macmillan, Oracle Corporation
10125 Michael Marth, Day Software
10126 Mary McRae, OASIS
10127 Ryan McVeigh, Oracle Corporation

10128 Juerg Meier, fme AG
10129 Gregory Melahn, IBM
10130 Pat Miller, Microsoft Corporation
10131 Florian Müller, Open Text Corporation
10132 Thomas Mueller, Day Software
10133 John Newton, Alfresco Software
10134 David Nuescheler, Day Software
10135 Conleth O'Connell, Vignette Corporation
10136 Marc Pallot, ESoCE-NET
10137 Rainer Pausch, WeWebU Software AG
10138 Dominique Pfister, Day Software
10139 Peeter Piegaze, Day Software
10140 David Pitfield, Oracle Corporation
10141 Thomas Pole, Harris Corp
10142 Norrie Quinn, EMC Corporation
10143 Craig Randall, Adobe Corporation
10144 Julian Reschke, Greenbytes GmbH
10145 Celso Rodriguez, ASG Software Solutions
10146 Steve Roth, Oracle Corporation
10147 Patrick Ryan, IBM
10148 Angela Schreiber, Day Software
10149 Spencer Shearer, Exalead, Inc.
10150 Madi Solomon, Pearson PLC
10151 Wojciech Specht, fme AG
10152 Dmitri Tcherevik, FatWire
10153 Jason Tesser, dotCMS
10154 David Torres, dotCMS
10155 Maik Uhlenberg, fme AG
10156 Oliver Walthard, Day Software
10157 Patrick Ward, Booz Allen Hamilton

10158
10159 Original Authors of the initial contribution:

10160 Al Brown, IBM
10161 David Choy, EMC
10162 Cornelia Davis, EMC
10163 Ethan Gur-Esh, Microsoft

10164
10165 Original Acknowledgements of the initial contribution:

10166 Al Brown, IBM
10167 David Caruana, Alfresco
10168 Derek Carr, IBM
10169 David Choy, EMC
10170 Cornelia Davis, EMC
10171 Paul Goetz, SAP
10172 Ethan Gur-Esh, Microsoft
10173 Martin Hermes, SAP
10174 Jens Hubel, OpenText
10175 Jay Brown, IBM
10176 Ryan McVeigh, Oracle
10177 Gregory Melahn, IBM
10178 Florian Müller, OpenText
10179 John Newton, Alfresco
10180 Norrie Quinn, EMC
10181 Steve Roth, Oracle
10182 Craig Randall, EMC

10183

B. Non-Normative Text

10184

10185

C. Revision History

10186

Revision	Date	Editor	Changes Made
1.0	01/11/2010	Al Brown	First specification

10187