

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	16
2.1.3.1	Object-Type Hierarchy and Inheritance	17 <u>1746</u>
2.1.3.2	Object-Type Attributes	17
2.1.3.3	Object-Type Property Definitions	19
2.1.4	Document Object.....	23
2.1.4.1	Content Stream.....	24
2.1.4.2	Renditions.....	24
2.1.4.3	Document Object-Type Definition	26 <u>2625</u>
2.1.5	Folder Object	34 <u>3433</u>
2.1.5.1	File-able Objects	34 <u>3433</u>
2.1.5.2	Folder Hierarchy	35 <u>3534</u>
2.1.5.3	Paths.....	36 <u>3635</u>
2.1.5.4	Folder Object-Type Definition	37 <u>3736</u>
2.1.6	Relationship Object	41 <u>4140</u>
2.1.6.1	Relationship Object-Type Definition.....	42 <u>4241</u>
2.1.7	Policy Object.....	47 <u>4746</u>
2.1.7.1	Policy Object-Type Definition	47
2.1.8	Access Control	51 <u>5150</u>
2.1.8.1	ACL, ACE, Principal, and Permission	51 <u>5150</u>
2.1.8.2	CMIS Permissions	51 <u>5150</u>
2.1.8.3	ACL Capabilities	52 <u>5251</u>
2.1.9	Versioning.....	61 <u>6160</u>
2.1.9.1	Version Series.....	61 <u>6160</u>
2.1.9.2	Latest Version.....	61 <u>6160</u>
2.1.9.3	Major Versions	61 <u>6160</u>
2.1.9.4	Services that modify Version Series	62 <u>6261</u>
2.1.9.5	Versioning Properties on Document Objects	63 <u>6362</u>
2.1.9.6	Document Creation and Initial Versioning State	64 <u>6463</u>
2.1.9.7	Version Specific/Independent membership in Folders	64 <u>6463</u>
2.1.9.8	Version Specific/Independent membership in Relationships	64 <u>6463</u>
2.1.9.9	Versioning visibility in Query Services	65 <u>6564</u>
2.1.10	Query.....	65 <u>6564</u>
2.1.10.1	Relational View Projection of the CMIS Data Model	66 <u>6665</u>
2.1.10.2	Query Language Definition	67 <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

36 2 Domain Model

37 2.1 Data Model

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

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

55 2.1.1 Repository

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

58 2.1.1.1 Optional Capabilities

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

66

67 **Navigation Capabilities:**

68 **capabilityGetDescendants**

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

71 [See section: 2.2.3.2 getDescendants](#)

72

73 **capabilityGetFolderTree**

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

75 [See section: 2.2.3.3 getFolderTree](#)

76

120 **Object Capabilities:**

121 **capabilityContentStreamUpdatability** (enumCapabilityContentStreamUpdates)

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

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

126 **pwconly**: The content stream may be updated only when checked out. The abbreviation PWC is
127 described in section 0

128 Versioning

- 129 • Versioning.

130 See Section: 2.1.4.1 Content StreamContent Stream

131

132 **capabilityChanges** (enumCapabilityChanges)

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

- 135 • **none**: The repository does not support the change log feature.
- 136 • **objectidonly**: The change log can return only the ObjectIDs for changed objects in
137 the repository and an indication of the type of change, not details of the actual change.
- 138 • **properties**: The change log can return properties and the ObjectID for the changed
139 objects
- 140 • **all**: The change log can return the ObjectIDs for changed objects in the repository and
141 more information about the actual change

142 See Section: 2.1.11 Change Log

143

144 **capabilityRenditions** (enumCapabilityRendition)

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

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

148

149 **Filing Capabilities:**

150 **capabilityMultifiling**

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

152 See Section: 2.1.5 Folder Object

153

154 **capabilityUnfiling**

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

156 See Section: 2.1.5 Folder Object

157

158 **capabilityVersionSpecificFiling**

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

160 See Section: 0

161 Versioning

162 Versioning

Formatted
0.21", Bull
+ Tab after

Formatted
0.21", Bull
+ Tab after

Formatted
Formatted

Formatted
0.21", Bull
+ Indent a

Formatted
0.21", Bull
+ Indent a

206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248

Versioning Capabilities:

capabilityPWCUpdatable

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

See Section: 0

[Versioning](#)

[Versioning](#)

capabilityPWCSearchable

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

See Section: 0

[Versioning](#)

[Versioning](#)

capabilityAllVersionsSearchable

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

See Section: 0

[Versioning](#)

[Versioning](#)

Query Capabilities:

capabilityQuery (enumCapabilityQuery)

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

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

See Section: 2.1.10 Query

capabilityJoin (enumCapabilityJoin)

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

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

Formatted
0.21", Bul
+ Tab afte

Formatted
0.21", Bul
+ Tab afte

- **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.

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

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

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

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

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

349 2.1.2.1 Property

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

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

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

361
362 A multi-valued property is either set or not set in its entirety. An individual value of a multi-valued property
363 MUST NOT be in an individual "value not set" state and hold a position in the list of values. An empty list
364 of values MUST NOT be allowed.

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

- 369 • string (xsd:string)
- 370 • boolean (xsd:boolean)
- 371 • decimal (see section 2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions)
- 372 • integer (xsd:integer)
- 373 • datetime (xsd:dateTime and see section 2.1.3.3.5 Attributes specific to Decimal Object-Type
- 374 Property Definitions)
- 375 • uri (xsd:anyURI)

376

418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438

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.

Formatted
0 pt, After

Formatted
Space After

Formatted

Formatted

Formatted

Formatted

500 2.1.3 Object-Type

501 | An **Object-Type** defines a fixed and non-hierarchical set of properties (“_schema_”) that all objects of that
502 | type have. This schema is used by a repository to validate objects and enforce constraints, and is also
503 | used by a user to compose object-type-based (structured) queries.

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

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

508 | A CMIS repository MUST expose exactly one collection of Object-Types via the “_Repository_” services
509 | (*getTypeChildren*, *getTypeDescendants*, *getTypeDefinition*).

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

515 2.1.3.1 Object-Type Hierarchy and Inheritance

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

- 517 | • A CMIS repository MUST have these base types:
 - 518 | ○ *cmis:document* object-type
 - 519 | ○ *cmis:folder* object-type
- 520 | • A CMIS repository MAY have these base types:
 - 521 | ○ *cmis:relationship* object-type
 - 522 | ○ *cmis:policy* object-type
- 523 | • Additional base types MUST NOT exist. Additional object-types MAY be defined as sub-types or
524 | descendant types of these four base types.
- 525 | • A **Base Type** does not have a parent type.
- 526 | • A non-base type has one and only one parent type. An object-type’s **Parent Type** is a part of the
527 | object-type definition.
- 528 | • An object-type definition includes a set of **object-type attributes** (e.g. Fileable, Queryable, etc.)
529 | and a property schema that will apply to Objects of that type.
 - 530 | ○ There is no inheritance of object-type attributes from a parent object-type to its sub-types.
- 531 | • The properties of a CMIS base type MUST be inherited by its descendant types.
- 532 | • A **Child Type** whose immediate parent is NOT its base type SHOULD inherit all the property
533 | definitions that are specified for its parent type. In addition, it MAY have its own property
534 | definitions.
 - 535 | ○ If a property is NOT inherited by a subtype, the exhibited behavior for query MUST be as if
536 | the value of this property is “_not set_” for all objects of this sub-type.
- 537 | • The scope of a query on a given object-type is automatically expanded to include all the
538 | **Descendant Types** of the given object-type with the attribute `includedInSuperTypeQuery`
539 | equals TRUE. This was added for synthetic types as well as to support different type hierarchies
540 | that are not necessarily the same as CMIS. Only the properties of the given object-type,

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

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

546 2.1.3.2 Object-Type Attributes

547 2.1.3.2.1 Attributes common to ALL Object-Type Definitions

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

- | | | |
|-----|--|------------------------------|
| 549 | id | ID |
| 550 | This opaque attribute identifies this object-type in the repository. | |
| 551 | | |
| 552 | localName | String (optional) |
| 553 | This attribute represents the underlying repository's name for the object-type. This field is | |
| 554 | opaque and has no uniqueness constraint imposed by this specification. | |
| 555 | Two properties with the same localName and localNamespace MUST have the same semantic | |
| 556 | equality. | |
| 557 | | |
| 558 | localNamespace | String (optional) |
| 559 | This attribute allows repositories to represent the internal namespace of the underlying | |
| 560 | repository's name for the object-type. | |
| 561 | | |
| 562 | queryName | String |
| 563 | Used for query and filter operations on object-types. This is an opaque String with limitations. | |
| 564 | This string SHOULD NOT contain any characters that negatively interact with the BNF grammar. | |
| 565 | | |
| 566 | The string MUST NOT contain: | |
| 567 | • whitespace <code>" "</code> , | |
| 568 | • comma <code>","</code> | |
| 569 | • double quotes <code>"\""</code> | |
| 570 | • single quotes <code>"'\""</code> | |
| 571 | • backslash <code>"\""</code> | |
| 572 | • the period <code>","</code> character or, | |
| 573 | • the open <code>"("</code> or close <code>")"</code> parenthesis characters. | |
| 574 | | |
| 575 | displayName | String (optional) |
| 576 | Used for presentation by application. | |
| 577 | | |
| 578 | baseId | Enum |
| 579 | A value that indicates whether the base type for this Object-Type is the Document, Folder, | |
| 580 | Relationship, or Policy base type. | |
| 581 | | |
| 582 | parentId | ID |

583 | The ID of the Object-Type's immediate parent type.
584 | It MUST be `"not set"` for a base type.
585 |
586 | **description** String (optional)
587 | Description of this object-type, such as the nature of content, or its intended use. Used for
588 | presentation by application.
589 |
590 | **creatable** Boolean
591 | Indicates whether new objects of this type MAY be created. If the value of this attribute is FALSE,
592 | the repository MAY contain objects of this type already, but MUST NOT allow new objects of this
593 | type to be created.
594 |
595 | **fileable** Boolean
596 | Indicates whether or not objects of this type are `file-able`.
597 |
598 | **queryable** Boolean
599 | Indicates whether or not this object-type can appear in the FROM clause of a query statement. A
600 | non-queryable object-type is not visible through the relational view that is used for query, and
601 | CAN NOT appear in the FROM clause of a query statement.
602 |
603 | **controllablePolicy** Boolean
604 | Indicates whether or not objects of this type are controllable via policies. Policy objects can only
605 | be applied to controllablePolicy objects.
606 |
607 | **controllableACL** Boolean
608 | This attribute indicates whether or not objects of this type are controllable by ACL's. Only objects
609 | that are controllableACL can have an ACL.
610 |
611 | **fulltextIndexed** Boolean
612 | Indicates whether objects of this type are indexed for full-text search for querying via the
613 | CONTAINS() query predicate.
614 |
615 | **includedInSupertypeQuery** Boolean
616 | Indicates whether this type and its subtypes appear in a query of this type's ancestor types.
617 | For example: if Invoice is a sub-type of cmis:document, if this is TRUE on Invoice then for a query
618 | on cmis:document, instances of Invoice will be returned if they match.
619 | If this attribute is FALSE, no instances of Invoice will be returned even if they match the query.

620 2.1.3.3 Object-Type Property Definitions

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

668 2.1.3.3.1 Property Types

669 Property types are defined in section 2.1.2.1 Property.

670 2.1.3.3.2 Attributes common to ALL Object-Type Property Definitions

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

672 **id** ID

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

675

676 **localName** String (optional)

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

679

680 **localNamespace** String (optional)

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

683

684 **queryName** String

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

687

688 **displayName** String (optional)

689 Used for presentation by application.

690

691 **description** String (optional)

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

693

694 **propertyType** Enum

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

697

698 **cardinality** Enum

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

700 Values:

- 701 • **single**: Property can have zero or one values (if property is not required), or exactly one
702 value (if property is required)
- 703 • **multi**: Property can have zero or more values (if property is not required), or one or more
704 values (if property is required).

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

708

709 **updatability** Enum

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

- 758 Values:
- 759 • **readonly:** The value of this property MUST NOT ever be set directly by an application. It
 - 760 is a system property that is either maintained or computed by the repository.
 - 761 ○ The value of a readonly property MAY be indirectly modified by other repository
 - 762 interactions (for example, calling `updateProperties()` on an object will change the
 - 763 object's last modified date, even though that property cannot be directly set via an
 - 764 `updateProperties()` service call.)
 - 765 • **readwrite:** The property value can be modified using the `updateProperties` service.
 - 766 4. **whencheckedout:** The property value MUST only be update-able using a `private`
 - 767 `working copy` Document.
 - 768 a. i.e. the update is either made on a `private working copy` object or made using a
 - 769 `check in` service.
 - 770 2. **oncreate:** The property value MUST only be update-able during the Create operation on
 - 771 that Object.

772

773 **inherited** Boolean

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

775 explicitly defined for this object-type when FALSE.

776

777 **required** Boolean

778

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

780 by the application.

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

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

783 MUST provide a value for this property.

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

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

786 exception.

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

788 `required` SHOULD be set to FALSE.

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

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

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

792

793 **queryable** Boolean

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

795 statement.

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

797 to FALSE.

798

799 **orderable** Boolean

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

801 or an ORDERBY parameter.

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

803

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

Formatted
0.21", Bull
+ Tab after
stops: Not

Formatted
Level: 2 +
1.14"

Formatted
Level: 1 +

Formatted
Level: 1 +

Formatted
Level: 2 +

Formatted
Level: 1 +

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

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

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

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

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

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

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

819 o Europe:

820 ▪ England

821 ▪ France

822 ▪ Germany

823 o North America

824 ▪ Canada

825 ▪ USA

826 ▪ Mexico

827 **openChoice** Boolean

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

832

833 **defaultValue** <PropertyType>

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

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

840 The attributes on the default value element are the same as the attributes on the property
841 definition.

842 2.1.3.3.3 Attributes specific to Integer Object-Type Property Definitions

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

847 **minValue** Integer

848 The minimum value allowed for this property.

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

894

895 **maxValue** Integer

896 The maximum value allowed for this property.

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

899

900 **2.1.3.3.4 Attributes specific to DateTime Object-Type Property Definitions**

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

905 **resolution** String Enumeration

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

- 907 • Year: Year resolution is persisted
- 908 • Date: Date resolution is persisted
- 909 • Time: Time resolution is persisted

910

911 **2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions**

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

916 **precision** Integer Enumeration

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

- 918 • 32: 32-bit precision (`"single"` as specified in IEEE-754-1985).
- 919 • 64: 64-bit precision (`"double"` as specified in IEEE-754-1985).

920

921 **minValue** Decimal

922 The minimum value allowed for this property.

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

925

926 **maxValue** Decimal

927 The maximum value allowed for this property.

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

930 **2.1.3.3.6 Attributes specific to String Object-Type Property Definitions**

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

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

939 **2.1.4 Document Object**

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

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

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

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

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

951 **2.1.4.1 Content Stream**

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

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

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

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

971 **2.1.4.2 Renditions**

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

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

984 **2.1.4.2.1 Rendition Attributes**

985 A rendition has the following attributes:

- 986 **streamId** ID
987 Identifies the rendition stream.
988
- 989 **mimeType** String
990 The MIME type of the rendition stream.
991
- 992 **length** Integer (optional)
993 The length of the rendition stream in bytes.
994
- 995 **title** String (optional)
996 Human readable information about the rendition.
997
- 998 **kind** String
999 A categorization String associated with the rendition.
1000
- 1001 **height** Integer (optional)
1002 Typically used for "image" renditions (expressed as pixels). SHOULD be present if kind =
1003 cmis:thumbnail.
1004
- 1005 **width** Integer (optional)
1006 Typically used for "image" renditions (expressed as pixels). SHOULD be present if kind =
1007 cmis:thumbnail.
1008
- 1009 **renditionDocumentId** ID (optional)
1010 If specified, then the rendition can also be accessed as a document object in the CMIS services.
1011 If not set, then the rendition can only be accessed via the rendition services. Referential integrity
1012 of this ID is repository-specific.

1054 **2.1.4.2.2 Rendition Kind**

1055 A Rendition may be categorized via its `kind`. The repository is responsible for assigning kinds to
1056 Renditions, including custom kinds. A repository kind does not necessarily identify a single Rendition for
1057 a given Object.

1058 CMIS defines the following kind:

- 1059
 - `cmis:thumbnail`: A rendition whose purpose is to provide an image preview of the document
1060 without requiring the client to download the full document content stream. Thumbnails are
1061 generally reduced fidelity representations.

1062 **2.1.4.3 Document Object-Type Definition**

1063 This section describes the definition of the Document Object-Type's attribute values and property
1064 definitions which must be present on Document instance objects. All attributes and property definitions
1065 are listed by their ID.

1066 **2.1.4.3.1 Attributes specific to Document Object-Types**

1067 The following Object **attributes** MUST only apply to Object-Type definitions whose baseId is the
1068 `cmis:document` Object-Type, in addition to the common attributes specified above:

1069 **versionable** Boolean
1070 Indicates whether or not objects of this type are version-able. (See section 0
1071 [Versioning](#)
1072 [Versioning](#).)
1073

1074 **contentStreamAllowed** Enum
1075 A value that indicates whether a content-stream MAY, MUST, or MUST NOT be included in
1076 objects of this type. Values:
1077

- **notallowed**: A content-stream MUST NOT be included
- **allowed**: A content-stream MAY be included
- **required**: A content-stream MUST be included (i.e. MUST be included when the object is
1078 created, and MUST NOT be deleted.)

1079
1080

1081 **2.1.4.3.2 Attribute Values**

1082 The Document Object-Type MUST have the following attribute values.

1083 Notes:

- 1084
 - A value of `<repository-specific>` indicates that the value of the property MAY be set to any valid
1085 value for the attribute type.
 - Unless explicitly stated otherwise, all values specified in the list MUST be followed for the Object-
1086 Type definition.
1087

1088
1089 **id**
1090 Value: `cmis:document`

1091
1092 **localName**
1093 Value: `<repository-specific>`
1094

1095 **localNamespace**
1096 Value: <repository-specific>
1097
1098 **queryName**
1099 Value: cmis:document
1100
1101 **displayName**
1102 Value: <repository-specific>
1103
1104 **baseId**
1105 Value: cmis:document
1106
1107 **parentId**
1108 Value: Not set
1109
1110 **description**
1111 Value: <repository-specific>
1112
1113 **creatable**
1114 Value: <repository-specific>
1115
1116 **fileable**
1117 Value: TRUE
1118
1119 **queryable**
1120 Value: SHOULD be TRUE
1121
1122 **controllablePolicy**
1123 Value: <repository-specific>
1124
1125 **includedInSupertypeQuery**
1126 Value: <repository-specific>
1127
1128 **versionable**
1129 Value: <repository-specific>
1130
1131 **contentStreamAllowed**
1132 Value: <repository-specific>
1133
1134 **controllableACL**
1135 Value: <repository-specific>
1136
1137 **fulltextIndexed**

1138 Value: <repository-specific>

1139 2.1.4.3.3 Property Definitions

1140 The Document base Object-Type MUST have the following property definitions, and MAY include
1141 additional property definitions. Any attributes not specified for the property definition are repository
1142 specific. For all property definitions on base types, the query name MUST be the same as the property
1143 ID. The repository MUST have the following property definitions on the Document Type:

1144		
1145	cmis:name	Name of the object
1146	Inherited:	False
1147	Property Type:	String
1148	Cardinality:	Single
1149		
1150	cmis:objectId	Id of the object
1151	Required:	False
1152	Inherited:	False
1153	Property Type:	ID
1154	Cardinality:	Single
1155	Updatability:	Read Only
1156	Choices:	Not Applicable
1157	Open Choice:	Not Applicable
1158	Repository MUST return this property with non-empty values when an object is requested and the	
1159	property filter does not exclude them	
1160		
1161		
1162	cmis:baseTypeid	Id of the base object-type for the object
1163	Required:	False
1164	Inherited:	False
1165	Property Type:	ID
1166	Cardinality:	Single
1167	Updatability:	Read Only
1168	Choices:	Not Applicable
1169	Open Choice:	Not Applicable
1170	Repository MUST return this property with non-empty values when an object is requested and the	
1171	property filter does not exclude them	
1172		
1173	cmis:objectTypeid	Id of the object's type
1174	Required:	True
1175	Inherited:	False
1176	Property Type:	ID
1177	Cardinality:	Single
1178	Updatability:	oncreate
1179	Choices:	Not Applicable
1180	Open Choice:	Not Applicable

1181 Repository MUST return this property with non-empty values when an object is requested and the
1182 property filter does not exclude them

1183

1184 **cmis:createdBy** User who created the object.

1185 Required: False

1186 Inherited: False

1187 Property Type: String

1188 Cardinality: Single

1189 Updatability: Read Only

1190 Choices: Not Applicable

1191 Open Choice: Not Applicable

1192 Queryable: True

1193 Orderable: True

1194 Repository MUST return this property with non-empty values when an object is requested and the
1195 property filter does not exclude them

1196

1197 **cmis:creationDate** DateTime when the object was created.

1198 Required: False

1199 Inherited: False

1200 Property Type: DateTime

1201 Cardinality: Single

1202 Updatability: Read Only

1203 Choices: Not Applicable

1204 Open Choice: Not Applicable

1205 Queryable: True

1206 Orderable: True

1207 Repository MUST return this property with non-empty values when an object is requested and the
1208 property filter does not exclude them

1209

1210 **cmis:lastModifiedBy** User who last modified the object.

1211 Required: False

1212 Inherited: False

1213 Property Type: String

1214 Cardinality: Single

1215 Updatability: Read Only

1216 Choices: Not Applicable

1217 Open Choice: Not Applicable

1218 Queryable: True

1219 Orderable: True

1220 Repository MUST return this property with non-empty values when an object is requested and the
1221 property filter does not exclude them

1222

1223 **cmis:lastModificationDate** DateTime when the object was last modified.

1224	Required:	False
1225	Inherited:	False
1226	Property Type:	DateTime
1227	Cardinality:	Single
1228	Updatability:	Read Only
1229	Choices:	Not Applicable
1230	Open Choice:	Not Applicable
1231	Queryable:	True
1232	Orderable:	True
1233	Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them	
1234		
1235		

1236 **cmis:changeToken** Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)

1237		
1238	Required:	False
1239	Inherited:	False
1240	Property Type:	String
1241	Cardinality:	Single
1242	Updatability:	Read Only
1243	Choices:	Not Applicable
1244	Open Choice:	Not Applicable

1245 ~~Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them~~ 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.

1250 **cmis:isImmutable** TRUE if the repository MUST throw an error at any attempt to update or delete the object.

1251		
1252	Required:	False
1253	Inherited:	False
1254	Property Type:	Boolean
1255	Cardinality:	Single
1256	Updatability:	Read Only
1257	Choices:	Not Applicable
1258	Open Choice:	Not Applicable

1259 Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them

1262 **cmis:isLatestVersion** See section 0

1263 Versioning

1264 ~~Versioning.~~

1265	Required:	False
1266	Inherited:	False
1267	Property Type:	Boolean

1268 Cardinality: Single
 1269 Updatability: Read Only
 1270 Choices: Not Applicable
 1271 Open Choice: Not Applicable
 1272 Repository MUST return this property with non-empty values when an object is requested and the
 1273 property filter does not exclude them. Version Property Values are repository-specific when a
 1274 document is defined as non-versionable.

1275

1276 **cmis:isMajorVersion** See section 0

1277 Versioning

1278 ~~Versioning.~~

1279 Required: False
 1280 Inherited: False
 1281 Property Type: Boolean
 1282 Cardinality: Single
 1283 Updatability: Read Only
 1284 Choices: Not Applicable
 1285 Open Choice: Not Applicable

1286 Repository MUST return this property with non-empty values when an object is requested and the
 1287 property filter does not exclude them. Version Property Values are repository-specific when a
 1288 document is defined as non-versionable.

1289

1290 **cmis:isLatestMajorVersion** See section 0

1291 Versioning

1292 ~~Versioning.~~

1293 Required: False
 1294 Inherited: False
 1295 Property Type: Boolean
 1296 Cardinality: Single
 1297 Updatability: Read Only
 1298 Choices: Not Applicable
 1299 Open Choice: Not Applicable

1300 Repository MUST return this property with non-empty values when an object is requested and the
 1301 property filter does not exclude them. Version Property Values are repository-specific when a
 1302 document is defined as non-versionable.

1303

1304 **cmis:versionLabel** See section 0

1305 Versioning

1306 ~~Versioning.~~

1307 Required: False
 1308 Inherited: False
 1309 Property Type: String
 1310 Cardinality: Single
 1311 Updatability: Read Only

1312 Choices: Not Applicable
 1313 Open Choice: Not Applicable
 1314 Repository MUST return this property with non-empty values when an object is requested and the
 1315 property filter does not exclude them. Version Property Values are repository-specific when a
 1316 document is defined as non-versionable.

1317
 1318 **cmis:versionSeriesId** See section 0

1319 [Versioning](#)
 1320 ~~Versioning.~~

1321 Required: False
 1322 Inherited: False
 1323 Property Type: ID
 1324 Cardinality: Single
 1325 Updatability: Read Only
 1326 Choices: Not Applicable
 1327 Open Choice: Not Applicable

1328 Repository MUST return this property with non-empty values when an object is requested and the
 1329 property filter does not exclude them. Version Property Values are repository-specific when a
 1330 document is defined as non-versionable.

1331
 1332 **cmis:isVersionSeriesCheckedOut** See section 0

1333 [Versioning](#)
 1334 ~~Versioning.~~

1335 Required: False
 1336 Inherited: False
 1337 Property Type: Boolean
 1338 Cardinality: Single
 1339 Updatability: Read Only
 1340 Choices: Not Applicable
 1341 Open Choice: Not Applicable

1342 Repository MUST return this property with non-empty values when an object is requested and the
 1343 property filter does not exclude them. Version Property Values are repository-specific when a
 1344 document is defined as non-versionable.

1345
 1346 **cmis:versionSeriesCheckedOutBy** See section 0

1347 [Versioning](#)
 1348 ~~Versioning.~~

1349 Required: False
 1350 Inherited: False
 1351 Property Type: String
 1352 Cardinality: Single
 1353 Updatability: Read Only
 1354 Choices: Not Applicable
 1355 Open Choice: Not Applicable

1356 Version Property Values are repository-specific when a document is defined as non-versionable.

1357

1358 **cmis:versionSeriesCheckedOutId** See section 0

1359 [Versioning](#)

1360 ~~Versioning.~~

1361 Required: False

1362 Inherited: False

1363 Property Type: ID

1364 Cardinality: Single

1365 Updatability: Read Only

1366 Choices: Not Applicable

1367 Open Choice: Not Applicable

1368 Version Property Values are repository-specific when a document is defined as non-versionable.

1369

1370 **cmis:checkinComment** See section 0

1371 [Versioning](#)

1372 ~~Versioning.~~

1373 Required: False

1374 Inherited: False

1375 Property Type: String

1376 Cardinality: Single

1377 Updatability: Read Only

1378 Choices: Not Applicable

1379 Open Choice: Not Applicable

1380 Version Property Values are repository-specific when a document is defined as non-versionable.

1381

1382 **cmis:contentStreamLength** Length of the content stream (in bytes).

1383 Required: False

1384 Inherited: False

1385 Property Type: Integer

1386 Cardinality: Single

1387 Updatability: Read Only

1388 Choices: Not Applicable

1389 Open Choice: Not Applicable

1390 Repository MUST return this property with non-empty values when an object is requested and the
1391 property filter does not exclude them and if the document has a content stream

1392

1393 **cmis:contentStreamMimeType** MIME type of the Content Stream

1394 Required: False

1395 Inherited: False

1396 Property Type: String

1397 Cardinality: Single

1398 Updatability: Read Only

1399	Choices:	Not Applicable
1400	Open Choice:	Not Applicable
1401	Repository MUST return this property with non-empty values when an object is requested and the	
1402	property filter does not exclude them and if the document has a content stream	
1403		
1404	cmis:contentStreamFileName	File name of the Content Stream
1405	Required:	False
1406	Inherited:	False
1407	Property Type:	String
1408	Cardinality:	Single
1409	Repository MUST return this property with non-empty values when an object is requested and the	
1410	property filter does not exclude them and if the document has a content stream	
1411		
1412	cmis:contentStreamId	Id of the stream
1413	Required:	False
1414	Inherited:	False
1415	Property Type:	ID
1416	Cardinality:	Single
1417	Updatability:	Read Only
1418	Choices:	Not Applicable
1419	Open Choice:	Not Applicable

1420 2.1.5 Folder Object

1421 A folder object serves as the anchor for a collection of *file-able* objects. The folder object has an *implicit*
1422 hierarchical relationship with each object in its collection, with the anchor folder object being the **Parent**
1423 object and each object in the collection being a **Child** object. This implicit relationship has specific
1424 containment semantics which MUST be maintained by the repository with implicit referential integrity.
1425 (That is, there will never be a dangling parent-relationship or a dangling child-relationship. Furthermore,
1426 object A is a parent of object B if and only if object B is a child of object A.) This system-maintained
1427 implicit relationship is distinct from an *explicit* relationship which is instantiated by an application-
1428 maintained Relationship Object. (See section 2.1.6 Relationship Object.)

1429 A folder object does not have a content-stream and is not version-able. A folder object MAY be
1430 associated with zero or more renditions (see section 2.1.4.2 Renditions).

1431 2.1.5.1 File-able Objects

1432 A *file-able* object is one that MAY be "filed" into a folder. That is, it MAY be a child object of a folder
1433 object. The following list defines whether the base CMIS Object-types are file-able:

1434	cmis:folder
1435	MUST be file-able
1436	
1437	cmis:document
1438	MUST be file-able
1439	
1440	cmis:relationship

1481 MUST NOT be file-able

1482

1483 **cmis:policy**

1484 MAY be file-able

1485 2.1.5.1.1 Document Version Series and Filing

1486 | Since document objects are versionable, a document object's membership in a folder MAY be version-
1487 specific or version-independent. That is, the folder membership MAY be restricted to that particular
1488 version of the document or MAY apply to all versions of the document. Whether or not a repository
1489 supports version-specific filing is discoverable via the `Get Repository Information` service
1490 (`getRepositoryInfo`).

1491 When the child objects of a folder are retrieved, a specific version of a document MAY be returned. If the
1492 repository supports version-specific filing, the specific version filed in that folder is returned. If the
1493 repository does not support version-specific filing, the latest version of the document is returned.

1494 Likewise, this version sensitivity in child-binding also affects the behavior of parent retrieval for a
1495 document object, as well as the scope of the `IN_FOLDER()` and `IN_TREE()` function calls in a query. For
1496 non-versionable fileable objects, their membership in a folder does not have version sensitivity.

1497 2.1.5.1.2 Filing Restrictions by Object-Type

1498 | A folder collection's membership MAY be restricted by object-type. Each folder object has a multi-valued
1499 `AllowedChildObjectTypes` property, which specifies that only objects of these types are allowed to be
1500 its children. If this property is `not set`, then objects of any file-able type MAY be filed in the Folder. It is
1501 repository-specific if subtypes of the types listed in the `AllowedChildObjectTypes` property MAY be filed
1502 in the folder.

1503 Because of these filing constraints, when a new folder object is created, an existing folder object MUST
1504 be specified as its parent.

1505 When a non-file-able object is created, a parent folder MUST NOT be specified.

1506 When a file-able object is deleted, it is removed from any folder collection in which the object is a
1507 member. In other words, when an object is deleted, all implicit parent-child relationships with the deleted
1508 object as a child cease to exist.

1509 2.1.5.2 Folder Hierarchy

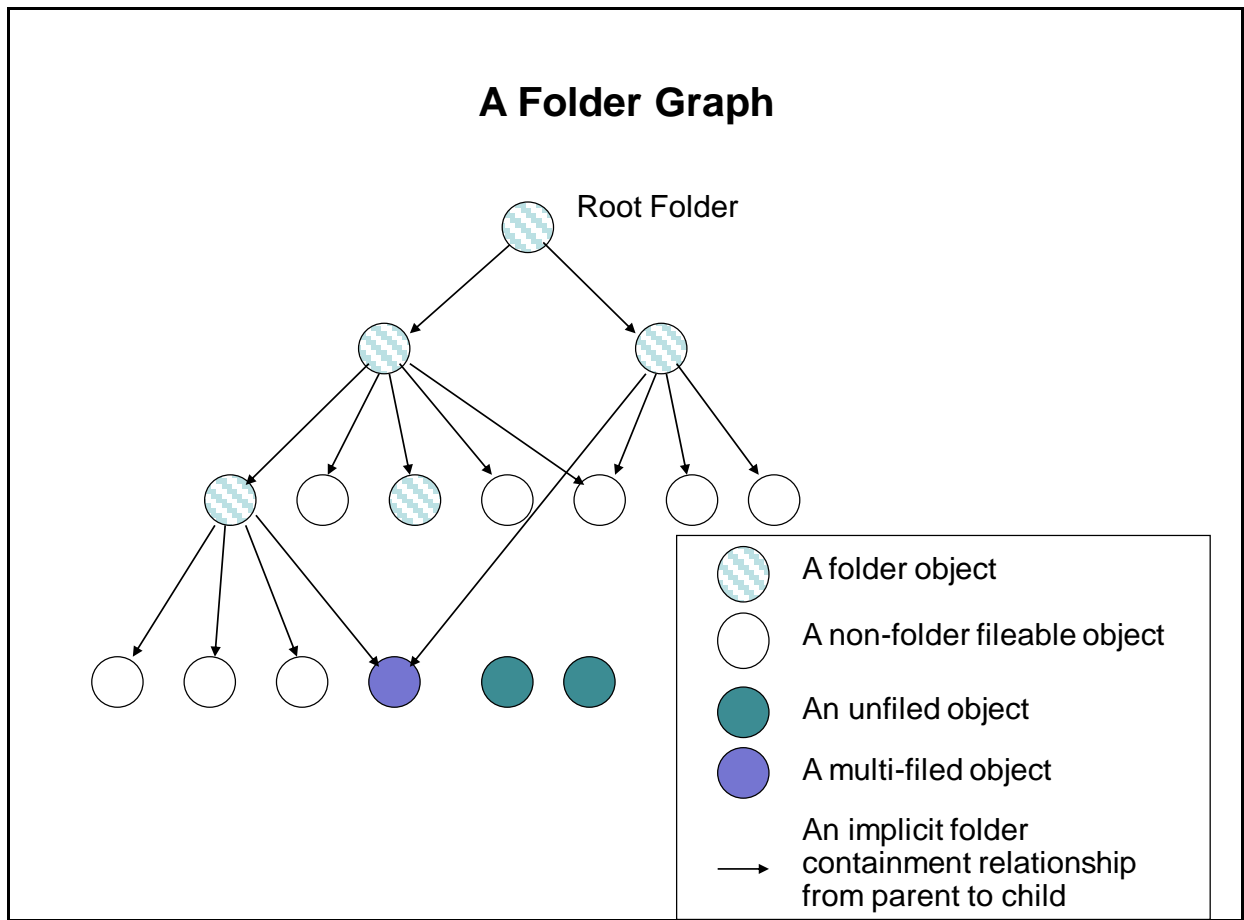
1510 CMIS imposes the following constraints on folder objects:

- 1511 | • Every folder object, except for one which is called the **Root Folder**, MUST have one and only
1512 | one parent folder. The Root Folder does not have a parent.
- 1513 | • A cycle in folder containment relationships is not allowed. That is, a folder object cannot have
1514 | itself as one of its descendant objects.
- 1515 | • A child object that is a folder object can itself be the parent object of other file-able objects.

1516 With these constraints, the folder objects in a CMIS repository necessarily form a strict hierarchy, with the
1517 Root Folder being the root of the hierarchy.

1518 The child objects of a given folder object, their child objects, and grandchild objects, etc., are called
1519 **Descendant** objects of the given folder object. A folder object together with all its descendant objects are
1520 collectively called a **Tree** rooted at that folder object.

1521 A non-folder object does not have any descendant object. Thus, a **Folder Graph** that consists of all
 1522 fileable objects as nodes, and all the implicit folder containment relationships as directed edges from
 1523 parent to child, is a directed acyclic graph, possibly with some disconnected (orphan) nodes. It follows
 1524 that the tree rooted at any given folder object is also a directed acyclic graph, although a non-folder object
 1525 in the tree MAY have ancestors that are not ancestors of the rooted folder.



1526

1527 Folder objects are handled using [the basic CRUD services for objects](#), and the folder graph is traversed
 1528 using the [Navigation Services](#).

1529 The **Root Folder** is a special folder such that it cannot be created, deleted, or moved using CMIS
 1530 services. Otherwise, it behaves like any other folder object.

1531 2.1.5.3 Paths

1532 A folder hierarchy MAY be represented in a canonical notation such as path. For CMIS, a path is
 1533 represented by:

- 1534 | • ["/](#) for the root folder
- 1535 | • All paths start with the root folder.
- 1536 | • A set of the folder and object path segments separated by ["/](#) in order of closest to the root.
- 1537 | • Folder and object path segments are specified by pathSegment tokens which can be retrieved by
 1538 | all services that take an `includePathSegments` parameter.
- 1539 | • A pathSegment token MUST not include a ["/](#) character.

1583 ○ It is repository specific how a repository chooses the value for pathSegment.
1584 Repositories might choose to use cmis:name or content stream filename for
1585 pathSegment token.

1586 • The pathSegment token for each item MUST uniquely identify the item in the folder.

1587

1588 E.g., if folder A is under the root, and folder B is under A, then the path would be /A/B.

1589 A path for an object may be calculated by taking the item's parent folder cmis:path property and
1590 appending the "/" character and the object's pathSegment. This constructed path may be given as
1591 input to the *getObjectByPath* service for object by path retrieval.

1592 The *getObjectParents* service returns *relativePathSegment* tokens. These tokens are the
1593 pathSegment of the input object relative to the parent folders.

1594 2.1.5.4 Folder Object-Type Definition

1595 This section describes the definition of the Folder Object-Type's attribute values and property definitions
1596 which must be present on Folder instance objects. All attributes and property definitions are listed by
1597 their ID.

1598 2.1.5.4.1 Attribute Values

1599 The Folder Object-Type MUST have the following attribute values.

1600 Notes:

- 1601 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid
1602 value for the attribute type.
- 1603 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1604 Object-Type definition.

1605

1606 **id**

Value: cmis:folder

1608

1609 **localName**

Value: <repository-specific>

1611

1612 **localNamespace**

Value: <repository-specific>

1614

1615 **queryName**

Value: cmis:folder

1617

1618 **displayName**

Value: <repository-specific>

1620

1621 **baseId**

Value: cmis:folder

1623

1624 **parentId**

Value: Not set

1625

Formatted
0.24", Bul
+ Tab afte
stops: Not

1626
 1627 **description**
 1628 Value: <repository-specific>
 1629
 1630 **creatable**
 1631 Value: <repository-specific>
 1632
 1633 **fileable**
 1634 Value: TRUE
 1635
 1636 **queryable**
 1637 Value: SHOULD be TRUE
 1638
 1639 **controllablePolicy**
 1640 Value: <repository-specific>
 1641
 1642 **includedInSupertypeQuery**
 1643 Value: <repository-specific>
 1644
 1645 **controllableACL**
 1646 Value: <repository-specific>
 1647
 1648 **fulltextIndexed**
 1649 Value: <repository-specific>
 1650

1651 **2.1.5.4.2 Property Definitions**

1652 The Folder base Object-Type MUST have the following property definitions, and MAY include additional
 1653 property definitions. Any attributes not specified for the Property Definition are repository specific. For all
 1654 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 Folder Type:

1656	cmis:name	Name of the object
1657	Inherited:	False
1658	Property Type:	String
1659	Cardinality:	Single
1660	Required:	True
1661		
1662	cmis:objectId	Id of the object
1663	Required:	False
1664	Inherited:	False
1665	Property Type:	ID
1666	Cardinality:	Single
1667	Updatability:	Read Only
1668	Choices:	Not Applicable

1669	Open Choice:	Not Applicable
1670	Repository MUST return this property with non-empty values when an object is requested and the	
1671	property filter does not exclude them	
1672		
1673	cmis:baseTypeId	Id of the base object-type for the object
1674	Required:	False
1675	Inherited:	False
1676	Property Type:	ID
1677	Cardinality:	Single
1678	Updatability:	Read Only
1679	Choices:	Not Applicable
1680	Open Choice:	Not Applicable
1681	Repository MUST return this property with non-empty values when an object is requested and the	
1682	property filter does not exclude them	
1683		
1684	cmis:objectTypeId	Id of the object's type
1685	Required:	False True
1686	Inherited:	False
1687	Property Type:	ID
1688	Cardinality:	Single
1689	Updatability:	oncreate
1690	Choices:	Not Applicable
1691	Open Choice:	Not Applicable
1692	Repository MUST return this property with non-empty values when an object is requested and the	
1693	property filter does not exclude them	
1694		
1695	cmis:createdBy	User who created the object.
1696	Required:	False
1697	Inherited:	False
1698	Property Type:	String
1699	Cardinality:	Single
1700	Updatability:	Read Only
1701	Choices:	Not Applicable
1702	Open Choice:	Not Applicable
1703	Queryable:	True
1704	Orderable:	True
1705	Repository MUST return this property with non-empty values when an object is requested and the	
1706	property filter does not exclude them	
1707		
1708	cmis:creationDate	DateTime when the object was created.
1709	Required:	False
1710	Inherited:	False
1711	Property Type:	DateTime

1712	Cardinality:	Single
1713	Updatability:	Read Only
1714	Choices:	Not Applicable
1715	Open Choice:	Not Applicable
1716	Queryable:	True
1717	Orderable:	True
1718	Repository MUST return this property with non-empty values when an object is requested and the	
1719	property filter does not exclude them	
1720		
1721	cmis:lastModifiedBy	User who last modified the object.
1722	Required:	False
1723	Inherited:	False
1724	Property Type:	String
1725	Cardinality:	Single
1726	Updatability:	Read Only
1727	Choices:	Not Applicable
1728	Open Choice:	Not Applicable
1729	Queryable:	True
1730	Orderable:	True
1731	Repository MUST return this property with non-empty values when an object is requested and the	
1732	property filter does not exclude them	
1733		
1734	cmis:lastModificationDate	DateTime when the object was last modified.
1735	Required:	False
1736	Inherited:	False
1737	Property Type:	DateTime
1738	Cardinality:	Single
1739	Updatability:	Read Only
1740	Choices:	Not Applicable
1741	Open Choice:	Not Applicable
1742	Queryable:	True
1743	Orderable:	True
1744	MUST be set on the object	
1745		
1746	cmis:changeToken	Token used for optimistic locking & concurrency checking.
1747	(see section 2.2.1.3 Change Tokens)	
1748	Required:	False
1749	Inherited:	False
1750	Property Type:	String
1751	Cardinality:	Single
1752	Updatability:	Read Only
1753	Choices:	Not Applicable
1754	Open Choice:	Not Applicable

1755 | ~~Repository MUST return this property with non-empty values when an object is requested and the~~
 1756 | ~~property filter does not exclude them~~Repository MUST return this property with non-empty values
 1757 | ~~when an object is requested and the property filter does not exclude them. If the repository does~~
 1758 | ~~not support change tokens, this property SHOULD not be set.~~

1759 |
 1760 | **cmis:parentId** ID of the parent folder of the folder.
 1761 | Required: False
 1762 | Inherited: False
 1763 | Property Type: ID
 1764 | Cardinality: Single
 1765 | Updatability: Read Only
 1766 | Choices: Not Applicable
 1767 | Open Choice: Not Applicable
 1768 | Repository MUST return this property with non-empty values when an object is requested and the
 1769 | property filter does not exclude them

1770 |
 1771 | **cmis:path** The fully qualified path to this folder. See section 2.1.5.3
 1772 | Paths.
 1773 | Required: False
 1774 | Inherited: False
 1775 | Property Type: String
 1776 | Cardinality: Single
 1777 | Updatability: Read Only
 1778 | Choices: Not Applicable
 1779 | Open Choice: Not Applicable
 1780 | Repository MUST return this property with non-empty values when an object is requested and the
 1781 | property filter does not exclude them

1782 |
 1783 | **cmis:allowedChildObjectTypes** Id's of the set of Object-types that can be created, moved or
 1784 | filed into this folder.
 1785 | Required: False
 1786 | Inherited: False
 1787 | Property Type: ID
 1788 | Cardinality: Multi
 1789 | Updatability: Read Only
 1790 | Choices: Not Applicable
 1791 | Open Choice: Not Applicable

1792 | 2.1.6 Relationship Object

1793 | A relationship object is semantically a *dependent* object. A relationship object MUST NOT have a
 1794 | content-stream, and MUST NOT be versionable, MAY be queryable, and MUST NOT be fileable,
 1795 | although it MAY be controllable.

1796 | If a repository does not support relationship objects, the relationship base object-type SHOULD NOT be
 1797 | returned by a `“_Get Types”` service call.

1798 A **Relationship Object** instantiates an explicit, binary, directional, non-invasive, and typed relationship
1799 between a **Source Object** and a **Target Object**. The source object and the target object MUST both be
1800 independent objects, such as a document object, a folder object, or a policy object. Whether a policy
1801 object is allowed to be the source or target object of a relationship object is repository-specific.

1802 The relationship instantiated by a relationship object is *explicit* since it is explicitly represented by an
1803 object and is explicitly managed by application.

1804 This relationship is *non-invasive* in the sense that creating or removing this relationship SHOULD NOT
1805 modify either the source or the target object. That is, it SHOULD NOT require an update capability (or
1806 permission) on either object; SHOULD NOT affect the versioning state of either object; and SHOULD
1807 NOT change their `"Last Modification Date"`.

1808 Explicit relationships can be used to create an arbitrary relationship graph among independent objects.
1809 Such a relationship graph is only structural in nature. No inheritance or transitive properties are attached
1810 to a relationship graph.

1811 The notion of a source object and a target object of a relationship is used solely to indicate the direction of
1812 the relationship. No semantics or implementation bias is implied by this terminology.

1813 The binding of a relationship object to a source document object or to a target document object MAY be
1814 either version-specific or version-independent. This version sensitivity is repository-specific, and is largely
1815 transparent to CMIS. An independent object MAY participate in any number of explicit relationships, as
1816 the source object for some and as the target object for others. Multiple relationships MAY exist between
1817 the same pair of source and target objects.

1818 Referential integrity, either between the source object and the target object, or between the relationship
1819 object and the source or target object, is repository-specific. Therefore, creating an explicit relationship
1820 between two objects MAY impose a constraint on any of the three objects, and removing a relationship or
1821 deleting either the source or the target object MAY be restricted by such a constraint. If the source or the
1822 target object of a relationship is deleted, the repository MAY automatically delete the relationship object.

1823 Like all CMIS objects, relationship objects are typed. Typing relationship allows them to be grouped,
1824 identified, and traversed by type id, and for properties to be defined for individual relationship types.

1825 Additionally, a relationship object-type MAY specify that only Objects of a specific Object-Type can
1826 participate as the source object or target object for relationship objects of that type. If no such constraints
1827 are specified, then an independent object of any type MAY be the source or the target of a relationship
1828 object of that type.

1829 When a relationship object is created, the source object ID and the target object ID MUST reference valid
1830 non-relationship CMIS objects.

1831 When a relationship object is retrieved, its source object or target object MAY no longer exist, since
1832 referential integrity MAY not be maintained by a repository.

1833 In addition to object CRUD services, a `"Get Relationships"` service (*getObjectRelationships*) may be
1834 used to return a set of relationship objects in which a given independent object is identified as the source
1835 or the target object, according to the binding semantics maintained by the repository (i.e., either a
1836 version-specific or a version-independent binding as described above).

1837 2.1.6.1 Relationship Object-Type Definition

1838 This section describes the definition of the Relationship Object-Type's attribute values and property
1839 definitions which must be present on Relationship instance objects. All attributes and property definitions
1840 are listed by their ID.

1884 2.1.6.1.1 Attributes specific to Relationship Object-Types

1885 The following Object **attributes** MUST only apply to Object-Type definitions whose baseId is the
1886 cmis:relationship Object-Type, in addition to the common attributes specified above:

1887 **allowedSourceTypes** ID (multi-valued)

1888 A list of object-type IDs, indicating that the source object of a relationship object of this type
1889 MUST only be one of the types listed.

1890 If this attribute is `"not set"`, then the source object MAY be of any type.

1891

1892 **allowedTargetTypes** ID (multi-valued)

1893 A list of object-type IDs, indicating that the target object of a relationship object of this type MUST
1894 only be one of the types listed.

1895 If this attribute is `"not set"`, then the target object MAY be of any type.

1896 2.1.6.1.2 Attribute Values

1897 The Relationship Object-Type MUST have the following attribute values.

1898 Notes:

- 1899 • A value of `<repository-specific>` indicates that the value of the property MAY be set to any
1900 valid value for the attribute type.
- 1901 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1902 Object-Type definition.

1903

1904 **id**

1905 Value: cmis:relationship

1906

1907 **localName**

1908 Value: `<repository-specific>`

1909

1910 **localNamespace**

1911 Value: `<repository-specific>`

1912

1913 **queryName**

1914 Value: cmis:relationship

1915

1916 **displayName**

1917 Value: `<repository-specific>`

1918

1919 **baseId**

1920 Value: cmis:relationship

1921

1922 **parentId**

1923 Value: Not set

1924

1925 **description**

1926 Value: `<repository-specific>`

Formatted
0.24", Bul
Tab after:

1927

1928 **creatable**

1929 Value: <repository-specific>

1930

1931 **fileable**

1932 Value: FALSE

1933

1934 **queryable**

1935 Value: <repository-specific>

1936

1937 **includedInSupertypeQuery**

1938 Value: <repository-specific>

1939

1940 **controllablePolicy**

1941 Value: <repository-specific>

1942

1943 **allowedSourceTypes**

1944 Value: <repository-specific>

1945

1946 **allowedTargetTypes**

1947 Value: <repository-specific>

1948

1949 **controllableACL**

1950 Value: <repository-specific>

1951

1952 **fulltextIndexed**

1953 Value: <repository-specific>

1954

1955 **2.1.6.1.3 Property Definitions**

1956 The Relationship base Object-Type MUST have the following property definitions, and MAY include
 1957 additional property definitions. Any attributes not specified by the Property Definitions are repository
 1958 specific. For all property definitions on base types, the query name MUST be the same as the property
 1959 ID. The repository MUST have the following property definitions on the Relationship Type:

1960

1961	cmis:name	Name of the object
1962	Inherited:	False
1963	Property Type:	String
1964	Cardinality:	Single
1965		
1966	cmis:objectId	Id of the object
1967	Required:	False
1968	Inherited:	False
1969	Property Type:	ID

1970	Cardinality:	Single
1971	Updatability:	Read Only
1972	Choices:	Not Applicable
1973	Open Choice:	Not Applicable
1974	Repository MUST return this property with non-empty values when an object is requested and the	
1975	property filter does not exclude them	
1976		
1977	cmis:baseTypeId	Id of the base object-type for the object
1978	Required:	False
1979	Inherited:	False
1980	Property Type:	ID
1981	Cardinality:	Single
1982	Updatability:	Read Only
1983	Choices:	Not Applicable
1984	Open Choice:	Not Applicable
1985	Repository MUST return this property with non-empty values when an object is requested and the	
1986	property filter does not exclude them	
1987		
1988	cmis:objectTypeId	Id of the object's type
1989	Required:	False True
1990	Inherited:	False
1991	Property Type:	ID
1992	Cardinality:	Single
1993	Updatability:	oncreate
1994	Choices:	Not Applicable
1995	Open Choice:	Not Applicable
1996	Repository MUST return this property with non-empty values when an object is requested and the	
1997	property filter does not exclude them	
1998		
1999	cmis:createdBy	User who created the object.
2000	Required:	False
2001	Inherited:	False
2002	Property Type:	String
2003	Cardinality:	Single
2004	Updatability:	Read Only
2005	Choices:	Not Applicable
2006	Open Choice:	Not Applicable
2007	Repository MUST return this property with non-empty values when an object is requested and the	
2008	property filter does not exclude them	
2009		
2010	cmis:creationDate	DateTime when the object was created.
2011	Required:	False
2012	Inherited:	False

2013	Property Type:	DateTime
2014	Cardinality:	Single
2015	Updatability:	Read Only
2016	Choices:	Not Applicable
2017	Open Choice:	Not Applicable
2018	Repository MUST return this property with non-empty values when an object is requested and the	
2019	property filter does not exclude them	
2020		
2021	cmis:lastModifiedBy	User who last modified the object.
2022	Required:	False
2023	Inherited:	False
2024	Property Type:	String
2025	Cardinality:	Single
2026	Updatability:	Read Only
2027	Choices:	Not Applicable
2028	Open Choice:	Not Applicable
2029	Repository MUST return this property with non-empty values when an object is requested and the	
2030	property filter does not exclude them	
2031		
2032	cmis:lastModificationDate	DateTime when the object was last modified.
2033	Required:	False
2034	Inherited:	False
2035	Property Type:	DateTime
2036	Cardinality:	Single
2037	Updatability:	Read Only
2038	Choices:	Not Applicable
2039	Open Choice:	Not Applicable
2040	Repository MUST return this property with non-empty values when an object is requested and the	
2041	property filter does not exclude them	
2042		
2043	cmis:changeToken	Opaque token used for optimistic locking & concurrency
2044		checking. (see section 2.2.1.3 Change Tokens)
2045	Required:	False
2046	Inherited:	False
2047	Property Type:	String
2048	Cardinality:	Single
2049	Updatability:	Read Only
2050	Choices:	Not Applicable
2051	Open Choice:	Not Applicable
2052	<u>Repository MUST return this property with non-empty values when an object is requested and the</u>	
2053	<u>property filter does not exclude them. If the repository does not support change tokens, this</u>	
2054	<u>property SHOULD not be set.</u>	
2055		
2056	cmis:sourceId	ID of the source object of the relationship.

2057	Required:	True
2058	Inherited:	False
2059	Property Type:	ID
2060	Cardinality:	Single
2061	Choices:	Not Applicable
2062	Open Choice:	Not Applicable
2063		
2064	cmis:targetId	ID of the target object of the relationship.
2065	Required:	True
2066	Inherited:	False
2067	Property Type:	ID
2068	Cardinality:	Single
2069	Choices:	Not Applicable
2070	Open Choice:	Not Applicable

2071 2.1.7 Policy Object

2072 A policy object represents an administrative policy that can be enforced by a repository, such as a
 2073 retention management policy. CMIS 1.0 does not specify what kinds of administrative policies that are
 2074 specifically supported, nor attempts to model administrative policy of any particular kind. Only a base
 2075 object-type is specified for policy objects. Each policy object holds the text of an administrative policy as a
 2076 repository-specific string, which is opaque to CMIS and which may be used to support policies of various
 2077 kinds. A repository may create subtypes of this base type to support different kinds of administrative
 2078 policies more specifically. If a repository does not support policy objects, the policy base object-type
 2079 | SHOULD NOT be returned by a `“_Get Types”` service call. This is an extension point for repositories that
 2080 want to expose other capabilities via CMIS that are not supported directly in CMIS 1.0.

2081 Aside from allowing an application to create and maintain policy objects, CMIS allows an application to
 2082 | `“_apply”` a policy to an object, and to remove an applied policy from an object. An object to which a policy
 2083 may be applied is called a *controllable* object. A policy MAY be applied to multiple controllable objects.
 2084 Conversely, a repository MAY allow multiple policies applied to a controllable object. (A repository may,
 2085 for example, impose constraints such as only one policy of each kind can be applied to an object.)
 2086 | Whether or not an object is controllable is specified by the object’s type definition. Applying a policy to an
 2087 object is to place the object under the control of that policy (while the object may also be under the control
 2088 of other policies at the same time), and removing an applied policy from one of its controlled objects is to
 2089 remove the corresponding control from that object. This control may change the state of the object, may
 2090 impose certain constraints on service calls operating on this object, or may cause certain management
 2091 actions to take place. The effect of this control, when this effect takes place, and how this control interacts
 2092 with other controls, are repository-specific. Only directly/explicitly applied policies are covered by CMIS
 2093 1.0. Indirectly applying policy to an object, e.g. through inheritance, is outside the scope of CMIS 1.0.

2094 A policy object does not have a content-stream and is not versionable. It may be fileable, queryable or
 2095 controllable. Policy objects are handled using the basic CRUD services for objects. If a policy is updated,
 2096 the change may alter the corresponding control on objects that the policy is currently applied to. If a
 2097 controlled object is deleted, all the policies applied to that object, if there are any, are removed from that
 2098 object. A policy object that is currently applied to one or more controllable objects CAN NOT be deleted.
 2099 That is, there is an implicit referential constraint from a controlled object to its controlling policy object(s).
 2100 | Besides the basic CRUD services, the `“_Apply Policy”` (*applyPolicy*) and the `“_Remove Policy”`
 2101 (*removePolicy*) services may be used to apply a policy object to a controllable object and respectively to
 2102 | remove an applied policy from one of its controlled objects. In addition, the `“_Get Applied Policies”`
 2103 (*getAppliedPolicies*) service may be used to obtain the policy objects that are currently applied to a
 2104 controllable object.


2148 **2.1.7.1 Policy Object-Type Definition**


2149 | This section describes the definition of the Policy Object-Type's attribute values and property definitions
2150 | which must be present on Policy instance objects. All attributes and property definitions are listed by their
2151 | ID.

2152 **2.1.7.1.1 Attribute Values**

2153 The Policy Object-Type MUST have the following attribute values.

2154 Notes:

2155 |  A value of <repository-specific> indicates that the value of the property MAY be set to
2156 | any valid value for the attribute type.

2157 |  Unless explicitly stated otherwise, all values specified in the table MUST be followed for
2158 | the Object-Type definition.

2159

2160 **id**

2161 Value: cmis:policy

2162

2163 **localName**

2164 Value: <repository-specific>

2165

2166 **localNamespace**

2167 Value: <repository-specific>

2168

2169 **queryName**

2170 Value: cmis:policy

2171

2172 **displayName**

2173 Value: <repository-specific>

2174

2175 **baseId**

2176 Value: cmis:policy

2177

2178 **parentId**

2179 Value: Not set

2180

2181 **description**

2182 Value: <repository-specific>

2183

2184 **creatable**

2185 Value: <repository-specific>

2186

2187 **fileable**

2188 Value: <repository-specific>

2189

2190 **queryable**

Formatted
0.24", Bull
+ Tab afte

2191 Value: <repository-specific>
 2192
 2193 **includedInSupertypeQuery**
 2194 Value: <repository-specific>
 2195
 2196 **controllablePolicy**
 2197 Value: <repository-specific>
 2198
 2199 **controllableACL**
 2200 Value: <repository-specific>
 2201
 2202 **fulltextIndexed**
 2203 Value: <repository-specific>
 2204

2205 **2.1.7.1.2 Property Definitions**

2206 The Policy base Object-Type MUST have the following property definitions, and MAY include additional
 2207 property definitions. Any attributes not specified by the Property Definitions are repository specific. For
 2208 all property definitions on base types, the query name MUST be the same as the property ID. The
 2209 repository MUST have the following property definitions on the Policy Type:

2210		
2211	cmis:name	Name of the object
2212	Inherited:	False
2213	Property Type:	String
2214	Cardinality:	Single
2215		
2216	cmis:objectId	Id of the object
2217	Required:	False
2218	Inherited:	False
2219	Property Type:	ID
2220	Cardinality:	Single
2221	Updatability:	Read Only
2222	Choices:	Not Applicable
2223	Open Choice:	Not Applicable
2224		
2225	cmis:baseTypeId	Id of the base object-type for the object
2226	Required:	False
2227	Inherited:	False
2228	Property Type:	ID
2229	Cardinality:	Single
2230	Updatability:	Read Only
2231	Choices:	Not Applicable
2232	Open Choice:	Not Applicable
2233		

2234	cmis:objectType	Id of the object's type
2235	Required:	False <u>True</u>
2236	Inherited:	False
2237	Property Type:	ID
2238	Cardinality:	Single
2239	Updatability:	oncreate
2240	Choices:	Not Applicable
2241	Open Choice:	Not Applicable
2242		
2243	cmis:createdBy	User who created the object.
2244	Required:	False
2245	Inherited:	False
2246	Property Type:	String
2247	Cardinality:	Single
2248	Updatability:	Read Only
2249	Choices:	Not Applicable
2250	Open Choice:	Not Applicable
2251		
2252	cmis:creationDate	DateTime when the object was created.
2253	Required:	False
2254	Inherited:	False
2255	Property Type:	DateTime
2256	Cardinality:	Single
2257	Updatability:	Read Only
2258	Choices:	Not Applicable
2259	Open Choice:	Not Applicable
2260		
2261	cmis:lastModifiedBy	User who last modified the object.
2262	Required:	False
2263	Inherited:	False
2264	Property Type:	String
2265	Cardinality:	Single
2266	Updatability:	Read Only
2267	Choices:	Not Applicable
2268	Open Choice:	Not Applicable
2269		
2270	cmis:lastModificationDate	DateTime when the object was last modified.
2271	Required:	False
2272	Inherited:	False
2273	Property Type:	DateTime
2274	Cardinality:	Single
2275	Updatability:	Read Only

2276	Choices:	Not Applicable
2277	Open Choice:	Not Applicable
2278		
2279	cmis:changeToken	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
2280		
2281	Required:	False
2282	Inherited:	False
2283	Property Type:	String
2284	Cardinality:	Single
2285	Updatability:	Read Only
2286	Choices:	Not Applicable
2287	Open Choice:	Not Applicable
2288		<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>
2289		
2290		
2291		
2292	cmis:policyText	User-friendly description of the policy
2293	Required:	True
2294	Inherited:	False
2295	Property Type:	String
2296	Cardinality:	Single
2297	Choices:	Not Applicable
2298	Open Choice:	Not Applicable

2299 2.1.8 Access Control

2300 A repository can support either a base set of CMIS-defined permissions and/or its own set of repository
2301 specific permissions.

2302 The getACL service allows the requestor to specify that the result be expressed using only the CMIS
2303 defined permissions. Without this restriction, the response may include, or be solely expressed in
2304 repository specific permissions. The applyACL service permits either CMIS permissions or repository
2305 permissions, or a combination of both, to be used.

2306 2.1.8.1 ACL, ACE, Principal, and Permission

2307 An **ACL** is a list of **Access Control Entries** (ACEs) and MAY hold zero or more ACEs. If an ACL has no
2308 ACEs, the behavior is the same as if the ACL is not set.

2309 An **ACE** holds:

- 2310 • one **Principal**: A principal represents a user management object, e.g. a user, group, or role.
2311 It holds one **String** with the **principalid**.
- 2312 • One or more **Strings** with the names of the **permissions**.
- 2313 • a **Boolean** flag **direct**, which indicates if TRUE the ACE is directly assigned to the object. If
2314 FALSE, that the ACE is somehow derived.

2315 2.1.8.2 CMIS Permissions

2316 There are three basic permissions predefined by CMIS:

- 2361 | • **cmis:read**: to be used to express “permission to read”. A Repository SHOULD express
2362 | the permission for reading properties AND reading content with this permission.
- 2363 | • **cmis:write**: to be used to express “permission to write”. SHOULD be used to express
2364 | permission to write properties and content of an object. MAY include other basic CMIS
2365 | permissions.
- 2366 | • **cmis:all**: SHOULD be used to express all the permissions of a repository. SHOULD
2367 | include all other basic CMIS permissions.

2368 | How these basic permissions can be mapped to the allowable actions is repository specific. However, the
2369 | actual repository semantics for the basic permissions with regard to allowable actions can be discovered
2370 | by the *mappings* parameter returned by *getRepositoryInfo* (see below).

2371 | Repositories MAY extend this set with repository-specific permissions.

2372 | 2.1.8.3 ACL Capabilities

2373 | Whether a repository supports ACLs at all, may be discovered via *capabilityACL* returned by
2374 | *getRepositoryInfo* (see section 2.1.1.1 Optional Capabilities). If *capabilityACL* is `none`, ACLs are not
2375 | supported by the repository.

2376 | If *capabilityACL* is `discover` or `manage`, additional information about the repositories permission model
2377 | and how changes to ACL are handled, can be discovered via the *getRepositoryInfo* service:

- 2378 | • ~~<Array>~~ **Enum propagation**: specifies, how non-direct ACEs can be handled by the
2379 | repository using the following values (see section 2.2.10.2 applyACL):
 - 2380 | ○ **objectonly** indicates, that the repository is able to apply ACEs to a document or folder,
2381 | without changing the ACLs of other objects.
 - 2382 | ○ **propagate**: indicates that the ACEs is to be applied to the given object and all inheriting
2383 | objects. *Propagate incorporates the support for objectonly.*
 - 2384 | ○ **repositorydetermined** indicates, that the repository has its own mechanism of
2385 | computing how changing an ACL for an object influences the non-direct ACEs of other
2386 | objects.
- 2387 | • **<Array> PermissionDefinition repositoryPermissions**: is a list with names and
2388 | descriptions of the supported permissions.
- 2389 | • **<Array> PermissionMapping mappings**: contains a list with mappings for the basic CMIS
2390 | permissions to allowed actions.

2391 | 2.1.8.3.1 Supported Permissions

2392 | The list of permission definitions returned by *getRepositoryInfo* lists all the *permissions* a repository
2393 | supports. This list also includes the CMIS permissions if supported by the repository.

2394 | A *PermissionDefinition* holds:

- 2395 | • **String permission**: the (technical) name of the permission (unique within the list of permission
2396 | definitions).
- 2397 | • **(Optional) String description**: an optional description of the permission that should be used as
2398 | the permission's name to be presented to the user.

2399 | 2.1.8.3.2 AllowableActions & Permission Mapping

2400 | CMIS provides a mechanism called “AllowableActions” which allows an application to discover the set of
2401 | service operations that can currently be performed on a particular object, without having to actually invoke
2402 | the service.

2403 | The set of allowable actions on an object at a point in time are affected not only by CMIS ACLs, but also
2404 | by other factors such as:

2451 | 1. Constraints inherent in the CMIS Domain Model based on the object's base type or current
2452 | versioning state.

2453 | 2. Policies or other control mechanisms that are opaque to CMIS.

2454 |
2455 | CMIS defines several services that applications can use at run-time to discover the AllowableActions for
2456 | an object.

2457 | If a Repository supports ACLs, then the repository MUST provide a mapping table that defines how the
2458 | permissions supported by the repository interact with the CMIS allowable actions, i.e. which permissions
2459 | are necessary for a principal to have on one or more objects in order to potentially perform each action,
2460 | subject to the other constraints on allowable actions above.

2461 | This section defines both the allowable actions as well as how those actions are presented in the
2462 | PermissionMapping table.

2463 | The Permission Mapping table contains a set of (*key*, *permissions*) pairs:

2464 | o **String Key:** Because several allowable actions may require permissions on more than one
2465 | object – for example, moving a document from one folder to another may require permissions
2466 | on the document and each of the folders – the mapping table is defined in terms of
2467 | permission “_keys_”, where each key combines the name of the allowable action as the object
2468 | for which the principal needs the required permission.

2469 | o For example – the canMoveObject.Source key indicates the permissions that the
2470 | principal must have on the “_source folder_” to move an object from that folder into
2471 | another folder.

2472 | o **<Array> String permissions:** The names of one or more permissions that the principal MUST
2473 | have. If more than one permission is specified, then the principal MUST be allowed to
2474 | perform the operation if they have ANY of the listed permissions.

2475 | The list below defines all mapping keys, as well as a permissions mapping that repositories SHOULD
2476 | use. Repositories MAY require additional permissions.

2477 | For convenience, the list below groups all mapping entries by the underlying Allowable Actions, and
2478 | includes descriptive information. For each Allowable Action the following information is given:

2479 Description:	The description and name of the service the AllowableAction enables.
2480 Base Object:	The base object-types for which the allowable action MAY be TRUE.
2481 Operand:	The object the permission applies to.
2482 Key:	The permission mapping key.
2483 Permissions:	The permission values.

2484 |

2485 | **Navigation Services:**

2486 | **canGetDescendants**

2487 Description:	Can get the descendants of the folder (<i>getDescendants</i>)
2488 Base Object:	cmis:folder
2489 Operand:	cmis:folder
2490 Key:	canGetDescendants.Folder
2491 Permission:	Read

2492 |

2493 | **canGetFolderTree**

2494 Description:	Can get the sub-folder tree of the folder (<i>getFolderTree</i>)
2495 Base Object:	cmis:folder
2496 Operand:	cmis:folder

2497 | ~~Key: canGetFolderTree.Folder~~

2498 | ~~Permission: Read~~

2499

2500 **canGetChildren**

2501 Description: Can get the children of the folder (`getChildren`)

2502 Base Object: `cmis:folder`

2503 Operand: `cmis:folder`

2504 Key: `canGetChildren.Folder`

2505 Permission: Read

2506

2507 **canGetFolderParent**

2508 Description: Can get the parent/ancestor folder(s) of the folder (`getFolderParent`)

2509 Base Object: `cmis:folder`

2510 Operand: `cmis:folder`

2511 | Key: `canGetFolderParent.FolderObject`

2512 Permission: Read

2513

2514 **canGetObjectParents**

2515 Description: Can get the parent folders of the object. (`getObjectParents`)

2516 Base Object: `cmis:document`, `cmis:folder`, `cmis:policy`

2517 Operand: Object

2518 | Key: `canGetObjectParents.Object`

2519 Permission: Read

2520

2521 **Object Services:**

2522 **canCreateDocument**

2523 Description: Can create a `cmis:document` Object in the folder (`createDocument`)

2524 Base Object: `cmis:folder`

2525 Operand: Folder

2526 Key: `canCreateDocument.Folder`

2527 Permission: Read

2528

2529 **canCreateFolder**

2530 Description: Can create a `cmis:folder` Object as a child of the specified folder
2531 (`createFolder`)

2532 Base Object: `cmis:folder`

2533 Operand: Folder

2534 Key: `canCreateFolder.Folder`

2535 Permission: Read

2536

2537 **canCreateRelationship**

2538 Description: Can create a Relationship in which this Object is a source
2539 (`createRelationship`)

2540 Base Object: cmis:document, cmis:folder
2541 Operand: Object
2542 Key: canCreateRelationship.Source
2543 Permission: Read

2544

2545 **canCreateRelationship**

2546 Description: Can create a Relationship in which this Object is a target
2547 (createRelationship)
2548 Base Object: cmis:document, cmis:folder
2549 Operand: Object
2550 Key: canCreateRelationship.Target
2551 Permission: Read

2552

2553 **canGetProperties**

2554 Description: Can read the properties of this object (getProperties)
2555 Base Object: cmis:document, cmis:folder, cmis:relationship, cmis:policy
2556 Operand: Object
2557 Key: canGetProperties.Object
2558 Permission: Read

2559

2560 ~~**canGetRenditions**~~

2561 ~~Description: Can retrieve the renditions of this object (getRenditions)~~
2562 ~~Base Object: cmis:document, or cmis:folder~~
2563 ~~Operand: Object~~
2564 ~~Key: canGetRenditions.Object~~
2565 ~~Permission: Read~~

2566

2567 ~~**canGetContentStream**~~

2568 ~~Description: Can get the content stream for the Document object~~
2569 ~~(getContentStream)~~
2570 ~~Base Object: cmis:document~~
2571 ~~Operand: Object~~
2572 ~~Key: canGetContentStream.Object~~
2573 ~~Permission: Read~~

2574

2575 **canUpdateProperties**

2576 Description: Can update the properties of this object (updateProperties)
2577 Base Object: cmis:document, cmis:folder, cmis:relationship, cmis:policy
2578 Operand: Object
2579 Key: canUpdateProperties.Object
2580 Permission: Write

2581

2582 **canMoveObject**

2583 Description: Can move the object (`moveObject`)

2584 Base Object: `cmis:document, cmis:folder, cmis:policy`

2585 Operand: Object

2586 Key: `canMoveObject.Object`

2587 Permission: Write

2588

2589 **canMoveObject**

2590 Description: Can move an object into this folder (`moveObject`)

2591 Base Object: `cmis:folder`

2592 Operand: Folder

2593 Key: `canMoveObject.Target`

2594 Permission: Read

2595

2596 **canMoveObject**

2597 Description: Can move an object from this folder (`moveObject`)

2598 Base Object: `cmis:folder`

2599 Operand: Folder

2600 Key: `canMoveObject.Source`

2601 Permission: Read

2602

2603 **canDeleteObject**

2604 Description: Can delete this object (`deleteObject`)

2605 Base Object: `cmis:document, cmis:folder, cmis:relationship, cmis:policy`

2606 Operand: Object

2607 Key: `canDelete.Object`

2608 Permission: Write

2609

2610 ~~**canDeleteObject**~~

2611 ~~Description: Can delete an object that is a child of this folder (`deleteObject`)~~

2612 ~~Base Object: `cmis:folder`~~

2613 ~~Operand: Folder~~

2614 ~~Key: `canDelete.Folder`~~

2615 ~~Permission: Read~~

2617 **canGetContentStream**

2618 Base Object: `cmis:document`

2619 Action: Can get the content stream for the Document object

2620 (`getContentStream`)

2621 Operand: Object

2622 Key: `canViewContent.Object`

2623 Permission: Read

2624

2625	canSetContentStream	
2626	Description:	Can set the content stream for the Document object
2627		(setContentStream)
2628	Base Object:	cmis:document
2629	Operand:	Object
2630	Key:	canSetContentStream.Document
2631	Permission:	Write
2632		
2633	canDeleteContentStream	
2634	Base Object:	cmis:document
2635	Action:	Can delete the content stream for the Document object
2636		(deleteContentStream)
2637	Operand:	Object
2638	Key:	canDeleteContentStream.Document
2639	Permission:	Write
2640		
2641	canDeleteTree	
2642	Base Object:	cmis:folder
2643	Action:	Can delete the folder and all contained objects (deleteTree)
2644	Operand:	Object
2645	Key:	canDeleteTree.Folder
2646	Permission:	Write
2647		
2648	Filing Services:	
2649	canAddObjectToFolder	
2650	Description:	Can file the document in a folder (addObjectToFolder)
2651	Base Object:	cmis:document, cmis:policy
2652	Operand:	Object
2653	Key:	canAddToFolder.Object
2654	Permission:	Read
2655		
2656	canAddObjectToFolder	
2657	Description:	Can file a document in the specified folder (addObjectToFolder)
2658	Base Object:	cmis:document, cmis:policy
2659	Operand:	Object
2660	Key:	canAddToFolder.Folder
2661	Permission:	Read
2662		
2663	canRemoveObjectFromFolder	
2664	Description:	Can unfile the specified document from a folder
2665		(removeObjectFromFolder)
2666	Base Object:	cmis:document, cmis:policy
2667	Operand:	Object

2668 | Key: canRemoveObjectFromFolder.Object
2669 | Permission: Read

2670

2671 **canRemoveObjectFromFolder**

2672 | Description: Can unfile a document from the specified folder
2673 | (removeObjectFromFolder)

2674 | Base Object: cmis:document, cmis:policy

2675 | Operand: Object

2676 | Key: canRemoveObjectFromFolder.Folder

2677 | Permission: Read

2678

2679 **Versioning Services:**

2680 **canCheckOut**

2681 | Description: Can check out the Document object (checkOut)

2682 | Base Object: cmis:document

2683 | Operand: Object

2684 | Key: ~~canCheckOut~~canCheckout.Document

2685 | Permission: Write

2686

2687 **canCancelCheckOut**

2688 | Description: Can cancel the check out the Document object (cancelCheckOut)

2689 | Base Object: cmis:document

2690 | Operand: Object

2691 | Key: canCancelCheckout.Document

2692 | Permission: Write

2693

2694 **canCheckIn**

2695 | Description: Can check in the Document object (checkIn)

2696 | Base Object: cmis:document

2697 | Operand: Object

2698 | Key: canCheckin.Document

2699 | Permission: Write

2700

2701 **canGetAllVersions**

2702 | Description: Can get the version series for the Document object (getAllVersions)

2703 | Base Object: cmis:document

2704 | Operand: Object

2705 | Key: canGetAllVersions.VersionSeriesDocument

2706 | Permission: Read

2707

2708 **Relationship Services:**

2709 **canGetObjectRelationships**

2710 Description: Can get the relationship in which this object is a source/target
2711 (getObjectRelationships)
2712 Base Object: cmis:document, cmis:folder, cmis:policy
2713 Operand: Object
2714 Key: canGetObjectRelationships.Object
2715 Permission: Read

2716

2717 Policy Services:

2718 **canApplyPolicy**

2719 Description: Can apply a policy to the Object (applyPolicy)
2720 Base Object: cmis:document, cmis:folder
2721 Operand: Object
2722 Key: canAddPolicy.Object
2723 Permission: Read

2724

2725 **canApplyPolicy**

2726 Description: Can apply the specified policy to an Object (applyPolicy)
2727 Base Object: cmis:policy
2728 Operand: Object
2729 Key: canAddPolicy.Policy
2730 Permission: Read

2731

2732 **canRemovePolicy**

2733 Description: Can remove a policy from the specified Object (removePolicy)
2734 Base Object: cmis:document, cmis:folder
2735 Operand: Object
2736 Key: canRemovePolicy.Object
2737 Permission: Read

2738

2739 **canRemovePolicy**

2740 Description: Can remove the specified policy from an Object (removePolicy)
2741 Base Object: cmis:document, cmis:folder
2742 Operand: cmis:policy
2743 Key: canRemovePolicy.Policy
2744 Permission: Read

2745

2746 **canGetAppliedPolicies**

2747 Description: Can get the list of Policies applied to the Object
2748 (getAppliedPolicies)

2749	Base Object:	cmis:document, cmis:folder
2750	Operand:	Object
2751	Key:	canGetAppliedPolicies.Object
2752	Permission:	Read
2753		
2754	ACL Services:	
2755	canGetACL	
2756	Description:	Can get ACL for object (<code>getACL</code>)
2757	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2758	Operand:	Object
2759	Key:	canGetACL.Object
2760	Permission:	Read
2761		
2762	canApplyACL	
2763	Description:	Can apply ACL to this object (<code>applyACL</code>)
2764	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2765	Operand:	Object
2766	Key:	canApplyACL.Object
2767	Permission:	Write
2768		

2769

2770 2.1.9 Versioning

2771 CMIS supports versioning of Document objects. Folder objects, relationship objects, and policy objects
2772 cannot be versioned.

2773 Whether or not a Document object is versionable (i.e. whether or not operations performed on the object
2774 via the Versioning Services MUST be allowed) is specified by the `“_versionable”` attribute on its Object-
2775 type.

2776 A **version** of a Document object is an explicit `“_deep”` copy of the object, preserving its state at a certain
2777 point in time. Each version of a Document object is itself a Document object, i.e. has its own *ObjectId*,
2778 property values, MAY be acted upon using all CMIS services that act upon Document objects, etc.

2779 2.1.9.1 Version Series

2780 A **version series** for a Document object is a transitively closed collection of all Document objects that
2781 have been created from an original Document in the Repository. Each version series has a unique,
2782 system-assigned, and immutable **version series ID**.

2783 The version series has transitive closure -- that is, if object B is a version of object A, and object C is a
2784 version of object B, then object C is also a version of object A. The objects in a version series can be
2785 conceptually sequenced by their respective *CreationDate* properties.

2786 Additionally, the repository MAY expose a textual **VersionLabel** that describes to a user the position of
2787 an individual object with respect to the version series. (For example, version 1.0).

2788 *Note:* A Document object that is NOT versionable will always have a single object in its Version Series. A
2789 versionable Document object MAY have one or more objects in its Version Series.

2790 2.1.9.2 Latest Version

2791 The version that has the most recent *LastModificationDate* is called the **Latest Version** of the series, or
2792 equivalently, the latest version of any Document object in the series.

2793 When the latest version of a version series is deleted, a previous version (if there is one) becomes the
2794 latest version.

2795 2.1.9.2.1 Behavioral constraints on non-Latest Versions

2796 Repositories NEED NOT allow the non-latest versions in a Version Series to be updated, queried, or
2797 searched.

2798 2.1.9.3 Major Versions

2799 A Document object in a Version Series MAY be designated as a **Major Version**.

2800 The CMIS specification does not define any semantic/behavioral differences between Major and non-
2801 Major versions in a Version Series. Repositories may enforce/apply additional constraints or semantics for
2802 Major versions, if the effect on CMIS services remains consistent with an allowable behavior of the CMIS
2803 model.

2804 If the Version Series contains one or more Major versions, the one that has the most recent
2805 *LastModificationDate* is the **Latest Major Version** of the version series.

2806 (Note that while a Version Series MUST always have a *Latest Version*, it NEED NOT have a *Latest Major*
2807 *Version*.)

2808 When the latest major version is deleted, a previous major version (if there is one) becomes the latest
2809 major version.

2855 2.1.9.4 Services that modify Version Series

2856 2.1.9.4.1 Checkout

2857 A new version of a versionable Document object is created when the *checkIn* service is invoked on the
2858 Private Working copy (PWC) of this object. A PWC is created by invoking *checkOut* on a versionable
2859 Document object. A repository MAY allow *any* Document object in a version series to be checked out, or
2860 MAY only allow the *Latest Version* to be checked out.

2861 The effects of invoking the *checkout* service MUST be as follows:

- 2862 • A new Document object, referred to herein as the **Private Working Copy (PWC)**, is created.
 - 2863 | ○ The PWC NEED NOT be visible to users who have permissions to view other Document
 - 2864 | objects in the Version Series.
 - 2865 | ○ Until it is checked in (using the *checkIn* service), the PWC MUST NOT be considered the
 - 2866 | *LatestMajorVersion* in the Version Series.
 - 2867 | ○ The property values for the PWC SHOULD be identical to the properties of the Document
 - 2868 | object on which the *checkout* service was invoked. Certain properties such as `cmis:objectId`
 - 2869 | may be different. Properties such as `cmis:creationDate` most likely will be different. The
 - 2870 | content-stream of the PWC MAY be identical to the content-stream of the Document object
 - 2871 | on which the *checkout* service was invoked, or MAY be `"not set"`.

2872 After a successful *checkout* operation is completed, and until such time when the PWC is deleted (via the
2873 *cancelCheckOut* service) or checked-in (via the *checkIn*) service, the effects on other Documents in the
2874 Version Series MUST be as follows:

- 2875 • The repository MUST throw an exception if the *checkout* service is invoked on any Document in
2876 the Version Series. (I.e. there can only be one PWC for a version series at a time.)
- 2877 • The value of the `cmis:isVersionSeriesCheckedOut` property MUST be TRUE.
- 2878 • The value of the `cmis:versionSeriesCheckedOutBy` property MAY be set to a value indicating
2879 | which user created the PWC. (The Repository MAY still show the `"not set"` value for this
2880 | property.)
- 2881 • The value of the `cmis:versionSeriesCheckedOutId` property MAY be set to the Objectid of the
2882 | PWC. (The Repository MAY still show the `"not set"` value for this property).
- 2883 • The repository MAY prevent operations that modify or delete the other Documents in the Version
2884 | Series.

2885 2.1.9.4.2 Updates to the Private Working Copy

2886 | If the repository supports the optional `"PWCUpdatable"` capability, then the repository MUST allow
2887 | authorized users to modify the PWC Object using the Object services (e.g. *UpdateProperties*).

2888 | If the repository does NOT support the `"PWCUpdatable"` capability, then the PWC object can only be
2889 | modified as part of the *checkIn* service call.

2890 2.1.9.4.3 Discarding Check out

2891 An authorized user MAY discard the check-out using the *cancelCheckOut* service on any Document in
2892 the Version Series or by using the *deleteObject* service on the PWC Object. The effects of discarding a
2893 check-out MUST be as follows:

- 2894 • The PWC Object MUST be deleted.
- 2895 • For all other Documents in the Version Series:
 - 2896 | ○ The value of the `cmis:isVersionSeriesCheckedOut` property MUST be FALSE.
 - 2897 | ○ The value of the `cmis:versionSeriesCheckedOutBy` property MUST be `"not set"`.
 - 2898 | ○ The value of the `cmis:versionSeriesCheckedOutId` property MUST be `"not set"`.
 - 2899 | ○ The repository MUST allow authorized users to invoke the *checkout* service.

2943 2.1.9.4.4 Checkin

2944 | An authorized user/application MAY “check in” the Private Working Copy object via the *checkIn* service.
2945 | The *checkIn* service allows users/applications to provide update property values and a content-stream for
2946 | the PWC object.

2947 | The effects of the checkIn service MUST be as follows for successful checkins:

- 2948 | • The PWC object MUST be updated as specified by the inputs to the *checkIn* service. (Note that for
2949 | repositories that do NOT support the “PWCUpdatable” property, this is the only way to update the
2950 | PWC object.)
- 2951 | • The Document object resulting from the *checkIn* operation MUST be considered the *Latest Version*
2952 | in the Version Series.
- 2953 | • If the inputs to the *checkIn* service specified that the PWC MUST be a “major version”, then the
2954 | PWC MUST be considered the *Latest Major Version* in the Version Series.
- 2955 | • If the checkin returns a new cmis:objected, then the PWC object MUST disappear if the *checkIn* call
2956 | was successful and the new checked in version will use the new specified id.
- 2957 | • For all Documents in the Version Series:
 - 2958 | ◦ The value of the cmis:isVersionSeriesCheckedOut property MUST be FALSE.
 - 2959 | ◦ The value of the cmis:versionSeriesCheckedOutBy property MUST be “not set”.
 - 2960 | ◦ The value of the cmis:versionSeriesCheckedOutId property MUST be “not set”.
 - 2961 | ◦ The repository MUST allow authorized users to invoke the *checkout* service.

2962 | *Note:* The Repository MAY change the ID of the PWC upon completion of the *checkin* service invocation.

2963 | *Note:* A repository MAY automatically create new versions of Document objects without an explicit
2964 | invocation of the checkout/checkin services.

2965 2.1.9.5 Versioning Properties on Document Objects

2966 | All Document objects will have the following read-only property values pertaining to versioning:

2967		
2968	cmis:isLatestVersion	Boolean
2969	TRUE if the Document object is the <i>Latest Version</i> in its <i>Version Series</i> . FALSE otherwise.	
2970		
2971	cmis:isMajorVersion	Boolean
2972	TRUE if the Document object is a <i>Major Version</i> in its <i>Version Series</i> . FALSE otherwise.	
2973		
2974	cmis:isLatestMajorVersion	Boolean
2975	TRUE if the Document object is the <i>Latest Major Version</i> in its <i>Version Series</i> . FALSE otherwise.	
2976		
2977	cmis:versionLabel	String (optional)
2978	Optional textual description the position of an individual object with respect to the version series. 2979 (For example, version 1.0).	
2980		
2981	cmis:versionSeriesId	ID
2982	ID of the Version Series for this Object.	
2983		
2984	cmis:isVersionSeriesCheckedOut	Boolean
2985	TRUE if there currently exists a Private Working Copy for this Version Series. FALSE otherwise	

Formatted
Level: 2 +
0.88"

3030

3031 **cmis:versionSeriesCheckedOutBy** String

3032 If IsVersionSeriesCheckedOut is TRUE: then an identifier for the user who created the Private

3033 Working Copy. `""`Not set otherwise.

3034

3035 **cmis:versionSeriesCheckedOutId** ID

3036 If IsVersionSeriesCheckedOut is TRUE: The Identifier for the Private Working Copy. `""`Not set

3037 otherwise.

3038

3039 **cmis:checkinComment** String

3040 Textual comment associated with the given version.

3041 *Note:* Changes made via the Versioning Services that affect the values of these properties MUST NOT

3042 constitute modifications to the Document objects in the Version Series (e.g. MUST NOT affect the

3043 cmis:lastModificationDate, etc.)

3044 2.1.9.6 Document Creation and Initial Versioning State

3045 A repository MAY create new Document objects in a `""Private Working Copy""` state when they are

3046 created via the *createDocument* or *createDocumentFromSource* services. This state is logically

3047 equivalent to having a Version Series that contains exactly one object (the PWC) and 0 other documents.

3048 The repository MAY also create new Document objects in a `""Major Version""` state. This state is logically

3049 equivalent to having a Version Series that contains exactly one Major Version and 0 other documents.

3050 The repository MAY also create new Document objects in a `""Non-Major Version""` state. This state is

3051 logically equivalent to having a Version Series that contains exactly one Non-Major Version and 0 other

3052 documents.

3053 If the repository does not support versioning the repository MUST ignore the value of the versioningState

3054 parameter.

3055 2.1.9.7 Version Specific/Independent membership in Folders

3056 Repositories MAY treat membership of a Document object in a folder collection as `""version-specific""` or

3057 `""version-independent""`.

3058 Repositories MUST indicate whether they support version-specific membership in a folder via the

3059 `""VersionSpecificFiling""` optional capability flag.

3060 If the repository is treating folder collection membership as `""version-independent""`, then:

- 3061 • Moving or Filing a Document Object into a folder MUST result in ALL Documents in the Version
- 3062 Series being moved/added into the folder.
- 3063 • The Repository MAY return only the latest-version OR latest major-version Document object in a
- 3064 version series in the response to Navigation service requests (*getChildren*, *getDescendants*), and
- 3065 NEED NOT return other Document Objects filed in the folder that are in the Version Series.

3066 If the repository is treating folder collection membership as `""version-specific""`, then moving or Filing a

3067 Document Object into a folder MUST NOT result in other Documents in the Version Series being

3068 moved/added.

3069 2.1.9.8 Version Specific/Independent membership in Relationships

3070 A relationship object MAY have either a version-specific or version-independent binding to its source

3071 and/or target objects. This behavior MAY vary between repositories and between individual relationship

3072 types defined for a Repository.

3073 If a relationship object has a version-independent binding to its source/target object, then:

Formatted
0.24", Bull
+ Tab after

3108 | • The getObjectRelationships service invoked on a Document Object MUST return the relationship
3109 | if Relationship was source/target is set to ANY Document Object in the Version Series.

3110 | If a relationship object has a version-specific binding to its source/target object, then:

3111 | • The getObjectRelationships service invoked on a Document Object MUST return the relationship
3112 | if Relationship was source/target is set to the ID of the Document Object on which the service was
3113 | invoked.

3114 | **2.1.9.9 Versioning visibility in Query Services**

3115 | Repositories MAY include non-latest-versions of Document Objects in results to the Discovery Services
3116 | (*query*).

3117 | Repositories MUST indicate whether they support querying for non-latest-versions via the
3118 | `"AllVersionsSearchable"` optional capability flag.

3119 | If `"AllVersionsSearchable"` is TRUE then the Repository MUST include in the query results ANY
3120 | Document Object in the Version Series that matches the query criteria. (subject to other query constraints
3121 | such as security.)

3122 | Additionally, repositories MAY include Private Working Copy objects in results in results to the Discovery
3123 | Services (*query*).

3124 | Repositories MUST indicate whether they support querying for Private Working Copy objects via the
3125 | `"PWCSearchable"` optional capability flag.

3126 | If `"PWCSearchable"` is TRUE then the Repository MUST include in the query results ANY Private
3127 | Working Copy Document Objects that matches the query criteria (subject to other query constraints such
3128 | as security.)

3129 | If `"PWCSearchable"` is FALSE then the Repository MUST NOT include in the query results ANY Private
3130 | Working Copy Document Objects that match the query criteria (subject to other query constraints such as
3131 | security.)

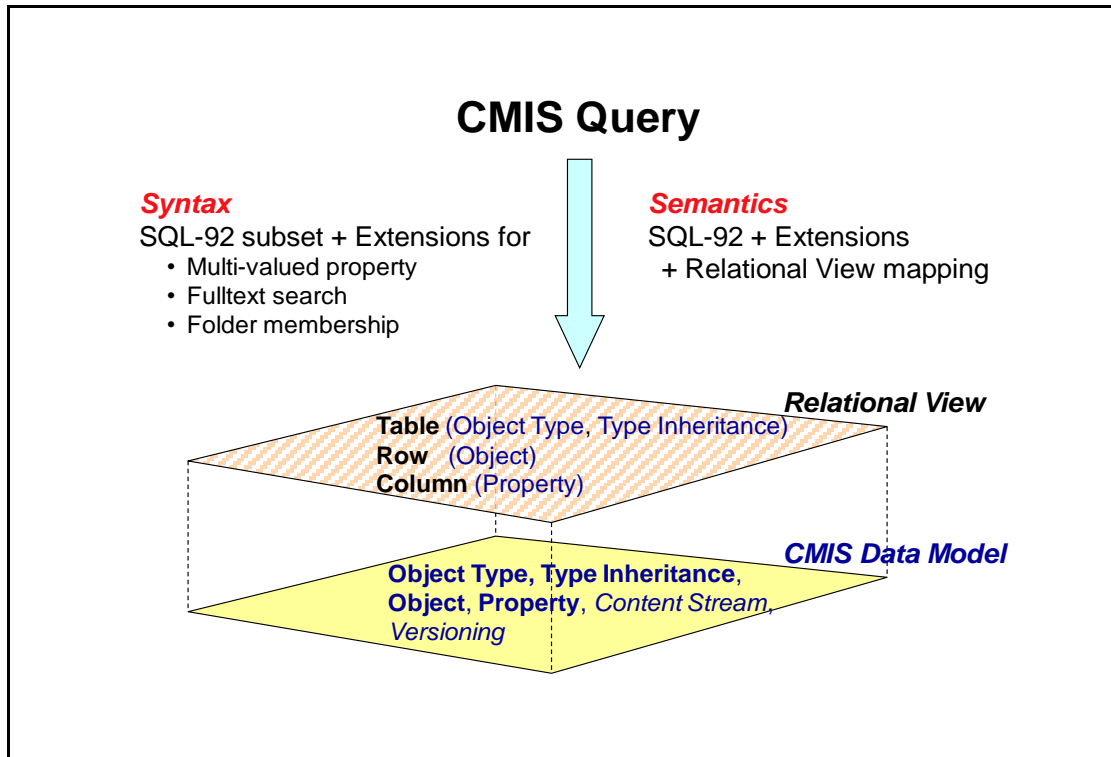
3132 | **2.1.10 Query**

3133 | CMIS provides a type-based query service for discovering objects that match specified criteria, by
3134 | defining a read-only projection of the CMIS data model into a *Relational View*.

3135 | Through this relational view, queries may be performed via a simplified SQL SELECT statement. This
3136 | query language is based on a subset of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database
3137 | Language SQL), with a few extensions to enhance its filtering capability for the CMIS data model, such as
3138 | existential quantification for multi-valued property, full-text search, and folder membership. Other
3139 | statements of the SQL language are not adopted by CMIS. The semantics of this query language is
3140 | defined by the SQL-92 standard, plus the extensions, in conjunction with the model mapping defined by
3141 | CMIS's relational view.

Formatted
0.24", Bul
+ Tab afte
stops: Not

Formatted
0.24", Bul
+ Tab afte
stops: Not



3142

3143 2.1.10.1 Relational View Projection of the CMIS Data Model

3144 The relational view of a CMIS repository consists of a collection of virtual tables that are defined on top of
3145 the CMIS data model. This relational view is used for query purposes only.

3146 In this relational view a **Virtual Table** is implicitly defined for each *queryable* Object-Type defined in the
3147 repository. (Non-queryable Object-Types are NOT exposed through this Relational View.)

3148 In each **Virtual Table**, a **Virtual Column** is implicitly defined for each property defined in the Object-Type
3149 Definition AND for all properties defined on ANY ancestor-type of the Object-Type but NOT defined in the
3150 Object-Type definition. Virtual Columns for properties defined on ancestor-types of the Object-type but
3151 NOT defined in the Object-Type definition MUST contain the SQL NULL value. Virtual Columns for
3152 properties whose value is `"not set"` MUST contain the SQL NULL value.

3153 An object-type's *queryName* attribute is used as the table name for the corresponding virtual table, and a
3154 property's *queryName* attribute is used as the column name for the corresponding table column. Please
3155 see the restrictions on *queryName* in the appropriate data model section.

3156 The Virtual Column for a multi-valued property MUST contain a single list value that includes all values of
3157 the property.

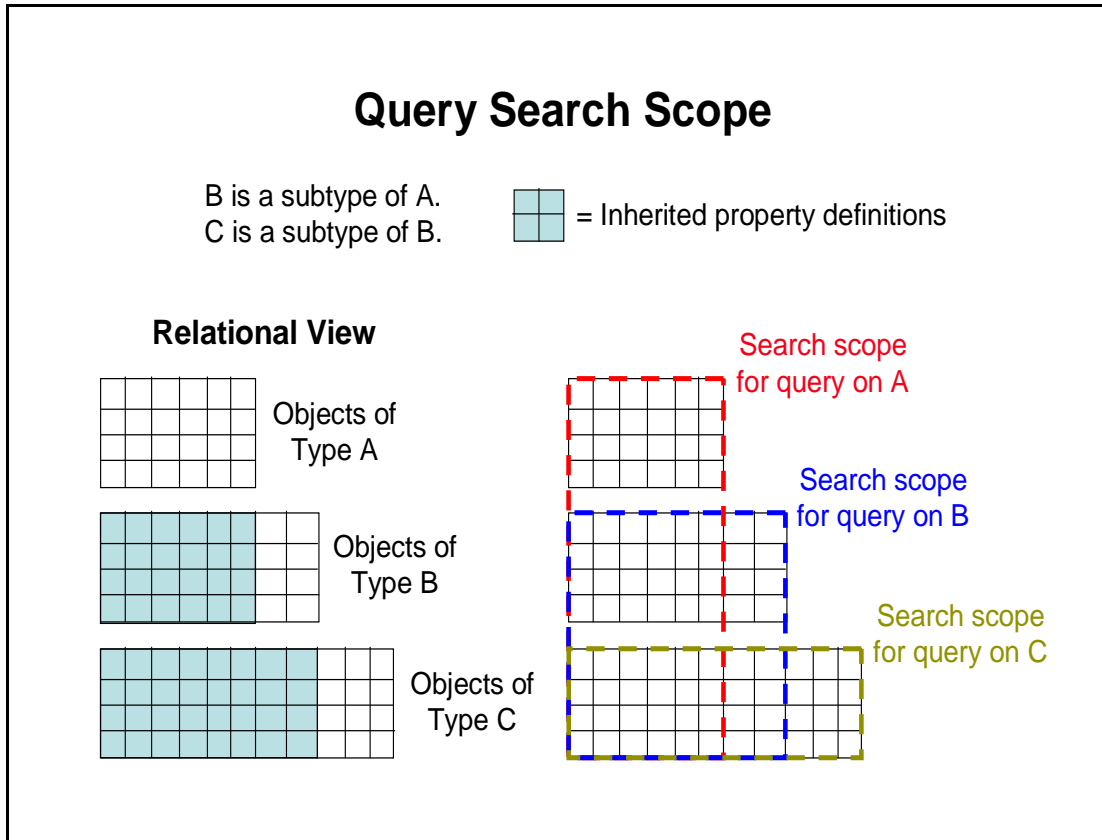
3158 2.1.10.1.1 Object-Type Hierarchy in the Relational View Projection

3159 The Relational View projection of the CMIS Data Model ensures that the Virtual Table for a particular
3160 Object-type is a complete super-set of the Virtual Table for any and all of its ancestor types.

3161 Additionally, an Object-Type definition's `"includedInSupertypeQuery"` specifies whether objects of that
3162 Object-Type MUST be included in the Virtual Table for any of its ancestor types. If the
3163 `"includedInSupertypeQuery"` attribute of the Object-Type is FALSE, then objects of that Object-Type
3164 MUST NOT be included in the Virtual Table for any of its ancestor types.

3165 Thus the Virtual Table for an Object-type includes a row not only for each Object of that type, but all
 3166 | Objects of any of that Object-types' Descendant Types for which the `includedInSupertypeQuery`
 3167 attribute is TRUE.

3168 But since the Virtual Table will include only columns for properties defined in the Object-Type underlying
 3169 the Virtual Table, a row that is a query result representing an Object of a Descendant Type can only
 3170 include those columns for properties defined on the Object-Type underlying the Virtual Table.



3171

3172 2.1.10.1.2 Content Streams

3173 Content-streams are NOT exposed through this relational view.

3174 2.1.10.1.3 Result Set

3175 When a query is submitted, a set of pseudo CMIS objects will be returned. These pseudo objects are
 3176 comprised of the properties specified in the select clause of the query statement.

3177 For each property in each object in the result set, the Repository MUST include the property definition ID
 3178 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
 3179 used).

3180 If the select clause of the query statement contains properties from a single type reference then the
 3181 repository MAY represent these pseudo-objects with additional object information.

3182 2.1.10.2 Query Language Definition

3183 This query languages is based on a subset of the SQL-92 grammar. CMIS-specific language extensions
 3184 to SQL-92 are called out explicitly.

3185 The basic structure of a CMIS query is a SQL statement that MUST include the following clauses:

3186 • **SELECT [virtual columns]:** This clause identifies the set of virtual columns that will be included
3187 in the query results for each row.

3188 • **FROM [Virtual Table Names]:** This clause identifies which Virtual Table(s) the query will run
3189 against.

3190 Additionally, a CMIS query MAY include the following clauses:

3191 • **WHERE [conditions]:** This clause identifies the constraints that rows MUST satisfy to be
3192 considered a result for the query.

3193 • **ORDER BY [sort specification]:** This clause identifies the order in which the result rows MUST
3194 be sorted in the result row set.

3195 2.1.10.2.1 BNF Grammar

3196 | This BNF grammar is a "subset" of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language
3197 SQL), except for some production alternatives. Specifically, except for these extensions, the following
3198 production rules are derived from the SQL-92 grammar. The non-terminals used in this grammar are also
3199 borrowed from the SQL-92 grammar without altering their semantics. Accordingly, the non-terminal
3200 <column name> is used for single-valued properties only so that the semantics of SQL can be preserved
3201 and borrowed. This approach not only facilitates comparison of the two query languages, and simplifies
3202 the translation of a CMIS query to a SQL query for a RDBMS-based implementation, but also allows
3203 future expansion of this query language to cover a larger subset of SQL with minimum conflict. The CMIS
3204 extensions are introduced primarily to support multi-valued properties and full-text search, and to test
3205 folder membership. Multi-valued properties are handled separately from single-valued properties, using
3206 separate non-terminals and separate production rules to prevent the extensions from corrupting SQL-92
3207 semantics.
3208

3209
3210 <CMIS 1.0 query statement> ::= <simple table> [<order by clause>]
3211 <simple table> ::= SELECT <select list> <from clause> [<where clause>]
3212 <select list> ::= "*"
3213 | <select sublist> [{ "_" <select sublist> }...]
3214 <select sublist> ::= <value expression> [[AS] <column name>]
3215 | <qualifier> "*"
3216 | <multi-valued-column reference>
3217 <value expression> ::= <column reference> | <numeric value function>
3218 <column reference> ::= [<qualifier> "_"] <column name>
3219 <multi-valued-column reference> ::= [<qualifier> "_"] <multi-valued-column name>
3220 <numeric value function> ::= SCORE()
3221 <qualifier> ::= <table name> | <correlation name>
3222 <from clause> ::= FROM <table reference>
3223 <table reference> ::= <table name> [[AS] <correlation name>]
3224 | <joined table>
3225 <joined table> ::= "(" <joined table>)"
3226 | <table reference> [<join type>] JOIN <table reference> <join specification>
3227 <join type> ::= INNER | LEFT [OUTER]
3228 <join specification> ::= ON <column reference> "=" <column reference>
3229 <where clause> ::= WHERE <search condition>
3230 <search condition> ::= <boolean term> | <search condition> OR <boolean term>
3231 <boolean term> ::= <boolean factor> | <boolean term> AND <boolean factor>
3232 <boolean factor> ::= [NOT] <boolean test>
3233 <boolean test> ::= <predicate> | "(" <search condition>)"
3234 <predicate> ::= <comparison predicate> | <in predicate> | <like predicate> | <null predicate>
3235 | <quantified comparison predicate> | <quantified in predicate>
3236 | <text search predicate> | <folder predicate>
3237 <comparison predicate> ::= <value expression> <comp op> <literal>
3238 <comp op> ::= "=" | "<" | ">" | "<=" | ">=" | "<=" | ">="
3239 <literal> ::= <signed numeric literal> | <character string literal> | <datetime literal> | <boolean literal>
3240 <in predicate> ::= <column reference> [NOT] IN "(" <in value list>)"
3241 <in value list> ::= <literal> [{ "_" <literal> }...]
3242 <like predicate> ::= <column reference> [NOT] LIKE <character string literal>
3243 <null predicate> ::= { <column reference> | <multi-valued-column reference> } IS [NOT] NULL
3244 <quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>
3245 <quantified in predicate> ::= ANY <multi-valued-column reference> [NOT] IN "(" <in value list>)"
3246 <text search predicate> ::= CONTAINS "("
3247 | [<qualifier> "_"] <quote> <text search expression> <quote>)"
3248 <folder predicate> ::= { IN_FOLDER | IN_TREE } "(" [<qualifier> "_"] <folder id>)"
3249 <order by clause> ::= ORDER BY <sort specification> [{ "_" <sort specification> }...]
3250 <sort specification> ::= <column reference> [ASC | DESC]
3251 <correlation name> ::= <identifier>

3293 <table name> ::= <identifier> !! This MUST be the name of an object-type.

3294 <column name> ::= <identifier> !! This MUST be the name of a single-valued property,

3295 or an alias for a scalar output value.

3296 <multi-valued-column name> ::= <identifier> !! This MUST be the name of a multi-valued property.

3297 <folder id> ::= <character string literal> !! This MUST be the object identity of a folder object.

3298 <identifier> ::= !! As defined by queryName attribute.

3299 <signed numeric literal> ::= !! As defined by SQL-92 grammar.

3300 <character string literal> ::= !! As defined by SQL-92 grammar. (i.e. enclosed in single-quotes)

3301

3302 | !! This is [an independent sub-grammar for](#) full-text search criteria. [It is isolatable from the query](#)

3303 | [statement grammar. \(See 2.1.10.3 Escaping\)](#)

3304 <text search expression> ::= <conjunct> [{<space> OR <space> <conjunct>} ...]

3305 <conjunct> ::= <term> [{<space> <term>} ...]

3306 | <term> ::= ["-"] <simple term>

3307 <simple term> ::= <word> | <phrase>

3308 <word> ::= <non space char> [{<non space char>} ...]

3309 <phrase> ::= <quote> <word> [{<space> <word>} ...] <quote>

3310 <space> ::= <space char> [{<space char>} ...]

3311 <non space char> ::= <char> - <space char>

3312 | <space char> ::= " "

3313 <char> ::= !! Any character

3314

3315 <datetime literal> ::= TIMESTAMP <quote> <datetime string> <quote>

3316 <datetime string> ::= YYYY-MM-DDThh:mm:ss.sss[Z | +hh:mm | -hh:mm]

3317 <boolean literal> ::= TRUE | FALSE | true | false

3318 | <quote> ::= "''"!! Single-quote only, consistent with SQL-92 string literal

3319 2.1.10.2.2 SELECT Clause

3320 The SELECT clause MUST contain exactly one of the following:

- 3321 • A comma separated list of one or more column names.
- 3322 | • If an explicit column list is provided: A repository MUST include in its result row set all of the
- 3323 | columns specified in the SELECT clause.
- 3324 • * : If this token is specified, then the repository MUST return columns for ALL single-valued
- 3325 | properties defined in the Object-Types whose Virtual Tables are listed in the FROM clause, and
- 3326 | SHOULD also return all multi-valued properties.

3327 | All column names MUST be valid "queryName" values for properties that are defined as "queryable" in

3328 | the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

3329 2.1.10.2.3 FROM Clause

3330 The FROM clause identifies which Virtual Table(s) the query will be run against, as described in the

3331 previous section.

3332 The FROM clause MUST contain only the queryNames of Object-Types whose *queryable* attribute value

3333 is TRUE.

Formatted
Level: 2 +

3334 **2.1.10.2.3.1 Join Support**

3335 CMIS repositories MAY support the use of SQL JOIN queries, and MUST indicate their support level
3336 using the [Optional Capability](#) attribute `capabilityJoin`.

3337 | • If the Repository's value for the capabilityJoin attribute is **none**, then no JOIN clauses can be
3338 used in queries.

3339 | • If the Repository's value for the capabilityJoin attribute is **inneronly**, then only inner JOIN
3340 clauses can be used in queries.

3341 | • If the Repository's value for the capabilityJoin attribute is **innerandouter**, then inner and/or outer
3342 JOIN clauses can be used in queries.

3343 | Only explicit joins using the `JOIN` keyword is supported. Queries MUST NOT include implicit joins as
3344 part of the WHERE clause of a CMIS query.

3345 | CMIS queries MUST only support join operations using the `equality` predicate on single-valued
3346 properties.

3347 **2.1.10.2.4 WHERE Clause**

3348 This clause identifies the constraints that rows MUST satisfy to be considered a result for the query.

3349 | All column names MUST be valid `queryName` or their aliased values for properties that are defined as
3350 `queryable` in the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

3351 | Properties are defined to not support a `null` value, therefore the `<null predicate>` MUST be interpreted
3352 as testing the not set or set state of the specified property.

3353 **2.1.10.2.4.1 Comparisons permitted in the WHERE clause.**

3354 | SQL's simple comparison predicate, IN predicate, and LIKE predicate are supported, for single-valued
3355 properties only (so that SQL's semantics is preserved). Boolean conjunction (AND), disjunction (OR), and
3356 negation (NOT) of predicates are also supported.

3357 Repositories SHOULD support the comparisons for the property types as described in the list below.
3358 Repositories MAY support additional comparisons and operators. Any additional operators not specified
3359 are repository-specific:

3360
3361 `<Property Type>`

3362 Supported Operators: `<List of Operators supported on Type>`

3363 Supported Literal: `<Supported type of Literal in comparison>`

3364
3365 String (Single)

3366 Supported Operators: `=`, `<>`, `[NOT] LIKE`

3367 Supported Literal: String

3368
3369 String (IN)

3370 Supported Operators: `[NOT] IN`

3371 Supported Literal: List of Strings

3372
3373 Decimal

3374 Supported Operators: `=`, `<>`, `<`, `<=`, `>`, `>=`

3375 Supported Literal: Decimal

3376
[cmis-v1.0-errata-1](#) cmis-spec-v1.0
Copyright © OASIS® 2010. All Rights Reserved.

3377 Decimal (IN)
3378 Supported Operators: [NOT] IN
3379 Supported Literal: List of Decimal
3380
3381 Integer
3382 Supported Operators: =, <>, <, <=, >, >=
3383 Supported Literal: Integer
3384
3385 Integer (IN)
3386 Supported Operators: [NOT] IN
3387 Supported Literal: List of Integer
3388
3389 Boolean
3390 Supported Operators: =
3391 Supported Literal: <boolean literal>
3392
3393 DateTime
3394 Supported Operators: =, <>, <*, <=*, >*, >=*
3395 Supported Literal: <datetime literal>
3396 * - comparison is based on chronological before or after date.
3397
3398 DateTime (IN)
3399 Supported Operators: [NOT] IN
3400 Supported Literal: List of <datetime literal>!
3401
3402 ID
3403 Supported Operators: =, <>
3404 Supported Literal: String
3405
3406 ID (IN)
3407 Supported Operators: [NOT] IN
3408 Supported Literal: List of strings
3409
3410 URI
3411 Supported Operators: =, <>
3412 Supported Literal: String
3413
3414 URI (IN)
3415 Supported Operators: [NOT] IN
3416 Supported Literal: List of strings
3417
3418 URI

3419 Supported Operators: [NOT] LIKE

3420 Supported Literal: String

3421

3422 Operations on the SCORE() output MUST be treated the same as decimal operations.

3423

3424 When using properties in a join statement, comparison MUST be allowed on properties of the same types
3425 as defined by the table above. Repositories MAY extend this behavior.

3426

3427 The ANY operation argument MUST be one of the properties found in the table above which supports
3428 equality operations

3429 **2.1.10.2.4.2 Multi-valued property support (SQL-92 Extension)**

3430 The CMIS query language includes several new non-terminals to expose semantics for querying multi-
3431 valued properties, in a way that does not alter the semantics of existing SQL-92 production rules.

3432 **2.1.10.2.4.2.1 Multi-valued column references**

3433 **BNF grammar structure:** <Multi-valued-column reference>, <multi-valued-column name>

- 3434 • These are non-terminals defined for multi-valued properties whereas SQL-92's <column reference>
3435 and <column name> are retained for single-valued properties only. This is to preserve the single-
3436 value semantics of a regular `"column"` in the SQL-92 grammar.

3437 **2.1.10.2.4.2.2 <Quantified comparison predicate>**

3438 The SQL-92 production rule for <quantified comparison predicate> is extended to accept a multi-valued
3439 property in place of a <table subquery>. This operation is restricted to equality tests only.

3440

3441 <Table subquery> is not supported in CMIS-SQL.

3442

3443 The SQL-92 <quantifier> is restricted to ANY only.

3444

3445 The SQL-92 <row value constructor> is restricted to a literal only.

3446 **Example:**

```
3447 SELECT      Y.CLAIM_NUM, X.PROPERTY_ADDRESS, Y.DAMAGE_ESTIMATES  
3448 | FROM (POLICY AS X JOIN CLAIMS AS Y ON (X.POLICY_NUM = Y.POLICY_NUM )  
3449 | WHERE      ( 100000 = ANY Y.DAMAGE_ESTIMATES )
```

3450 *(Note: DAMAGE_ESTIMATES is a multi-valued Integer property.)*

3451 **2.1.10.2.4.2.3 IN/ANY Predicate**

3452 **BNF grammar structure:** <Quantified in predicate>

3453

3454 CMIS-SQL exposes a new IN predicate defined for a multi-valued property. It is modeled after the SQL-
3455 92 IN predicate, but since the entire predicate is different semantically, it has its own production rule in
3456 the BNF grammar below.

3457

3458 The quantifier is restricted to ANY. The predicate MUST be evaluated to TRUE if at least one of the
3459 | property's values is (or, is not, if NOT is specified) among the given list of literal values. Otherwise the
3460 predicate is evaluated to FALSE.

3461

3506 The ANY operation argument MUST be one of the properties found in the comparison list above which
3507 supports IN operations.

3508 **Example:**

```
3509         SELECT      *  
3510         FROM    CAR_REVIEW  
3511         WHERE      (MAKE = "buick") OR  
3512                   ( ANY FEATURES IN ("NAVIGATION SYSTEM", "SATELLITE RADIO", "MP3")  
3513                   ) (Note: FEATURES is a multi-valued String property.)
```

3514 2.1.10.2.4.3 CONTAINS() predicate function (CMIS-SQL Extension)

3515 **BNF grammar structure::** CONTAINS ([<qualifier> ,] " <text search expression> ")

3516 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
3517 by a Repository (See previous section.)

3518 **Inputs:**

3519 <Qualifier>

3520 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
3521 FROM clause for the query.

- 3522 | ● If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
3523 | Table, but a repository MAY ignore the value of the parameter.
- 3524 | ● If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
3525 | throw an exception if the qualifier is not specified.

3526 <Text Search Expression>

3527 The <text search expression> parameter MUST be a character string , specifying the full-text
3528 search criteria.

3530 | The Text Search Expression may be a set of terms or phrases with an optional " to signal
3531 | negation. A phrase is defined as a word or group of words. A group of words must be
3532 | surrounded by quotes to be considered a single phrase.

3534 | Terms separated by whitespace are AND_{ed} together.

3535 | Terms separated by "OR" are OR_{ed} together

3536 | Implicit "AND" has higher precedence than "OR"

3537 | Within a word or phrase, each (single-)quote must also be escaped by a preceding backslash "\"

3538 **Return value:**

3539 The predicate returns a Boolean value.

3540 | The predicate MUST return TRUE if the object is considered by the repository as "relevant" with
3541 | respect to the given <text search expression> parameter.

3542 | The predicate MUST return FALSE if the object is considered by the repository as not "relevant"
3543 | with respect to the given <text search expression> parameter.

3544 **Constraints:**

3545 At most one CONTAINS() function MUST be included in a single query statement. The repository
3546 MUST throw an exception if more than one CONTAINS() function is found.

3547 The return value of the CONTAINS() function MAY only be included conjunctively (ANDed) with the
3548 aggregate of all other predicates, if there is any, in the WHERE clause.

Formatted
Level: 1 +
0.64" + Ir

Formatted

3550 **2.1.10.2.4.4 SCORE() predicate function**

3551 **BNF grammar structure:** SCORE ()

3552 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
3553 by a Repository (See [previous section](#).)

3554 **Inputs:** No inputs MUST be provided for this predicate function.

3555 **Return value:**

3556 The SCORE() predicate function returns a decimal value in the interval [0,1] .

3557 A repository MUST return the value 0 if the object is considered by the repository as having
3558 absolutely no relevance with respect to the CONTAINS() function specified in the query.

3559 A repository MUST return the value 1 if the object is considered by the repository as having
3560 absolutely complete relevance with respect to the CONTAINS() function specified in the query.

3561 **Constraints:**

3562 The SCORE() function MUST only be used in queries that also include a CONTAINS() predicate
3563 function

3564 The SCORE() function MUST only be used in the SELECT clause of a query. It MUST NOT be
3565 used in the WHERE clause or in the ORDER BY clauses.

3566 An alias column name defined for the SCORE() function call in the SELECT clause (i.e.,
3567 `""SELECT SCORE() AS column_name ...""`) may be used in the ORDER BY clause.

3568 If SCORE() is included in the SELECT clause and an alias column name is not provided, then a
3569 query name of SEARCH_SCORE is used for the query output, and the property definition ID is
3570 repository-specific.

3571 **2.1.10.2.4.5 IN_FOLDER() predicate function**

3572 **BNF grammar structure:** IN_FOLDER([<qualifier>,] <folder id>)

3573 **Usage:** This is a predicate function that tests whether or not a candidate object is a child-object of the
3574 folder object identified by the given <folder id>.

3575 **Inputs:**

3576 **<qualifier>**

3577 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
3578 FROM clause for the query.

- 3579
- 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.
 - If not specified, applies to the single virtual table. If the query is a join, a server SHOULD throw an exception if the qualifier is not specified.
- 3581
- 3582

3583 **<folder id>**

3584 The value of this parameter MUST be the ID of a folder object in the repository.

3585 **Return value:**

3586 The predicate function MUST return TRUE if the object is a child-object of the folder specified by
3587 <folder id>.

3588 The predicate function MUST return FALSE if the object is a NOT a child-object of the folder
3589 specified by <folder id>.

3634 2.1.10.2.4.6 IN_TREE() predicate function

3635 **BNF grammar structure:** IN_TREE([<qualifier>,] <folder id>)

3636 **Usage:** This is a predicate function that tests whether or not a candidate object is a descendant-object of
3637 the folder object identified by the given <folder id>.

3638 **Inputs:**

3639 **<qualifier>**

3640 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
3641 FROM clause for the query.

3642 |

- 3643 | • o If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
3644 | Table, but a repository MAY ignore the value of the parameter.

3645 |

- 3646 | • o If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
3647 | throw an exception if the qualifier is not specified.

3646 **<folder id>**

3647 The value of this parameter MUST be the ID of a folder object in the repository.

3648 **Return value:**

3649 The predicate function MUST return TRUE if the object is a descendant-object of the folder
3650 specified by <folder id>.

3651 The predicate function MUST return FALSE if the object is a NOT a descendant -object of the
3652 folder specified by <folder id>.

3653 2.1.10.2.5 ORDER BY Clause

3654 This clause MUST contain a comma separated list of one or more column names.

3655 | All column names referenced in this clause MUST be valid `"_queryName_"` or their aliased values for
3656 properties defined as *orderable* in the Object-type(s) whose Virtual Tables are listed in the FROM clause.

3657 Only columns in the SELECT clause MAY be in the ORDER BY clause.

3658 Collation rules for the ORDER BY clause are repository specific.

3659 2.1.10.3 Escaping

3660 | Character escaping for character strings differs from SQL-92's escaping. A repository MUST support the
3661 escaping of certain literal characters in a character string, or in a text expression, using a backslash
3662 character (\) in the following manner. For a <character string literal>, which MUST BE a string enclosed
3663 in single-quotes according to the SQL-92 grammar, any occurrence of the single-quote character (') and
3664 the escape character (\) in the string MUST BE escaped. This applies to <folder id>, which is a <character
3665 string literal>. Furthermore, when a <character string literal> is used in a LIKE predicate, any occurrence
3666 of the percent character (%) and the underscore character (_) in the string as a literal MUST BE escaped
3667 also. Therefore, within a quoted string in a query:

3668 |

- 3669 | • The double character \' represents a literal single-quote (') character.

3670 |

- 3671 | • The double character \\ represents a literal backslash (\) character.

3672 |

- 3673 | • Within a LIKE string, the double characters \% and _ represent a literal percent (%) character
3674 | and a literal underscore (_) character respectively.

3675 |

- 3676 | • All other instances of a backslash (\) character are errors.

3677 | Using double single-quotes (") as a SQL-92 way to escape a literal single-quote (') character SHOULD
3678 | BE supported as an allowable alternative to the double character \.

3679 | For a <text search expression>, a second-level character escaping is required so that the <text search
3680 | expression> sub-grammar is *isolatable* from the query statement-level grammar. When a text search

3727 [expression is composed for a query according to the <text search expression> sub-grammar, any](#)
3728 [occurrence of the following three characters in the expression as a literal character MUST BE escaped:](#)
3729 [hyphen \(-\), single-quote \('\), and the escape character \(\\). Then, before this expression is enclosed in](#)
3730 [single-quotes and inserted into a CONTAINS\(\) predicate, the query statement-level escaping rules](#)
3731 [described in the above MUST BE applied. This two-level character escaping allows a query statement](#)
3732 [parser, using statement-level escaping rules, to correctly extract a <text search expression> as a](#)
3733 [character string literal independent of the <text search expression> sub-grammar. This extracted <text](#)
3734 [search expression> can then be correctly interpreted by a full-text search parser independent of the](#)
3735 [query-statement grammar, using second-level escaping rules. Since the <text search expression> sub-](#)
3736 [grammar is isolated from the SQL-92 grammar, double single-quotes is not a valid way to escape a literal](#)
3737 [single-quote character for second-level character escaping.](#)

3738
3739 [An <identifier> in a query statement MUST conform to the SQL-92 identifier syntax, and MUST NOT](#)
3740 [require character escaping.](#)

3741
3742 **Example 1:**

3743 [A query statement that contains a full-text search for the literal string "John'sPresentation-Version2" may](#)
3744 [be composed as:](#)

3745 `SELECT ... FROM ... WHERE ... CONTAINS('John\\'sPresentation\\-Version2') ...`

3746 [A query parser extracts from this statement the text search expression "John'sPresentation-Version2" as](#)
3747 [a character string literal, and passes it to a text-search parser, which interprets it as a single-word full-text](#)
3748 [search criteria: *John'sPresentation-Version2*.](#)

3749
3750 **Example 2:**

3751 [A query statement that contains a full-text search for the phrase "Content Management" may be](#)
3752 [composed as:](#)

3753 `SELECT ... FROM ... WHERE ... CONTAINS('\Content Management')`

3754 **2.1.10.3** [A query parser extracts from this statement the text search expression "Content Management"](#)
3755 [as a character string literal, and passes it to a text-search parser, which interprets it as a full-text search](#)
3756 [criteria consisting of a single phrase: *Content Management*. There is no second-level escaping.](#)

3757 [Repositories MUST support the escaping of characters using a backslash \(\\) in the query statement. The](#)
3758 [backslash character \(\\) will be used to escape characters within quoted strings in the query as follows:](#)

3759 [1. \' will represent a single-quote \('\) character](#)

3760 [2. \\ will represent a backslash \(\\) character](#)

3761 [3. Within a LIKE string, \% and _ will represent the literal characters % and _, respectively.](#)

3762 [4. All other instances of a \ are errors.](#)

3763 [4.](#)

3764 [4.](#)

3765 [4.](#)

3766 [4.](#)

3767 [4.](#)

3768 [4.](#)

3769 [4.](#)

3770 [4.](#)

3771 [4.](#)

3772 [4.](#)

3773 [4.](#)


3774 

3775 [4.](#)

Formatte
Formatte

Formatte
Formatte
Formatte

Formatte

3819 | [4.](#)
3820 | [4.](#)
3821 |  [4.](#)
3822 | [4.](#)

3823 | 2.1.11 Change Log

3824 | CMIS provides a “change log” mechanism to allow applications to easily discover the set of changes
3825 | that have occurred to objects stored in the repository since a previous point in time. This change log can
3826 | then be used by applications such as search services that maintain an external index of the repository to
3827 | efficiently determine how to synchronize their index to the current state of the repository (rather than
3828 | having to query for all objects currently in the repository).

3829 | Entries recorded in the change log are referred to below as “change events”.

3830 | Note that change events in the change log MUST be returned in ascending order from the time when the
3831 | change event occurred.

3832 | 2.1.11.1 Completeness of the Change Log

3833 | The Change Log mechanism exposed by a repository MAY be able to return an entry for every change
3834 | ever made to content in the repository, or may only be able to return an entry for all changes made since
3835 | a particular point in time. This “completeness” level of the change log is indicated via the [optional](#)
3836 | [changesIncomplete](#) value found on the [getRepositoryInfo](#) service response

3837 | However, repositories MUST ensure that if an application requests the entire contents of the repository’s
3838 | change log, that the contents of the change log includes ALL changes made to any object in the
3839 | repository *after* the first change listed in the change log. (I.e. repositories MAY truncate events from the
3840 | change log on a “first-in first-out” basis, but not in any other order.)

3841 | A Repository MAY record events such as filing/unfiling/moving of Documents as change events on the
3842 | Documents, their parent Folder(s), or both the Documents and the parent Folders.

3843 | 2.1.11.2 Change Log Token

3844 | The primary index into the change log of a repository is the “change log token”. The change log token is
3845 | an opaque string that uniquely identifies a particular change in the change log.

3846 | 2.1.11.2.1 “Latest Change Token” repository information

3847 | Repositories that support the `changeLogToken` event MUST expose the latest change log token (i.e. the
3848 | change log token corresponding to the most recent change to any object in the repository) as a property
3849 | returned by the `getRepositoryInfo` service.

3850 | This will enable applications to begin “subscribing” to the change log for a repository by discovering
3851 | what change log token they should use on a going-forward basis to discover change events to the
3852 | repository.

3853 | 2.1.11.3 Change Event

3854 | A change event represents a single action that occurred to an object in the repository that affected the
3855 | persisted state of the object.

3856 | A Repository that supports the change log capability MUST expose at least the following information for
3857 | each change object:

- 3858 | • **ID ObjectID:** The ObjectID of the object to which the change occurred
- 3859 | • **Enum ChangeType:** An enumeration that indicates the type of the change. Valid values are:
 - 3860 | ○ `created`: The object was created.
 - 3861 | ○ `updated`: The object was updated.

- 3862 | o deleted: The object was deleted
- 3863 | o security: The access control or security policy for the object were changed.
- 3864 | • **<Properties> properties:** Additionally, for events of changeType `"_updated"`, the repository MAY
- 3865 | optionally include the new values of properties on the object (if any).
- 3866 | Repositories MUST indicate whether they include properties for `"_updated"` change events via the
- 3867 | [optional enumCapabilityChanges](#) `capability`.
- 3868 |

3869 | **2.2 Services**

- 3870 | The Services section of the CMIS specification defines a set of services that are described in a
- 3871 | protocol/binding-agnostic fashion.
- 3872 | Every protocol binding of the CMIS specification MUST implement all of the methods described in this
- 3873 | section or explain why the service is not implemented.
- 3874 | However, the details of how each service & method is implemented will be described in those protocol
- 3875 | binding specifications.

3876 | **2.2.1 Common Service Elements**

- 3877 | The following elements are common across many of the CMIS services.

3878 | **2.2.1.1 Paging**

- 3879 | All of the methods that allow for the retrieval of a collection of CMIS objects support paging of their result
- 3880 | sets except where explicitly stated otherwise. The following pattern is used:

3881 | **Input Parameters:**

- 3882 | • **(optional) Integer maxItems:** This is the maximum number of items to return in a response. The
- 3883 | repository MUST NOT exceed this maximum. Default is repository-specific.
- 3884 | • **(optional) Integer skipCount:** This is the number of potential results that the repository MUST
- 3885 | skip/page over before returning any results. Defaults to 0.

3886 | **Output Parameters:**

- 3887 | • **Boolean hasMoreItems:** TRUE if the Repository contains additional items after those contained in
- 3888 | the response. FALSE otherwise. If TRUE, a request with a larger skipCount or larger maxItems is
- 3889 | expected to return additional results (unless the contents of the repository has changed).
- 3890 | • **Integer numItems:** If the repository knows the total number of items in a result set, the repository
- 3891 | SHOULD include the number here. If the repository does not know the number of items in a result
- 3892 | set, this parameter SHOULD not be set. The value in the parameter MAY NOT be accurate the next
- 3893 | time the client retrieves the result set or the next page in the result set.

- 3894 | If the caller of a method does not specify a value for maxItems, then the Repository MAY select an
- 3895 | appropriate number of items to return, and MUST use the hasMoreItems output parameter to indicate if
- 3896 | any additional results were not returned.

- 3897 | Repositories MAY return a smaller number of items than the specified value for maxItems.

- 3898 | Each binding will express the above in context and may have different mechanisms for communicating
- 3899 | hasMoreItems and numItems.

3900 | **2.2.1.2 Retrieving additional information on objects in CMIS service calls**

- 3901 | Several CMIS services that return object information have the ability to return dependent object
- 3902 | information as part of their response, such as the Allowable Actions for an object, rendition information,
- 3903 | etc.

3947 The CMIS service methods that support returning a result set of objects will include the ability to return
3948 the following object information:

- 3949 • Properties (retrieves a subset instead of additional information)
- 3950 • Relationships
- 3951 • Renditions
- 3952 • ACLs
- 3953 • AllowableActions

3954

3955 This section describes the input parameter & output pattern for those services. All input parameters are
3956 optional.

3957 2.2.1.2.1 Properties

3958 **Description:** All of the methods that allow for the retrieval of properties for CMIS Objects have a
3959 `"Property Filter"` as an optional parameter, which allows the caller to specify a subset of properties for
3960 Objects that MUST be returned by the repository in the output of the method.

3961 **Optional Input Parameter:**

- 3962 • **String filter:** Value indicating which properties for Objects MUST be returned. Values are:
 - 3963 • `Not set`: The set of properties to be returned MUST be determined by the repository.
 - 3964 • **A comma-delimited list of property definition Query Names: The properties listed MUST be**
3965 **returned.**
 - 3966 • `"*"`: All properties MUST be returned for all objects.

3967 ~~Repositories SHOULD return only the properties specified in the property filter if they exist on the object's~~
3968 ~~type definition. If a property is requested by a filter, a property element MUST be returned for that~~
3969 ~~property. A repository MAY return additional properties. If a returned property is in a "not set" state, a~~
3970 ~~value element MUST NOT be returned for that property.~~

3971

3972 If a property filter specifies a property that is `not set`, it MUST be represented as a property element
3973 without a value element.

3974 2.2.1.2.2 Relationships

3975 **Description:** Used to retrieve the relationships in which the object(s) are participating.

3976 **Optional Input Parameter:**

- 3977 • **Enum includeRelationships:** Value indicating what relationships in which the objects returned
3978 participate MUST be returned, if any. Values are:
 - 3979 `none`: No relationships MUST be returned. (Default).
 - 3980 `source`: Only relationships in which the objects returned are the source MUST be
3981 returned.
 - 3982 `target`: Only relationships in which the objects returned are the target MUST be
3983 returned.
 - 3984 `both`: Relationships in which the objects returned are the source or the target MUST be
3985 returned.

3986 **Output Parameter for each object:**

- 3987 • **<Array> Relationships:** A collection of the relationship objects.

3988 2.2.1.2.3 Policies

3989 **Description:** Used to retrieve the policies currently applied to the object(s).

3990 **Optional Input Parameter:**

- 3991 • **Boolean** `includePolicyIds`: If TRUE, then the Repository MUST return the Ids of the policies
3992 applied to the object. Defaults to FALSE.

3993 **Output Parameter or each object:**

- 3994 • **<Array> Policies**: A collection of the policy objects.

3995 **2.2.1.2.4 Renditions**

3996 **Description:** Used to retrieve the renditions of the object(s).

3997 **Optional Input Parameter:**

- 3998 • **String** `renditionFilter`: The Repository MUST return the set of renditions whose kind matches this
3999 filter. See section below for the filter grammar.

- 4000 ○ Defaults to `"cmis:none"`.

4001 **Output Parameter for each object:**

- 4002 • **<Array> Renditions**: The set of renditions.

4003 **2.2.1.2.4.1 Rendition Filter Grammar**

4004 The Rendition Filter grammar is defined as follows:

4005 `<renditionInclusion> ::= <none> | <wildcard> | <termlist>`

4006 `<termlist> ::= <term> | <term> ',', ',' <termlist>`

4007 `<term> ::= <kind> | <mimetype>`

4008 `<kind> ::= <text>`

4009 `<mimetype> ::= <type> '/' '/' <subtype>`

4010 `<type> ::= <text>`

4011 `<subtype> ::= <text> | <wildcard>`

4012 `<text> ::= / * ! ! any char except whitespace * /`

4013 `<wildcard> ::= ' ' * ' '`

4014 `<none> ::= ' ' cmis:none ' '`

4015 An inclusion pattern allows:

- 4016 • **Wildcard** : include all associated Renditions
4017 • **Comma-separated list of Rendition kinds or mimetypes** : include only those Renditions that
4018 match one of the specified kinds or mimetypes
4019 • **cmis:none**: (Default) exclude all associated Renditions

4020 Examples:

- 4021 • `*` (include all Renditions)
4022 • `cmis:thumbnail` (include only Thumbnails)
4023 • `Image/*` (include all image Renditions)
4024 • `application/pdf, application/x-shockwave-flash` (include web ready Renditions)
4025 • `cmis:none` (exclude all Renditions)

4026 **2.2.1.2.5 ACLs**

4027 **Description:** Used to retrieve the ACLs for the object(s) described in the service response.

4028 **Optional Input Parameter:**

- 4029 • **Boolean** `includeACL`: If TRUE, then the Repository MUST return the ACLs for each object in the
4030 result set. Defaults to FALSE.

4031 **Output Parameter for each object:**

- 4032 • **<Array> ACLs**: The list of access control entries of the ACL for the object.

4033 **2.2.1.2.6 Allowable Actions**

4034 **Description:** Used to retrieve the allowable actions for the object(s) described in the service response.

4035 **Optional Input Parameter:**

- 4036 • **Boolean** `includeAllowableActions`: If TRUE, then the Repository MUST return the available
4037 actions for each object in the result set. Defaults to FALSE.

4038 **Output Parameter for each object:**

- 4039 • **<Array> AllowableActions**: [The list of allowable actions for the object. See](#)
4040 [cmisAllowableActionsType in the CMIS schema.](#)

4041 **2.2.1.3 Change Tokens**

4042 | The CMIS base object-type definitions include an opaque string `"_ChangeToken"` property that a
4043 Repository MAY use for optimistic locking and/or concurrency checking to ensure that user updates do
4044 not conflict.

4045 If a Repository provides values for the ChangeToken property for an Object, then all invocations of the
4046 `"_update"` methods on that object (`updateProperties`, `setContentStream`, `deleteContentStream`) MUST
4047 provide the value of the `changeToken` property as an input parameter, and the Repository MUST throw
4048 an `updateConflictException` if the value specified for the `changeToken` does NOT match the
4049 `changeToken` value for the object being updated.

4050 **2.2.1.4 Exceptions**

4051 The following sections list the complete set of exceptions that MAY be returned by a repository in
4052 response to a CMIS service method call.

4053 **2.2.1.4.1 General Exceptions**

4054 The following exceptions MAY be returned by a repository in response to ANY CMIS service method call.

4055 | The `"_Cause"` field indicates the circumstances under which a repository SHOULD return a particular
4056 exception.

4057 **`invalidArgument`**

4058 Cause: One or more of the input parameters to the service method is missing or invalid.

4059

4060 **`objectNotFound`**

4061 Cause: The service call has specified an object that does not exist in the Repository.

4062

4063 **`notSupported`**

4064 Cause: The service method invoked requires [an optional capability](#) not supported by the
4065 repository.

4066

4067 **`permissionDenied`**

4068 Cause: The caller of the service method does not have sufficient permissions to perform the
4069 operation.

4070

4113 **runtime**

4114 Cause: Any other cause not expressible by another CMIS exception.

4115 2.2.1.4.2 Specific Exceptions

4116 The following exceptions MAY be returned by a repository in response to one or more CMIS service
4117 methods calls.

4118 For each exception, the general intent is listed as well as a list of the methods which MAY cause the
4119 exception to be thrown.

4120 **constraint**

4121 Intent: The operation violates a Repository- or Object-level constraint defined in the CMIS
4122 domain model.

4123 Methods:

- 4124 • **Navigation Services:**
- 4125 ○ getObjectParents
- 4126 • **Object Services:**
- 4127 ○ createDocument
- 4128 ○ createDocumentFromSource
- 4129 ○ createFolder
- 4130 ○ createRelationship
- 4131 ○ createPolicy
- 4132 ○ updateProperties
- 4133 ○ moveObject
- 4134 ○ deleteObject
- 4135 ○ setContentStream
- 4136 ○ deleteContentStream
- 4137 • **Multi-filing Services:**
- 4138 ○ addObjectToFolder
- 4139 • **Versioning Services:**
- 4140 ○ checkOut
- 4141 ○ cancelCheckOut
- 4142 ○ checkIn
- 4143 • **Policy Services:**
- 4144 ○ applyPolicy
- 4145 ○ removePolicy
- 4146 • **Change Log Services:**
- 4147 ○ getContentChanges

4148 **contentAlreadyExists**

4149 Intent: The operation attempts to set the content stream for a Document that already has a
4150 content stream without explicitly specifying the `overwriteFlag` parameter.

4151 Methods:

- 4152 • **Object Services:**
- 4153 ○ setContentStream

Formatted
Level: 1 +
0.45" + Ir
0.45"

Formatted
Level: 1 +
0.45" + Ir
0.45"

4198
4199
4200
4201
4202
4203
4204
4205
4206
4207
4208
4209
4210
4211
4212
4213
4214
4215
4216
4217
4218
4219
4220
4221
4222
4223
4224
4225
4226
4227
4228
4229
4230
4231
4232
4233
4234
4235
4236
4237
4238
4239
4240

filterNotValid

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

Methods:

- **Navigation Services:**
 - getDescendants
 - getChildren
 - getFolderParent
 - getObjectParents
 - getCheckedOutDocs
- **Object Services:**
 - getProperties
 - getRenditions
 - getObject
 - getObjectByPath
- **Versioning Services:**
 - getPropertiesOfLatestVersion
 - getAllVersions
- **Policy Services:**
 - getAppliedPolicies

nameConstraintViolation

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

Methods:

- **Object Services:**
 - createDocument
 - createDocumentFromSource
 - createFolder
 - createRelationship
 - createPolicy
 - updateProperties
 - moveObject

storage

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

Methods:

- **Object Services:**
 - createDocument
 - createDocumentFromSource
 - createFolder
 - createRelationship

Formatted
Level: 1 +
0.45" + Ir
0.45"

Formatted
Level: 1 +
0.45" + Ir
0.45"

Formatted
Level: 1 +
0.45" + Ir
0.45"

- 4284 ○ createPolicy
- 4285 ○ updateProperties
- 4286 ○ moveObject
- 4287 ○ setContentStream
- 4288 ○ deleteContentStream

- **Versioning Services:**

- 4290 ○ checkOut
- 4291 ○ checkIn

streamNotSupported

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

Methods:

- **Object Services:**

- 4298 ○ createDocument
- 4299 ○ createDocumentFromSource
- 4300 ○ getContentStream
- 4301 ○ setContentStream

- **Versioning Services:**

- 4304 ○ checkIn

updateConflict

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

Methods:

- **Object Services:**

- 4310 ○ updateProperties
- 4311 ○ moveObject
- 4312 ○ deleteObject
- 4313 ○ deleteTree
- 4314 ○ setContentStream
- 4315 ○ deleteContentStream

- **Versioning Services:**

- 4318 ○ checkOut
- 4319 ○ cancelCheckOut
- 4320 ○ checkIn

versioning

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

Methods:

- **Object Services:**

Formatted
Level: 1 +
0.45" + Ir
0.45"

Formatted
Level: 1 +
0.45" + Ir
0.45"

Formatted
Level: 1 +
0.45" + Ir
0.45"

- 4363 ○ updateProperties
- 4364 ○ moveObject
- 4365 ○ setContentStream
- 4366 ○ deleteContentStream
- 4367 • **Versioning Services:**
- 4368 ○ checkOut
- 4369 ○ cancelCheckOut
- 4370 ○ checkIn

4371 2.2.1.5 ACLs

4372 Those services which allow for the setting of ACLs may take the optional macro cmis:user which allows
4373 the caller to indicate the operation applies to the current authenticated user.

4374 2.2.2 Repository Services

4375 The Repository Services (getRepositories, getRepositoryInfo, getTypeChildren, getTypeDescendants,
4376 getTypeDefinition) are used to discover information about the repository, including information about the
4377 repository and the object-types defined for the repository.

4378 2.2.2.1 getRepositories

4379 **Description:** Returns a list of CMIS repositories available from this CMIS service endpoint.

4380 2.2.2.1.1 Inputs

4381 None.

4382 2.2.2.1.2 Outputs

4383 A list of repository information, with (at least) the following information for each entry:

- 4384 • **ID repositoryId:** The identifier for the Repository.
- 4385 • **String repositoryName:** A display name for the Repository.

4386 2.2.2.1.3 Exceptions Thrown & Conditions

4387 See section 2.2.1.4.1 General Exceptions

4388 2.2.2.2 getRepositoryInfo

4389 **Description:** Returns information about the CMIS repository, the [optional capabilities](#) it supports and its
4390 Access Control information if applicable. .

4391 2.2.2.2.1 Inputs

4392 **Required:**

- 4393 • **ID repositoryId:** The identifier for the Repository.

4394 2.2.2.2.2 Outputs

- 4395 • **ID repositoryId:** The identifier for the Repository.
- 4396 ○ Note: This MUST be the same identifier as the input to the method.
- 4397 • **String repositoryName:** A display name for the Repository.
- 4398 • **String repositoryDescription:** A display description for the Repository.

Formatted
0.24", Bul
+ Tab afte
stops: Not

Formatted
0.24", Bul
+ Tab afte
stops: Not

Formatted
0.24", Bul
+ Tab afte
stops: Not

Formatted
0.24", Bul
+ Tab afte
stops: Not

- 4445 • **String** `vendorName`: A display name for the vendor of the Repository's underlying application.
- 4446 • **String** `productName`: A display name for the Repository's underlying application.
- 4447 • **String** `productVersion`: A display name for the version number of the Repository's
- 4448 underlying application.
- 4449 • **ID** `rootFolderId`: The ID of the Root Folder Object for the Repository.
- 4450 • **<List of capabilities>**: The set of values for the repository-optional capabilities specified in
- 4451 [section 2.1.1.1 Optional Capabilities](#)
- 4452 • **String** `latestChangeLogToken`: The change log token corresponding to the most recent
- 4453 change event for any object in the repository.
- 4454 • **String** `cmisVersionSupported`: A decimal that indicates what version of the CMIS
- 4455 specification this repository supports as specified in 2.1.1.2 Implementation Information.
- 4456 • **URI** `thinClientURI`: A optional repository-specific URI pointing to the repository's web
- 4457 interface.
- 4458 • **Boolean** `changesIncomplete`: Indicates whether or not the repository's change log can return
- 4459 all changes ever made to any object in the repository or only changes made after a particular
- 4460 point in time. Applicable when the repository's optional capability `capabilityChanges` is not
- 4461 `none`.
 - 4462 ○ If `FALSE`, then the change log can return all changes ever made to every object.
 - 4463 ○ If `TRUE`, then the change log includes all changes made since a particular point in time,
 - 4464 but not all changes ever made.
- 4465 • **<List of enum values>** `changesOnType`: Indicates whether changes are available for base
- 4466 types in the repository. Valid values are from `enumBaseObjectTypes`. See [section 2.1.11](#)
- 4467 [Change Log](#).
 - 4468 ○ **cmis:document**
 - 4469 ○ **cmis:folder**
 - 4470 ○ **cmis:policy**
 - 4471 ○ **cmis:relationship**
- 4472 • **Enum** `supportedPermissions`: specifies which types of permissions are supported.
 - 4473 ○ **basic**: indicates that the CMIS Basic permissions are supported.
 - 4474 ○ **repository**: Indicates that repository specific permissions are supported.
 - 4475 ○ **both**: indicates that both CMIS basic permissions and repository specific permissions are
 - 4476 supported.
- 4477 • **Enum** `propagation`: The list of allowed values for `applyACL`, which control how non-direct
- 4478 ACEs are handled by the repository:
 - 4479 ○ **objectonly**: indicates that the repository is able to apply ACEs without changing the
 - 4480 ACLs of other objects – i.e. ACEs are applied, potentially [“breaking”](#) the [“sharing”](#)
 - 4481 [dependency](#) for non-direct ACEs.
 - 4482 ○ **propagate**: indicates that the repository is able to apply ACEs to a given object and
 - 4483 propagate this change to all inheriting objects – i.e. ACEs are applied with the (intended)
 - 4484 side effect to inheriting objects.
 - 4485 ○ **repositorydetermined**: indicates that the repository uses its own mechanisms to
 - 4486 handle non-direct ACEs when applying ACLs.
- 4487 • **<Array> Permission** `permissions`: The list of repository-specific permissions the repository
- 4488 supports for managing ACEs (see [section 2.8 Access Control](#)).
- 4489 • **<Array> PermissionMapping** `mapping`: The list of mappings for the CMIS Basic permissions to
- 4490 allowable actions (see [section 2.8 Access Control](#)).

Formatted
0.25", Bul
+ Tab afte
stops: Not

Formatted
0.25", Bul
+ Tab afte
stops: Not

Formatted
0.25", Bul
+ Tab afte
stops: Not

4529 | • **String** `principalAnonymous`: If set, this field holds the principal who is used for anonymous
4530 | access. This principal can then be passed to the ACL services to specify what permissions
4531 | anonymous users should have.

4532 | • **String** `principalAnyone`: If set, this field holds the principal who is used to indicate any
4533 | authenticated user. This principal can then be passed to the ACL services to specify what
4534 | permissions any authenticated user should have.

4535 | The `cmisRepositoryInfoType` schema describes the markup that will be included in all CMIS protocol
4536 | bindings to implement this service.

4537 | **2.2.2.2.3 Exceptions Thrown & Conditions**

4538 | See section 2.2.1.4.1 General Exceptions

4539 | **2.2.2.3 getTypeChildren**

4540 | **Description:** Returns the list of [Object-Types](#) defined for the Repository that are children of the specified
4541 | Type.

4542 | **2.2.2.3.1 Inputs**

4543 | **Required:**

4544 | • **String** `repositoryId`: The identifier for the Repository.

4545 | **Optional:**

- 4546 | • **String** `typeId`: The typeId of an Object-Type specified in the Repository.
- 4547 | ○ If specified, then the Repository **MUST** return all of child types of the specified type.
- 4548 | ○ If not specified, then the Repository **MUST** return all Base Object-Types.
- 4549 | • **Boolean** `includePropertyDefinitions`: If TRUE, then the Repository **MUST** return the property
4550 | definitions for each Object-Type returned.
- 4551 | ○ If FALSE (default), the Repository **MUST** return only the attributes for each Object-Type.
- 4552 | • **Integer** `maxItems`: See section 2.2.1.1 Paging.
- 4553 | • **Integer** `skipCount`: See section 2.2.1.1 Paging.

4554 | **2.2.2.3.2 Outputs**

4555 | **<Array> Object-Types**: The list of child [Object-Types](#) defined for the given typeId.

4556 | **Boolean** `hasMoreItems`: See section 2.2.1.1 Paging.

4557 | **Optional:**

4558 | **Integer** `numItems`: See section 2.2.1.1 Paging.

4559 | **2.2.2.3.3 Exceptions Thrown & Conditions**

4560 | See section 2.2.1.4.1 General Exceptions

4561 | **2.2.2.4 getTypeDescendants**

4562 | **Description:** Returns the set of descendant [Object-Types](#) defined for the Repository under the specified
4563 | Type.

4564 | **Notes:**

- 4565 | • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 4566 | • The order in which results are returned is repository-specific.

Formatted
Level: 1 +

4602 2.2.2.4.1 Inputs

4603 Required:

- 4604 • **String repositoryId:** The identifier for the Repository.

4605 Optional:

- 4606 • **String typeId:** The typeId of an Object-Type specified in the Repository.
 - 4607 ○ If specified, then the Repository MUST return all descendant types for the specified type.
 - 4608 ○ If not specified, then the Repository MUST return all types and MUST ignore the value of
 - 4609 the depth parameter
- 4610 • **Integer depth:** The number of levels of depth in the type hierarchy from which to return results.
Valid values are:
 - 4612 ○ **1:** Return only types that are children of the type.
 - 4613 ○ **<Integer value greater than 1>:** Return only types that are children of the type and
 - 4614 descendants up to <value> levels deep.
 - 4615 ○ **-1:** Return ALL descendant types at all depth levels in the CMIS hierarchy.
 - 4616 ○ The default value is repository specific and SHOULD be at least 2 or -1.
- 4617 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
- 4618 definitions for each Object-Type returned.
 - 4619 ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

4620 2.2.2.4.2 Outputs

4621 <Array> **Object-Types:** The hierarchy of [Object-Types](#) defined for the Repository.

4622 2.2.2.4.3 Exceptions Thrown & Conditions

4623 See section 2.2.1.4.1 General Exceptions

- 4624 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
- 4625 an invalid depth.

4626 2.2.2.5 getTypeDefinition

4627 **Description:** Gets the definition of the specified Object-Type.Inputs

4628 2.2.2.5.1 Inputs

4629 Required:

- 4630 • **String repositoryId:** The identifier for the Repository.
- 4631 • **String typeId:** The typeId of an Object-Type specified in the Repository.

4632 2.2.2.5.2 Outputs

- 4633 • Object-type including all property definitions. See section 2.1.3.3 (Object-Type Property
- 4634 Definitions) for further details.

4635 2.2.2.5.3 Exceptions Thrown & Conditions

4636 See section 2.2.1.4.1 General Exceptions

4677 2.2.3 Navigation Services

4678 The Navigation Services (getDescendants, getChildren, getFolderParent, getObjectParents,
4679 getCheckedoutDocs), are used to traverse the folder hierarchy in a CMIS **Repository**, and to locate
4680 Documents that are checked out.

4681 2.2.3.1 getChildren

4682 **Description:** Gets the list of child objects contained in the specified folder.

4683 **Notes:**

- 4684 • If the Repository supports the optional `VersionSpecificFiling` capability, then the repository
4685 MUST return the document versions filed in the specified folder.
- 4686 ○ Otherwise, the latest version of the documents MUST be returned.

4687 2.2.3.1.1 Inputs

4688 **Required:**

- 4689 • **ID repositoryId:** The identifier for the Repository.
- 4690 • **ID folderId:** The identifier for the folder.

4691 **Optional:**

- 4692 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 4693 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 4694 • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and
4695 the ascending modifier `ASC` or the descending modifier `DESC` for each query name. A
4696 repository's handling of the orderBy input is repository-specific.
- 4697 • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the
4698 matched object if they exist on the matched object type definition and in the filter.
- 4699 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4700 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4701 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4702 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
4703 child object for use in constructing that object's path.

4704 2.2.3.1.2 Outputs

- 4705 • **<Array> ObjectResults:** A list of the child objects for the specified folder. Each object result
4706 MUST include the following elements if they are requested:
 - 4707 ○ **<Array> Properties:** The list of properties for the object.
 - 4708 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 4709 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 4710 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 4711 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.
- 4712 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4713 **Optional:**

4714 **Integer numItems:** See section 2.2.1.1 Paging.

4715 2.2.3.1.3 Exceptions Thrown & Conditions

- 4716 • See section 2.2.1.4.1 General Exceptions

- 4758 • `filterNotValid`: The Repository MUST throw this exception if the `is_` property [or rendition](#) filter
4759 input parameter is not valid.
- 4760 • `invalidArgument`: if the specified folder is not a folder

4761 2.2.3.2 getDescendants

4762 **Description:** Gets the set of [descendant objects](#) contained in the specified folder or any of its child-
4763 folders.

4764 Notes:

- 4765 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 4766 • The order in which results are returned is repository-specific..
- 4767 • If the Repository supports the optional capability `capabilityVersionSpecificFiling`, then
4768 the repository MUST return the document versions filed in the specified folder or its descendant
4769 folders. Otherwise, the latest version of the documents MUST be returned.
- 4770 • If the Repository supports the optional capability `capabilityMutlifiling` and the same
4771 document is encountered multiple times in the hierarchy, then the repository MUST return that
4772 document each time is encountered.

4773 2.2.3.2.1 Inputs

4774 Required:

- 4775 • **ID repositoryId:** The identifier for the Repository.
- 4776 • **ID folderId:** The identifier for the folder.

4777 Optional:

- 4778 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results. ← **Formatted**
Level: 1 +
4779 Valid values are:
 - 4780 ○ **1:** Return only objects that are children of the folder.
 - 4781 ○ **<Integer value greater than 1>:** Return only objects that are children of the folder and
4782 descendants up to <value> levels deep.
 - 4783 ○ **-1:** Return ALL descendant objects at all depth levels in the CMIS hierarchy.
 - 4784 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 4785 • **String filter:** See section 2.2.1.2.1 Properties.
- 4786 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4787 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4788 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4789 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
4790 child object for use in constructing that object's path.

4791 2.2.3.2.2 Outputs

- 4792 • **<Array> ObjectResults:** A list of the descendant objects for the specified folder. Each object ← **Formatted**
Level: 1 +
4793 result MUST include the following elements if they are requested:
 - 4794 ○ **<Array> Properties:** The list of properties for the object.
 - 4795 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 4796 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 4797 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 4798 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

4838 2.2.3.2.3 Exceptions Thrown & Conditions

4839 See section 2.2.1.4.1 General Exceptions

- 4840 | • `filterNotValid`: The Repository MUST throw this exception if the `is` property `or rendition` filter
- 4841 | input parameter is not valid.
- 4842 | • `invalidArgument`: The Repository MUST throw this exception if the service is invoked with
- 4843 | `"depth = 0"`.
- 4844 | • `invalidArgument`: if the specified folder is not a folder

4845 2.2.3.3 getFolderTree

4846 **Description:** Gets the set of descendant folder objects contained in the specified folder.

4847

4848 **Notes:**

- 4849 | • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 4850 | • The order in which results are returned is repository-specific..

4851 2.2.3.3.1 Inputs

4852 **Required:**

- 4853 | • **ID repositoryId:** The identifier for the Repository.
- 4854 | • **ID folderId:** The identifier for the folder.

4855 **Optional:**

- 4856 | • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results. ← **Formatted**
- 4857 | Valid values are: **Formatted**
- 4858 | o **1:** Return only folders that are children of the folder.
- 4859 | o **<Integer value greater than 1>:** Return only folders that are children of the folder and
- 4860 | descendant folders up to <value> levels deep.
- 4861 | o **-1:** Return ALL descendant folders at all depth levels in the CMIS hierarchy.
- 4862 | o The default value is repository specific and SHOULD be at least 2 or -1
- 4863 | • **String filter:** See section 2.2.1.2.1 Properties.
- 4864 | • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4865 | • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4866 | • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4867 | • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
- 4868 | child object for use in constructing that object's path.

4869 2.2.3.3.2 Outputs

- 4870 | • **<Array> ObjectResults:** A list of the descendant folders for the specified folder. Each object ← **Formatted**
- 4871 | result MUST include the following elements if they are requested: **Formatted**
- 4872 | o **<Array> Properties:** The list of properties for the object.
- 4873 | o **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 4874 | o **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 4875 | o **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4876 | o **String pathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

4914 2.2.3.3 Exceptions Thrown & Conditions

- 4915 • See section 2.2.1.4.1 General Exceptions
- 4916 • `filterNotValid`: The Repository MUST throw this exception if the `eis` property `or rendition` filter
4917 input parameter is not valid.
- 4918 • `invalidArgument`: The Repository MUST throw this exception if the service is invoked with
4919 an invalid depth
- 4920 • `invalidArgument`: if the specified folder is not a folder

4921
4922

4923 2.2.3.4 getFolderParent

4924 **Description:** Gets the parent folder object for the specified folder object.

4925 2.2.3.4.1 Inputs

4926 **Required:**

- 4927 • **ID repositoryId**: The identifier for the Repository.
- 4928 • **ID folderId**: The identifier for the folder.

4929 **Optional:**

- 4930 • **String filter**: See section 2.2.1.2.1 Properties.

4931 2.2.3.4.2 Outputs

- 4932 • **Object**: The parent folder object of the specified folder.

4933 2.2.3.4.3 Exceptions Thrown & Conditions

- 4934 • See section 2.2.1.4.1 General Exceptions
- 4935 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
4936 parameter is not valid.
- 4937 • `invalidArgument`: The Repository MUST throw this exception if the folderId input is the root
4938 folder.

4939 2.2.3.5 getObjectParents

4940 **Description:** Gets the parent folder(s) for the specified non-folder, fileable object.

4941 2.2.3.5.1 Inputs

4942 **Required:**

- 4943 • **ID repositoryId**: The identifier for the Repository.
- 4944 • **ID objectId**: The identifier for the object.

4945 **Optional:**

- 4946 • **String filter**: See section 2.2.1.2.1 Properties
- 4947 • **Enum includeRelationships**: See section 2.2.1.2.2 Relationships.
- 4948 • **String renditionFilter**: See section 2.2.1.2.4 Renditions.
- 4949 • **Boolean includeAllowableActions**: See section 2.2.1.2.6 Allowable Actions.
- 4950 • **Boolean includeRelativePathSegment**: See section 2.1.5.3 Paths.

Formatted
0.25", Bul
+ Tab afte
stops: Not

Formatted
Level: 1 +

Formatted
Level: 1 +

4992 2.2.3.5.2 Outputs

- 4993 • **<Array> ObjectResults:** A list of the parent folder(s) of the specified objects. Empty for unfiled
4994 objects or for the root folder. Each object result MUST include the following elements if they are
4995 requested:
 - 4996 ○ **<Array> Properties:** The list of properties for the object.
 - 4997 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 4998 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 4999 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 5000 ○ **String relativePathSegment:** If includeRelativePathSegment was TRUE. See section
5001 2.1.5.3 Paths.

5002 2.2.3.5.3 Exceptions Thrown & Conditions

- 5003 • See section 2.2.1.4.1 General Exceptions
- 5004 • **constraint:** The Repository MUST throw this exception if this method is invoked on an object
5005 who Object-Type Definition specifies that it is not fileable.
- 5006 • **filterNotValid:** The Repository MUST throw this exception if the [this](#) property [or rendition](#) filter
5007 input parameter is not valid.

5008 2.2.3.6 getCheckedOutDocs

5009 **Description:** Gets the list of documents that are checked out that the user has access to.

5010 2.2.3.6.1 Inputs

5011 **Required:**

- 5012 • **ID repositoryId:** The identifier for the Repository.

5013 **Optional:**

- 5014 • **ID folderId:** The identifier for a folder in the repository from which documents should be returned.
 - 5015 ○ If specified, the Repository MUST only return checked out documents that are child-
5016 objects of the specified folder.
 - 5017 ○ If not specified, the Repository MUST return checked out documents from anywhere in
5018 the repository hierarchy.
- 5019 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 5020 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 5021 • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and
5022 the ascending modifier `"ASC"` or the descending modifier `"DESC"` for each query name. A
5023 repository's handling of the orderBy input is repository-specific.
- 5024 • **String filter:** See section 2.2.1.2.1 Properties.
- 5025 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 5026 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 5027 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

5028 2.2.3.6.2 Outputs

- 5029 • **<Array> ObjectResults:** A list of checked out documents. Each object result MUST include the
5030 following elements if they are requested:
 - 5031 ○ **<Array> Properties:** The list of properties for the object.
 - 5032 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.

- 5074 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 5075 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 5076 ● **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

5077 **Optional:**

- 5078 ● **Integer numItems:** See section 2.2.1.1 Paging.

5080 **2.2.3.6.3 Exceptions Thrown & Conditions**

- 5081 ● See section 2.2.1.4.1 General Exceptions
- 5082 ● `filterNotValid`: The Repository MUST throw this exception if [this-the](#) property [or rendition](#)
- 5083 filter input parameter is not valid.

5084 **2.2.4 Object Services**

5085 CMIS provides ID-based CRUD (**C**reate, **R**etrieve, **U**date, **D**elete), operations on objects in a Repository.

5086 **2.2.4.1 createDocument**

5087 **Description:** Creates a document object of the specified type (given by the `cmis:objectTypeId` property)

5088 in the (optionally) specified location.

5089 **2.2.4.1.1 Inputs**

5090 **Required:**

- 5091 **ID repositoryId:** The identifier for the Repository.
- 5092 **<Array> properties:** The property values that MUST be applied to the newly-created Document
- 5093 Object.

5094 **Optional:**

- 5095 ● **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
 - 5096 created Document Object.
 - 5097 ○ This parameter MUST be specified if the Repository does NOT support the optional
 - 5098 `"unfiling"` capability.
- 5099 ● **<contentStream> contentStream:** The Content Stream that MUST be stored for the newly-
 - 5100 created Document Object. The method of passing the contentStream to the server and the
 - 5101 encoding mechanism will be specified by each specific binding. MUST be required if the type
 - 5102 requires it.
- 5103 ● **Enum versioningState:** An enumeration specifying what the versioning state of the newly-created
 - 5104 object MUST be. If the repository does not support versioning, the repository MUST ignore the
 - 5105 versioningState parameter. Valid values are:
 - 5106 ○ `none`: The document MUST be created as a non-versionable document.
 - 5107 ○ `checkedout`: The document MUST be created in the checked-out state. The checked-
 - 5108 out document MAY be visible to other users.
 - 5109 ○ `major (default)`: The document MUST be created as a major version
 - 5110 ○ `minor`: The document MUST be created as a minor version.
- 5111 ● **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
- 5112 object.
- 5113 ● **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
- 5114 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.

- 5159 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
5160 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
5161 specified.

5162 2.2.4.1.2 Outputs

5163 **ID objectId:** The ID of the newly-created document.

5164 2.2.4.1.3 Exceptions Thrown & Conditions

- 5165 • See section 2.2.1.4.1 General Exceptions
- 5166 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
5167 met:
- 5168 ○ The cmis:objectId property value is not an Object-Type whose baseType is
5169 "Document".
 - 5170 ○ The cmis:objectId property value is NOT in the list of AllowedChildObjectTypes of
5171 the parent-folder specified by folderId.
 - 5172 ○ The value of any of the properties violates the min/max/required/length constraints
5173 specified in the property definition in the Object-Type.
 - 5174 ○ The "contentStreamAllowed" attribute of the Object-Type definition specified by the
5175 cmis:objectId property value is set to "required" and no contentStream input
5176 parameter is provided.
 - 5177 ○ The "versionable" attribute of the Object-Type definition specified by the
5178 cmis:objectId property value is set to FALSE and a value for the versioningState
5179 input parameter is provided that is something other than "none".
 - 5180 ○ The "versionable" attribute of the Object-Type definition specified by the
5181 cmis:objectId property value is set to TRUE and the value for the versioningState
5182 input parameter is provided that is "none".
 - 5183 ○ The "controllablePolicy" attribute of the Object-Type definition specified by the
5184 cmis:objectId property value is set to FALSE and at least one policy is provided.
 - 5185 ○ The "controllableACL" attribute of the Object-Type definition specified by the
5186 cmis:objectId property value is set to FALSE and at least one ACE is provided.
 - 5187 ○ At least one of the permissions is used in an ACE provided which is not supported by the
5188 repository.
- 5189 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository
5190 detects a violation with the given cmis:name property value, the repository MAY throw this
5191 exception or chose a name which does not conflict.
- 5192 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 5193 • **streamNotSupported:** The Repository MUST throw this exception if the
5194 "contentStreamAllowed" attribute of the Object-Type definition specified by the
5195 cmis:objectId property value is set to "not allowed" and a contentStream input parameter is
5196 provided.

5197 2.2.4.2 createDocumentFromSource

5198 **Description:** Creates a document object as a copy of the given source document in the (optionally)
5199 specified location.

5200 2.2.4.2.1 Inputs

5201 **Required:**

5202 • **ID repositoryId:** The identifier for the Repository.

5246 | **ID sourceId:** The identifier for the source document.

5247 | **Optional:**

- 5248 | • **<Array> properties:** The property values that MUST be applied to the Object. This list of
5249 | properties SHOULD only contain properties whose values differ from the source document.
- 5250 | • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
5251 | created Document Object.
 - 5252 | ○ This parameter MUST be specified if the Repository does NOT support the optional
5253 | `"unfiling"` capability.
- 5254 | • **Enum versioningState:** An enumeration specifying what the versioning state of the newly-
5255 | created object MUST be. [If the repository does not support versioning, the repository MUST](#)
5256 | [ignore the versioningState parameter.](#) Valid values are:
 - 5257 | ○ `none`: The document MUST be created as a non-versionable document.
 - 5258 | ○ `checkedout`: The document MUST be created in the checked-out state.
 - 5259 | ○ `major (default)`: The document MUST be created as a major version
 - 5260 | ○ `minor`: The document MUST be created as a minor version.
- 5261 | • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
5262 | object.
- 5263 | • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
5264 | object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 5265 | • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
5266 | Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
5267 | specified.

Formatted
Level: 1 +

Formatted
Level: 1 +

5268 | 2.2.4.2.2 Outputs

5269 | **ID objectId:** The ID of the newly-created document.

5270 | 2.2.4.2.3 Exceptions Thrown & Conditions

- 5271 | • See section 2.2.1.4.1 General Exceptions
- 5272 | • `constraint`: The Repository MUST throw this exception if ANY of the following conditions are
5273 | met:
 - 5274 | ○ The sourceId is not an Object whose baseType is `"Document"`.
 - 5275 | ○ The source document's `cmis:objectId` property value is NOT in the list of
5276 | AllowedChildObjectIds of the parent-folder specified by folderId.
 - 5277 | ○ The `"versionable"` attribute of the Object-Type definition specified by the
5278 | `cmis:objectId` property value is set to FALSE and a value for the versioningState
5279 | input parameter is provided that is something other than `"none"`.
 - 5280 | ○ The `"versionable"` attribute of the Object-Type definition specified by the
5281 | `cmis:objectId` property value is set to TRUE and the value for the versioningState
5282 | input parameter is provided that is `"none"`.
 - 5283 | ○ The `"controllablePolicy"` attribute of the Object-Type definition specified by the
5284 | `cmis:objectId` property value is set to FALSE and at least one policy is provided.
 - 5285 | ○ The `"controllableACL"` attribute of the Object-Type definition specified by the
5286 | `cmis:objectId` property value is set to FALSE and at least one ACE is provided.
 - 5287 | ○ At least one of the permissions is used in an ACE provided which is not supported by the
5288 | repository.

- 5331 • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. If the repository
5332 detects a violation with the given `cmis:name` property value, the repository MAY throw this
5333 exception or chose a name which does not conflict.
- 5334 • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 5335 • `streamNotSupported`: The Repository MUST throw this exception if the
5336 `"_contentStreamAllowed"` attribute of the Object-Type definition specified by the
5337 `cmis:objectTypeId` property value is set to `"_not allowed"` and a `contentStream` input parameter is
5338 provided.

5339 2.2.4.3 createFolder

5340 **Description:** Creates a folder object of the specified type in the specified location.

5341 2.2.4.3.1 Inputs

5342 Required:

- 5343 • **ID repositoryId:** The identifier for the Repository.
- 5344 • **<Array> properties:** The property values that MUST be applied to the newly-created Folder
5345 Object.
- 5346 • **ID folderId:** The identifier for the folder that MUST be the parent folder for the newly-created
5347 Folder Object.

5348 Optional:

- 5349 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Folder object.
- 5350 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Folder object,
5351 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 5352 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
5353 Folder object, either using the ACL from folderId if specified, or being ignored if no folderId is
5354 specified.

5355 2.2.4.3.2 Outputs

- 5356 • **ID objectId:** The ID of the newly-created folder.

5357 2.2.4.3.3 Exceptions Thrown & Conditions

- 5358 • See section 2.2.1.4.1 General Exceptions
- 5359 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
5360 met:
 - 5361 ○ The `cmis:objectTypeId` property value is not an Object-Type whose baseType is
5362 `"_Folder"`.
 - 5363 ○ The value of any of the properties violates the min/max/required/length constraints
5364 specified in the property definition in the Object-Type.
 - 5365 ○ The `cmis:objectTypeId` property value is NOT in the list of AllowedChildObjectTypes of
5366 the parent-folder specified by folderId.
 - 5367 ○ The `"_controllablePolicy"` attribute of the Object-Type definition specified by the
5368 `cmis:objectTypeId` property value is set to FALSE and at least one policy is provided.
 - 5369 ○ The `"_controllableACL"` attribute of the Object-Type definition specified by the
5370 `cmis:objectTypeId` property value is set to FALSE and at least one ACE is provided.
 - 5371 ○ At least one of the permissions is used in an ACE provided which is not supported by the
5372 repository.

- 5416 • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. If the repository
5417 detects a violation with the given `cmis:name` property value, the repository MAY throw this
5418 exception or chose a name which does not conflict.
- 5419 • `storage`: See section 2.2.1.4.2 Specific Exceptions.

5420 2.2.4.4 createRelationship

5421 **Description:** Creates a relationship object of the specified type

5422 2.2.4.4.1 Inputs

5423 **Required:**

- 5424 • **ID repositoryId:** The identifier for the Repository.
- 5425 • **<Array> properties:** The property values that MUST be applied to the newly-created Relationship
5426 Object.

5427 **Optional:**

- 5428 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Replationship
5429 object.
- 5430 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Relationship
5431 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
5432 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
5433 Relationship object, either using the ACL from folderId if specified, or being ignored if no folderId
5434 is specified.

5435 2.2.4.4.2 Outputs

- 5436 • **ID objectId:** The ID of the newly-created relationship.

5437 2.2.4.4.3 Exceptions Thrown & Conditions

- 5438 • See section 2.2.1.4.1 General Exceptions
- 5439 • `constraint`: The Repository MUST throw this exception if ANY of the following conditions are
5440 met:
 - 5441 ○ The `cmis:objectId` property value is not an Object-Type whose baseType is
5442 `"Relationship"`.
 - 5443 ○ The value of any of the properties violates the min/max/required/length constraints
5444 specified in the property definition in the Object-Type.
 - 5445 ○ The `sourceObjectId`'s Object-Type is not in the list of `"allowedSourceTypes"` specified by
5446 the Object-Type definition specified by `cmis:objectId` property value.
 - 5447 ○ The `targetObjectId`'s Object-Type is not in the list of `"allowedTargetTypes"` specified by
5448 the Object-Type definition specified by `cmis:objectId` property value.
 - 5449 ○ The `"controllablePolicy"` attribute of the Object-Type definition specified by the
5450 `cmis:objectId` property value is set to FALSE and at least one policy is provided.
 - 5451 ○ The `"controllableACL"` attribute of the Object-Type definition specified by the
5452 `cmis:objectId` property value is set to FALSE and at least one ACE is provided.
 - 5453 ○ At least one of the permissions is used in an ACE provided which is not supported by the
5454 repository.
- 5455 • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. If the repository
5456 detects a violation with the given `cmis:name` property value, the repository MAY throw this
5457 exception or chose a name which does not conflict.
- 5458 • `storage`: See section 2.2.1.4.2 Specific Exceptions.

Formatted
0.24", Bull
+ Tab afte
stops: Not

5501 2.2.4.5 createPolicy

5502 **Description:** Creates a policy object of the specified type

5503 2.2.4.5.1 Inputs

5504 Required:

- 5505 • **ID repositoryId:** The identifier for the Repository.
- 5506 • **<Array> properties:** The property values that MUST be applied to the newly-created Policy
- 5507 Object.

5508 Optional:

- 5509 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
5510 created Policy Object.
 - 5511 • This parameter MUST be specified if the Repository does NOT support the optional
5512 `"_unfiling"` capability.
- 5513 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Policy object.
- 5514 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Policy object,
5515 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 5516 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
5517 Policy object, either using the ACL from folderId if specified, or being ignored if no folderId is
5518 specified.

5519 2.2.4.5.2 Outputs

- 5520 • **ID objectId:** The ID of the newly-created Policy Object.

5521 2.2.4.5.3 Exceptions Thrown & Conditions

- 5522 • See section 2.2.1.4.1 General Exceptions
- 5523 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
5524 met:
 - 5525 • The `cmis:objectId` property value is not an Object-Type whose `baseType` is
5526 `"Policy"`.
 - 5527 • The value of any of the properties violates the min/max/required/length constraints
5528 specified in the property definition in the Object-Type.
 - 5529 • The `cmis:objectId` property value is NOT in the list of `AllowedChildObjectTypes` of
5530 the parent-folder specified by `folderId`.
 - 5531 • The `"controllablePolicy"` attribute of the Object-Type definition specified by the
5532 `cmis:objectId` property value is set to `FALSE` and at least one policy is provided.
 - 5533 • The `"controllableACL"` attribute of the Object-Type definition specified by the
5534 `cmis:objectId` property value is set to `FALSE` and at least one ACE is provided.
 - 5535 • At least one of the permissions is used in an ACE provided which is not supported by the
5536 repository.
- 5537 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository
5538 detects a violation with the given `cmis:name` property value, the repository MAY throw this
5539 exception or chose a name which does not conflict.
- 5540 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

5541 2.2.4.6 getAllowableActions

5542 **Description:** Gets the list of allowable actions for an Object (see section.2.2.1.2.6 Allowable Actions).

5577 2.2.4.6.1 Inputs

5578 Required:

- 5579 • **ID repositoryId:** The identifier for the Repository.
- 5580 • **ID objectId:** The identifier for the object

5581 2.2.4.6.2 Outputs

- 5582 • **<Array> AllowableActions:** see section 2.2.1.2.6 Allowable Actions.

5583 2.2.4.6.3 Exceptions Thrown & Conditions

5584 See section 2.2.1.4.1 General Exceptions

5585 2.2.4.7 getObject

5586 **Description:** Gets the specified information for the Object.

5587 2.2.4.7.1 Inputs

5588 Required:

- 5589 • **ID repositoryId:** The identifier for the Repository.
- 5590 • **ID objectId:** The identifier for the object

5591 Optional:

- 5592 • **String filter:** See section 2.2.1.2.1 Properties.
- 5593 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 5594 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 5595 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 5596 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 5597 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

5598 2.2.4.7.2 Outputs

- 5599 • **<Array> Properties:** The list of properties for the object.
- 5600 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 5601 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 5602 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 5603 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 5604 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

5605 2.2.4.7.3 Exceptions Thrown & Conditions

5606 See section 2.2.1.4.1 General Exceptions

5607 | **filterNotValid:** The Repository MUST throw this exception if the [eis](#) property [or rendition](#) filter
5608 | input parameter is not valid.

5609 2.2.4.8 getProperties

5610 **Description:** Gets the list of properties for an Object.

Formatted
0.25", Bul
+ Tab afte
stops: Not

Formatted
0.25" + Ir

Formatted
0.25" + Ir

Formatted
0.25" + Ir

5648 2.2.4.8.1 Inputs

5649 Required:

- 5650 • **ID repositoryId:** The identifier for the Repository.
- 5651 • **ID objectId:** The identifier for the object

5652 Optional:

- 5653 • **String filter:** See section 2.2.1.2.1 Properties.

5654 2.2.4.8.2 Outputs

5655 <Array> **Properties:** The list of properties for the object.

5656 2.2.4.8.3 Exceptions Thrown & Conditions

5657 See section 2.2.1.4.1 General Exceptions

5658 `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter
5659 is not valid.

5660 2.2.4.9 getObjectByPath

5661 **Description:** Gets the specified object.

5662 2.2.4.9.1 Inputs

5663 Required:

- 5664 • **ID repositoryId:** The identifier for the Repository.
- 5665 • **String path:** The path to the object. See section 2.1.5.3 Paths.

5666 Optional:

- 5667 • **String filter:** See section 2.2.1.2.1 Properties.
- 5668 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 5669 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 5670 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 5671 • **Boolean includePolicyIds:** See section 2.2.1.2.2 Relationships.
- 5672 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.

5673 2.2.4.9.2 Outputs

- 5674 • <Array> **Properties:** The list of properties for the object.
- 5675 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

5676 2.2.4.9.3 Exceptions Thrown & Conditions

5677 • See section 2.2.1.4.1 General Exceptions

5678 • `filterNotValid`: The Repository MUST throw this exception if the is property or rendition filter
5679 input parameter is not valid.

5680 2.2.4.10 getContentStream

5681 **Description:** Gets the content stream for the specified Document object, or gets a rendition stream for a
5682 specified rendition of a document or folder object.

5683 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,
5684 in a manner appropriate to that protocol.

5722 2.2.4.10.1 Inputs

5723 Required:

- 5724 • **ID repositoryId:** The identifier for the Repository.
- 5725 • **ID objectId:** The identifier for the object

5726 Optional:

- 5727 • **ID streamId:** The identifier for the rendition stream, when used to get a rendition stream. For Documents, if not provided then this method returns the content stream. For Folders, it MUST be provided.

5730 2.2.4.10.2 Outputs

- 5731 • **<Stream> ContentStream:** The specified content stream or rendition stream for the object.

5732 2.2.4.10.3 Exceptions Thrown & Conditions

- 5733 • See section 2.2.1.4.1 General Exceptions
- 5734 `constraint:` The Repository MUST throw this exception if the object specified by objectId does
- 5735 NOT have a content stream or rendition stream.

5736 2.2.4.11 getRenditions

5737 **Description:** Gets the list of associated Renditions for the specified object. Only rendition attributes are returned, not rendition stream.

5738 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream, in a manner appropriate to that protocol.

5741 2.2.4.11.1 Inputs

5742 Required:

- 5743 • **ID repositoryId:** The identifier for the Repository.
- 5744 • **ID objectId:** The identifier for the object

5745 Optional:

- 5746 • **String renditionFilter:** See Section 2.2.1.2.4
- 5747 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 5748 • **Integer skipCount:** See section 2.2.1.1 Paging.

5749 2.2.4.11.2 Outputs

- 5750 • **<Array> Renditions:** The set of renditions available on this object

5751 2.2.4.11.3 Exceptions Thrown & Conditions

- 5752 • See section 2.2.1.4.1 General Exceptions
- 5753 • `notSupported:` The service method requires functionality that is not supported by the repository
- 5754 • `filterNotValid:` The [rendition](#) filter specified is not valid

5756 2.2.4.12 updateProperties

5757 **Description:** Updates properties of the specified object.

5758 **Notes:**

- 5797 | • A Repository MAY automatically create new Document versions as part of an update properties operation. Therefore, the objectId output NEED NOT be identical to the objectId input.
- 5798
- 5799 | • Each CMIS protocol bindings MUST specify whether the updateProperties service MUST always
- 5800 | include all updatable properties, or only those properties whose values are different than the
- 5801 | original value of the object.

5802 2.2.4.12.1 Inputs

5803 Required:

- 5804 | • **ID repositoryId:** The identifier for the Repository.
- 5805 | **ID objectId:** The identifier of the object to be updated.
- 5806 | • **<Array> properties:** The updated property values that MUST be applied to the Object.

5807 Optional:

- 5808 | • **String changeToken:** See section 2.2.1.3 Change Tokens.

5809 2.2.4.12.2 Outputs

- 5810 | • **ID objectId:** The ID of the updated object.
- 5811 | • **String changeToken:** See section 2.2.1.3 Change Tokens.

5812 2.2.4.12.3 Exceptions Thrown & Conditions

- 5813 | • See section 2.2.1.4.1 General Exceptions
- 5814 | • **constraint:** The Repository MUST throw this exception if the value of any of the properties
- 5815 | violates the min/max/required/length constraints specified in the property definition in the Object-
- 5816 | Type.
- 5817 | • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY
- 5818 | throw this exception or chose a name which does not conflict.
- 5819 | • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 5820 | • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 5821 | • **versioning:** The Repository MUST throw this exception if ANY of the following conditions are
- 5822 | met:
 - 5823 | ○ The object is not checked out and ANY of the properties being updated are defined in
 - 5824 | their Object-Type definition have an attribute value of *Updatability* when checked-out.
 - 5825 | ○ Additionally, the repository MAY throw this exception if the object is a non-current
 - 5826 | Document Version.

5827 2.2.4.13 moveObject

5828 **Description:** Moves the specified file-able object from one folder to another.

5829 2.2.4.13.1 Inputs

5830 Required:

- 5831 | • **ID repositoryId:** The identifier for the Repository.
- 5832 | • **ID objectId:** The identifier of the object to be moved.
- 5833 | • **ID targetFolderId:** The folder into which the object is to be moved.
- 5834 | • **ID sourceFolderId:** The folder from which the object is to be moved.

5875 2.2.4.13.2 Outputs

- 5876 • **ID objectId:** The identifier of the object to be moved.

5877 2.2.4.13.3 Exceptions Thrown & Conditions

- 5878 • See section 2.2.1.4.1 General Exceptions
- 5879 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with a
5880 missing sourceFolderId or the sourceFolderId doesn't match the specified object's parent folder
5881 (or one of the parent folders if the repository supports multifiling.).
- 5882 • **constraint:** The Repository MUST throw this exception if the cmis:objectId property value
5883 of the given object is NOT in the list of AllowedChildObjectTypes of the parent-folder specified
5884 by targetFolderId.
- 5885 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY
5886 throw this exception or chose a name which does not conflict.
- 5887 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 5888 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 5889 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
5890 Version.

5891 2.2.4.14 deleteObject

5892 **Description:** Deletes the specified object.

5893 2.2.4.14.1 Inputs

5894 Required:

- 5895 • **ID repositoryId:** The identifier for the Repository.
- 5896 • **ID objectId:** The identifier of the object to be deleted.

5897 Optional:

- 5898 • **Boolean allVersions:** If TRUE (default), then delete all versions of the document. If FALSE,
5899 delete only the document object specified. The Repository MUST ignore the value of this
5900 parameter when this service is invoke on a non-document object or non-versionable document
5901 object.

5902

5903 2.2.4.14.2 Exceptions Thrown & Conditions

- 5904 • See section 2.2.1.4.1 General Exceptions
- 5905 • **constraint:** The Repository MUST throw this exception if the method is invoked on a Folder
5906 object that contains one or more objects.
- 5907 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

5908 2.2.4.15 deleteTree

5909 **Description:** Deletes the specified folder object and all of its child- and descendant-objects.

5910 Notes:

- 5911 • A Repository MAY attempt to delete child- and descendant-objects of the specified folder in any
5912 order.
- 5913 • Any child- or descendant-object that the Repository cannot delete MUST persist in a valid state in
5914 the CMIS domain model.

- 5955
- This is not atomic.
- 5956
- 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.
- 5957
- 5958

5959 2.2.4.15.1 Inputs

5960 Required:

- 5961
- **ID repositoryId:** The identifier for the Repository.
- 5962
- **ID folderId:** The identifier of the folder to be deleted.

5963 Optional:

- 5964
- **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.
- 5965
- 5966
- 5967
- **Enum unfileObjects:** An enumeration specifying how the repository **MUST** process file-able child- or descendant-objects. Valid values are:
- 5968
- `unfile`: Unfile all fileable objects.
- 5969
- `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.
- 5970
- 5971
- `delete` (**default**): Delete all fileable objects.
- 5972
- **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.
- 5973
- If `FALSE` (**default**), then the repository **SHOULD** abort this method when it fails to delete a single child- or descendant-object.
- 5974
- 5975
- 5976
- 5977

Formatted
Level: 2 +

5978 2.2.4.15.2 Outputs

- 5979
- **<Array> ID failedToDelete:** A list of identifiers of objects in the folder tree that were not deleted.

5980 2.2.4.15.3 Exceptions Thrown & Conditions

- 5981
- See section 2.2.1.4.1 General Exceptions
- 5982
- `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.

5983 2.2.4.16 setContentStream

5984 **Description:** Sets the content stream for the specified Document object.

5985 **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.

5986

5987 2.2.4.16.1 Inputs

5988 Required:

- 5989
- **ID repositoryId:** The identifier for the Repository.
- 5990
- **ID objectId:** The identifier for the Document object.
- 5991
- **<contentStream> contentStream:** The Content Stream

5992 Optional:

- 5993
- **Boolean overwriteFlag:** If `TRUE` (**default**), then the Repository **MUST** replace the existing content stream for the object (if any) with the input `contentStream`.
- 5994

Formatted
Level: 1 +

Formatted
Level: 1 +

6033 ○ If FALSE, then the Repository MUST only set the input contentStream for the object if the
6034 object currently does not have a content-stream.

6035 • **String changeToken:** See section 2.2.1.3 Change Tokens.

6036 **2.2.4.16.2 Outputs**

6037 • **ID objectId:** The ID of the document.

6038 • **String changeToken:** See section 2.2.1.3 Change Tokens.

6039 **2.2.4.16.3 Exceptions Thrown & Conditions**

6040 • See section 2.2.1.4.1 General Exceptions

6041 • **contentAlreadyExists:** The Repository MUST throw this exception if the input parameter
6042 **overwriteFlag** is FALSE and the Object already has a content-stream.

6043 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

6044 • **streamNotSupported:** The Repository MUST throw this exception if the
6045 **"_contentStreamAllowed"** attribute of the Object-Type definition specified by the
6046 **cmis:objectId** property value of the given document is set to **"_notallowed"**.

6047 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

6048 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
6049 Version.

6050 **2.2.4.17 deleteContentStream**

6051 **Description:** Deletes the content stream for the specified Document object.

6052 **Notes:** A Repository MAY automatically create new Document versions as part of this service method.
6053 Therefore, the objectId output NEED NOT be identical to the objectId input.

6054 **2.2.4.17.1 Inputs**

6055 **Required:**

6056 • **ID repositoryId:** The identifier for the Repository.

6057 • **ID objectId:** The identifier for the Document object.

6058 **Optional:**

6059 • **String changeToken:** See section 2.2.1.3 Change Tokens.

6060 **2.2.4.17.2 Outputs**

6061 • **ID objectId:** The ID of the Document object.

6062 • **String changeToken:** See section 2.2.1.3 Change Tokens.

6063 **2.2.4.17.3 Exceptions Thrown & Conditions**

6064 • See section 2.2.1.4.1 General Exceptions

6065 • **constraint:** The Repository MUST throw this exception if the Object's Object-Type definition
6066 **"_contentStreamAllowed"** attribute is set to **"_required"**.

6067 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

6068 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

6069 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
6070 Version.

Formatted
Level: 1 +

Formatted
Level: 1 +

6107 **2.2.5 Multi-filing Services**

6108 The Multi-filing services (*addObjectToFolder*, *removeObjectFromFolder*) are supported only if the
6109 repository supports the multifiling or unfileing [optional capabilities](#). The Multi-filing Services are used to
6110 file/un-file objects into/from folders.

6111 This service is NOT used to create or delete objects in the repository.

6112 **2.2.5.1 addObjectToFolder**

6113 **Description:** Adds an existing fileable non-folder object to a folder.

6114 **2.2.5.1.1 Inputs**

6115 **Required:**

- 6116 • **ID repositoryId:** The identifier for the Repository.
- 6117 • **ID objectId:** The identifier of the object.
- 6118 • **ID folderId:** The folder into which the object is to be filed.

6119 **Optional:**

- 6120 • **Boolean allVersions:** Add all versions of the object to the folder if the repository supports
6121 version-specific filing. Defaults to TRUE.

6122 **2.2.5.1.2 Exceptions Thrown & Conditions**

- 6123 • See section 2.2.1.4.1 General Exceptions.
- 6124 • **constraint:** The Repository MUST throw this exception if the `cmis:objectId` property value
6125 of the given object is NOT in the list of `AllowedChildObjectTypes` of the parent-folder specified
6126 by `folderId`.

6127 **2.2.5.2 removeObjectFromFolder**

6128 **Description:** Removes an existing fileable non-folder object from a folder.

6129 **2.2.5.2.1 Inputs**

6130 **Required:**

- 6131 • **ID repositoryId:** The identifier for the Repository.
- 6132 • **ID objectId:** The identifier of the object.

6133 **Optional:**

- 6134 • **ID folderId:** The folder from which the object is to be removed.
 - 6135 ○ If no value is specified, then the Repository MUST remove the object from all folders in which
6136 it is currently filed.

6137 **2.2.5.2.2 Exceptions Thrown & Conditions**

- 6138 • See section 2.2.1.4.1 General Exceptions

6139 **2.2.6 Discovery Services**

6140 The Discovery Services (*query*) are used to search for query-able objects within the Repository.

6141 **2.2.6.1 query**

6142 **Description:** Executes a CMIS query statement against the contents of the Repository.

6185 **2.2.6.1.1 Inputs**

6186 **Required:**

- 6187 • **ID repositoryId:** The identifier for the Repository.
- 6188 • **String statement:** CMIS query to be executed. (See section 2.1.10 Query.)

6189 **Optional:**

- 6190 • **Boolean searchAllVersions:**
 - 6191 ○ If TRUE, then the Repository MUST include latest and non-latest versions of document
 - 6192 objects in the query search scope.
 - 6193 ○ If FALSE (**default**), then the Repository MUST only include latest versions of documents
 - 6194 in the query search scope.
 - 6195 ○ If the Repository does not support the optional `capabilityAllVersionsSearchable`
 - 6196 capability, then this parameter value MUST be set to FALSE.
- 6197 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
 - 6198 ○ Note: For query statements where the SELECT clause contains properties from only one
 - 6199 virtual table reference (i.e. referenced object-type), any value for this enum may be used.
 - 6200 If the SELECT clause contains properties from more than one table, then the value of this
 - 6201 parameter MUST be `"_none_"`.
- 6202 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
 - 6203 ○ If the SELECT clause contains properties from more than one table, then the value of this
 - 6204 parameter MUST not be set.
- 6205 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 6206 ○ Note: For query statements where the SELECT clause contains properties from only one
 - 6207 virtual table reference (i.e. referenced object-type), any value for this parameter may be
 - 6208 used. If the SELECT clause contains properties from more than one table, then the value
 - 6209 of this parameter MUST be `"_FALSE_"`.
- 6210 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 6211 • **Integer skipCount:** See section 2.2.1.1 Paging.

6212 **2.2.6.1.2 Outputs**

- 6213 • **<Array> Object QueryResults:** The set of results for the query. (See section 2.1.10 Query.).
- 6214 Each object result MUST include the following elements if they are requested:
 - 6215 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 6216 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 6217 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 6218 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

6219 **Optional:**

- 6220 • **Integer numItems:** See section 2.2.1.1 Paging.

6221

6222 **2.2.6.1.3 Exceptions Thrown & Conditions**

- 6223 • See section 2.2.1.4.1 General Exceptions
- 6224 • If the select clause includes properties from more than a single type reference, then the
- 6225 repository SHOULD throw an exception if includeRelationships is something other than `"_none_"`
- 6226 or includeAllowableActions is specified as TRUE.

6269 2.2.6.2 getContentChanges

6270 **Description:** Gets a list of content changes. This service is intended to be used by search crawlers or
6271 other applications that need to efficiently understand what has changed in the repository.

6272 Notes:

- 6273 • The content stream is NOT returned for any change event.
- 6274 • The definition of the authority needed to call this service is repository specific.
- 6275 • The latest change log token for a repository can be acquired via the getRepositoryInfo service.

6276 2.2.6.2.1 Inputs

6277 Required:

- 6278 • **ID repositoryId:** The identifier for the Repository.

6279 Optional:

- 6280 • **String changeLogToken:**

- 6281 ○ If specified, then the Repository MUST return the change event corresponding to the
6282 value of the specified change log token as the first result in the output.
- 6283 ○ If not specified, then the Repository MUST return the first change event recorded in the
6284 change log.

- 6285 • **Boolean includeProperties:**

- 6286 ○ If TRUE, then the Repository MUST include the updated property values for `"_updated"`
6287 change events if the repository supports returning property values as specified by
6288 `capabilityChanges`.
- 6289 ○ If FALSE (default), then the Repository MUST NOT include the updated property values
6290 for `"_updated"` change events. The single exception to this is that the `objectId` MUST
6291 always be included.

- 6292 • **Boolean includePolicyIds:**

6293 If TRUE, then the Repository MUST include the IDs of Policies applied to the object referenced in
6294 each change event, if the change event modified the set of policies applied to the object.

6295 If FALSE (default), then the Repository will not include policy information.

- 6296 • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the
6297 matched object if they exist on the matched object type definition and in the filter.
- 6298 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 6299 • **Integer maxItems:** See section 2.2.1.1 Paging.

6300 2.2.6.2.2 Outputs

6301 • **<Array> changeEvents:** A collection of CMIS objects that MUST include the information as
6302 specified in 2.1.11.3. Each result MUST include the following elements if they are requested:

- 6303 ○ **<Array> policyIDs:** The IDs of Policies applied to the object referenced in the change
6304 event.
- 6305 ○ **<Array> ACLs:** The ACLs applied to the object reference in the change event.

6306 • **String latestChangeLogToken:** The change log token corresponding to the last change event in
6307 `changeEvents`.

6308 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

6309 Optional:

- 6310 • **Integer numItems:** See section 2.2.1.1 Paging.

6348 2.2.6.2.3 Exceptions Thrown & Conditions

- 6349 • See section 2.2.1.4.1 General Exceptions
- 6350 • `constraint`: The Repository MUST throw this exception if the event corresponding to the
6351 change log token provided as an input parameter is no longer available in the change log. (E.g.
6352 because the change log was truncated).

6353 2.2.7 Versioning Services

6354 The Versioning services (`checkOut`, `cancelCheckOut`, `getPropertiesOfLatestVersion`, `getAllVersions`,
6355 `deleteAllVersions`) are used to navigate or update a Document Version Series.

6356 2.2.7.1 checkOut

6357 **Description:** Create a private working copy of the document.

6358 2.2.7.1.1 Inputs

6359 **Required:**

- 6360 • **ID repositoryId:** The identifier for the Repository.
- 6361 • **ID objectId:** The identifier of the document version.

6362 2.2.7.1.2 Outputs

- 6363 • **ID objectId:** The identifier for the `"Private Working Copy"` document.
- 6364 • **Boolean contentCopied:** TRUE if the content-stream of the Private Working Copy is a copy of
6365 the contentStream of the Document that was checked out.
- 6366 • **Boolean contentCopied:** FALSE if the content-stream of the Private Working Copy is `"not set"`.

6367 2.2.7.1.3 Exceptions Thrown & Conditions

- 6368 • See section 2.2.1.4.1 General Exceptions
- 6369 • `constraint`: The Repository MUST throw this exception if the Document's Object-Type
6370 definition's `versionable` attribute is FALSE.
- 6371 • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 6372 • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.
- 6373 • `versioning`: The repository MAY throw this exception if the object is a non-current Document
6374 Version.

6375 2.2.7.2 cancelCheckOut

6376 **Description:** Reverses the effect of a check-out. Removes the private working copy of the checked-out
6377 document, allowing other documents in the version series to be checked out again. If the private working
6378 copy has been created by `createDocument`, `cancelCheckOut` MUST delete the created document.

6379 2.2.7.2.1 Inputs

6380 **Required:**

- 6381 • **ID repositoryId:** The identifier for the Repository.
- 6382 • **ID objectId:** The identifier of the Private Working Copy.

6383 2.2.7.2.2 Exceptions Thrown & Conditions

- 6384 • See section 2.2.1.4.1 General Exceptions

- 6427 • **constraint**: The Repository MUST throw this exception if the Document's Object-Type
6428 definition's *versionable* attribute is FALSE.
- 6429 • **updateConflict**: See section 2.2.1.4.2 Specific Exceptions.
- 6430 • **versioning**: The repository MAY throw this exception if the object is a non-current Document
6431 Version.

Formatted
Level: 1 +

6432 2.2.7.3 checkIn

6433 **Description**: Checks-in the Private Working Copy document.

6434 **Notes**:

- 6435 • For repositories that do NOT support the optional `"_capabilityPWCUpdatable"` *capability*, the
6436 *properties* and *contentStream* input parameters MUST be provided on the checkIn method for
6437 updates to happen as part of checkIn.
- 6438 • Each CMIS protocol bindings MUST specify whether the checkin service MUST always include all
6439 updatable properties, or only those properties whose values are different than the original value
6440 of the object.

6441 2.2.7.3.1 Inputs

6442 **Required**:

- 6443 • **ID repositoryId**: The identifier for the Repository.
- 6444 • **ID objectId**: The identifier of the document.

6445 **Optional**:

- 6446 • **Boolean major**: TRUE (**default**) if the checked-in Document Object MUST be a major version.
6447 • FALSE if the checked-in Document Object MUST NOT be a major version.
- 6448 • **<Array> properties**: The property values that MUST be applied to the checked-in Document
6449 Object.
- 6450 • **<contentStream> contentStream**: The Content Stream that MUST be stored for the checked-in
6451 Document Object. The method of passing the contentStream to the server and the encoding
6452 mechanism will be specified by each specific binding.
- 6453 • **String checkinComment**: See section 2.1.9.5 Versioning Properties on Document Objects.
- 6454 • **<Array> policies**: A list of policy IDs that MUST be applied to the newly-created Document
6455 object.
- 6456 • **<Array> ACE addACEs**: A list of ACEs that MUST be added to the newly-created Document
6457 object.
- 6458 • **<Array> ACE removeACEs**: A list of ACEs that MUST be removed from the newly-created
6459 Document object.

Formatted
Level: 2 +

6460 2.2.7.3.2 Outputs

6461 **ID objectId**: The ID of the checked-in document.

6462 2.2.7.3.3 Exceptions Thrown & Conditions

- 6463 • See section 2.2.1.4.1 General Exceptions
- 6464 • **constraint**: The Repository MUST throw this exception if the Document's Object-Type
6465 definition's *versionable* attribute is FALSE.
- 6466 • **storage**: See section 2.2.1.4.2 Specific Exceptions.
- 6467 • **streamNotSupported**: The Repository MUST throw this exception if the
6468 `"_contentStreamAllowed"` attribute of the Object-Type definition specified by the

- 6507 | cmis:objectTypeId property value is set to `"not allowed"` and a `contentStream` input parameter is
6508 | provided.
- 6509 | • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.

6510 | 2.2.7.4 getObjectOfLatestVersion

6511 | **Description:** Get a the latest Document object in the Version Series.

6512 | 2.2.7.4.1 Inputs

6513 | **Required:**

- 6514 | • **ID repositoryId:** The identifier for the Repository.
- 6515 | • **ID objectId:** The identifier for the Version Series.

6516 | **Optional:**

- 6517 | • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major
6518 | version object in the Version Series.
 - 6519 | ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
6520 | major) version object in the Version Series.
- 6521 | • **String filter:** See section 2.2.1.2.1 Properties.
- 6522 | • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 6523 | • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 6524 | • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 6525 | • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 6526 | • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

6527 | 2.2.7.4.2 Outputs

- 6528 | • **<Array> Properties:** The list of properties for the object.
- 6529 | • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 6530 | • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 6531 | • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 6532 | • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 6533 | • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

6534 | 2.2.7.4.3 Exceptions Thrown & Conditions

- 6535 | • See section 2.2.1.4.1 General Exceptions
- 6536 | • `filterNotValid`: The Repository MUST throw this exception if the `is` property or rendition filter
6537 | input parameter is not valid.
- 6538 | • `objectNotFound`: The Repository MUST throw this exception if the input parameter `major` is
6539 | TRUE and the Version Series contains no major versions.

6540 | 2.2.7.5 getPropertiesOfLatestVersion

6541 | **Description:** Get a subset of the properties for the latest Document Object in the Version Series.

6542 | 2.2.7.5.1 Inputs

6543 | **Required:**

- 6544 | • **ID repositoryId:** The identifier for the Repository.

- 6583 • **ID objectId:** The identifier for the Version Series.
- 6584 **Optional:**
- 6585 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major version
6586 object in the Version Series.
- 6587 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
6588 major) version object in the Version Series.
- 6589 • **String filter:** See section 2.2.1.2.1 Properties.

6590 2.2.7.5.2 Outputs

6591 **<Array> Properties:** The list of properties for the object.

6592 2.2.7.5.3 Exceptions Thrown & Conditions

- 6593 • See section 2.2.1.4.1 General Exceptions
- 6594 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
6595 parameter is not valid.
- 6596 • `objectNotFound`: The Repository MUST throw this exception if the input parameter major is
6597 TRUE and the Version Series contains no major versions.

6598 2.2.7.6 getAllVersions

6599 **Description:** Returns the list of all Document Objects in the specified Version Series, sorted by
6600 `cmis:creationDate` descending.

6601 Notes:

- 6602 • The result set for this operation MUST include the Private Working Copy, subject to caller's
6603 access privileges.

6604 2.2.7.6.1 Inputs

6605 Required:

- 6606 • **ID repositoryId:** The identifier for the Repository.
- 6607 • **ID objectId:** The identifier for the Version Series.

6608 Optional:

- 6609 • **String filter:** See section 2.2.1.2.1 Properties.
- 6610 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

6611 2.2.7.6.2 Outputs

- 6612 • **<Array> ObjectResults:** A list of Document Objects in the specified Version Series. Each object
6613 result MUST include the following elements if they are requested:
- 6614 ○ **<Array> Properties:** The list of properties for the object.
- 6615 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 6616

6617 2.2.7.6.3 Exceptions Thrown & Conditions

- 6618 • See section 2.2.1.4.1 General Exceptions
- 6619 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
6620 parameter is not valid.

6663 2.2.8 Relationship Services

6664 The Relationship Services (*getObjectRelationships*) are used to retrieve the dependent Relationship
6665 objects associated with an independent object.

6666 2.2.8.1 getObjectRelationships

6667 **Description:** Gets all or a subset of relationships associated with an independent object.

6668 2.2.8.1.1 Inputs

6669 **Required:**

- 6670 • **ID repositoryId:** The identifier for the Repository.
- 6671 • **ID objectId:** The identifier of the object.

6672

6673 **Optional:**

- 6674 • **Boolean includeSubRelationshipTypes:** If TRUE, then the Repository MUST return all
6675 relationships whose Object-Types are descendant-types of the given object's `cmis:objectTypeId`
6676 property value as well as relationships of the specified type.
 - 6677 ○ Default is FALSE
 - 6678 ○ If FALSE, then the Repository MUST only return relationships whose Object-Type is
6679 equivalent to the given object's `cmis:objectTypeId` property value.

- 6680 • **Enum relationshipDirection:** An enumeration specifying whether the Repository MUST
6681 return relationships where the specified Object is the source of the relationship, the target of the
6682 relationship, or both. Valid values are:

- 6683 ○ **source: (default)** The Repository MUST return only relationship objects where the specified
6684 object is the source object.
- 6685 ○ **target:** The Repository MUST return only relationship objects where the specified object is
6686 the target object.
- 6687 ○ **either:** The Repository MUST return relationship objects where the specified object is
6688 either the source or the target object.

- 6689 • **ID typeId:** If specified, then the Repository MUST return only relationships whose Object-Type is
6690 of the type specified

- 6691 ○ If not specified, then the repository MUST return Relationship objects of all types.

- 6692 • **Integer maxItems:** See section 2.2.1.1 Paging.

- 6693 • **Integer skipCount:** See section 2.2.1.1 Paging.

- 6694 • **String filter:** See section 2.2.1.2.1 Properties.

- 6695 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

6696 2.2.8.1.2 Outputs

- 6697 • **<Array> Objects:** A list of the relationship objects. Each object result MUST include the following
6698 elements if they are requested:

- 6699 ○ **<Array> Properties:** The list of properties for the object.
- 6700 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

- 6701 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

6702 **Optional:**

- 6703 • **Integer numItems:** See section 2.2.1.1 Paging.

6704

6740 2.2.8.1.3 Exceptions Thrown & Conditions

- 6741 • See section 2.2.1.4.1 General Exceptions
- 6742 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
6743 parameter is not valid.

Formatted
Level: 1 +

6744 2.2.9 Policy Services

6745 The Policy Services (*applyPolicy*, *removePolicy*, *getAppliedPolicies*) are used to apply or remove a policy
6746 object to a *controllablePolicy* object.

6747 2.2.9.1 applyPolicy

6748 **Description:** Applies a specified policy to an object.

6749 2.2.9.1.1 Inputs

6750 **Required:**

- 6751 • **ID repositoryId**: The identifier for the Repository.
- 6752 • **ID policyId**: The identifier for the Policy to be applied.
- 6753 • **ID objectId**: The identifier of the object.

Formatted
Level: 1 +

6754 2.2.9.1.2 Exceptions Thrown & Conditions

6755 See section 2.2.1.4.1 General Exceptions

6756 `constraint`: The Repository MUST throw this exception if the specified object's Object-Type
6757 definition's attribute for *controllablePolicy* is FALSE.

6758 2.2.9.2 removePolicy

6759 **Description:** Removes a specified policy from an object.

6760 2.2.9.2.1 Inputs

6761 **Required:**

- 6762 • **ID repositoryId**: The identifier for the Repository.
- 6763 • **ID policyId**: The identifier for the Policy to be removed.
- 6764 • **ID objectId**: The identifier of the object.

Formatted
Level: 1 +

6765 2.2.9.2.2 Exceptions Thrown & Conditions

6766 • See section 2.2.1.4.1 General Exceptions

6767 • `constraint`: The Repository MUST throw this exception if the specified object's Object-Type
6768 definition's attribute for *controllablePolicy* is FALSE.

Formatted
Level: 1 +

6769 2.2.9.3 getAppliedPolicies

6770 **Description:** Gets the list of policies currently applied to the specified object.

6771 2.2.9.3.1 Inputs

6772 **Required:**

- 6773 • **ID repositoryId**: The identifier for the Repository.
- 6774 • **ID objectId**: The identifier of the object.

Formatted
Level: 1 +

6811 **Optional:**
6812 **String filter:** See section 2.2.1.2.1 Properties.

6813 **2.2.9.3.2 Outputs**

6814 **<Array> Objects:** A list of Policy Objects.

6815 **2.2.9.3.3 Exceptions Thrown & Conditions**

- 6816 • See section 2.2.1.4.1 General Exceptions
- 6817 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
6818 parameter is not valid.

Formatted
Level: 1 +

6819 **2.2.10 ACL Services**

6820 **2.2.10.1 getACL**

6821 **Description:** Get the ACL currently applied to the specified document or folder object.

6822 **2.2.10.1.1 Inputs**

6823 **Required:**

- 6824 • **ID repositoryId:** The identifier for the repository.
- 6825 • **ID objectId:** The identifier for the object

6826 **Optional:**

- 6827 • **Boolean onlyBasicPermissions:** See section 2.8 Access Control. The repository SHOULD
6828 make a best effort to fully express the native security applied to the object
 - 6829 ○ **TRUE:** (default value if not provided) indicates that the client requests that the returned
6830 ACL be expressed using only the CMIS Basic permissions.
 - 6831 ○ **FALSE:** indicates that the server may respond using either solely CMIS Basic
6832 permissions, or repository specific permissions or some combination of both.

6833 **2.2.10.1.2 Outputs**

- 6834 • **<Array> AccessControlEntryType:** The list of access control entries of the ACL for the object.

6835 **Optional:**

- 6836 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
6837 i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.

6838 **2.2.10.1.3 Exceptions Thrown & Conditions**

- 6839 • See section 2.2.1.4.1 General Exceptions

6840 **2.2.10.1.4 Notes**

6841 This service MUST be supported by a repository, if `getRepository` returns `capabilityACL=discover` or
6842 `=manage`.

6843 How an ACL for the object is computed is up to the repository. A client MUST NOT assume that the ACEs
6844 from the ACL as returned by this service can be applied via `applyACL`.

6845 **2.2.10.2 applyACL**

6846 **Description:** Adds or removes the given ACEs to or from the ACL of document or folder object.

6885 **2.2.10.2.1 Inputs**

6886 **Required:**

- 6887 • **ID repositoryId:** The identifier for the repository.
- 6888 • **ID objectId:** The identifier for the object

6889 **Optional:**

- 6890 • **<Array> AccessControlEntryType addACEs:** The ACES to be added.
- 6891 • **<Array> AccessControlEntryType removeACEs:** The ACES to be removed.
- 6892 • **Enum ACLPropagation:** Specifies how ACES should be handled:
 - 6893 ○ **objectonly:** ACES must be applied without changing the ACLs of other objects.
 - 6894 | ○ **propagate:** ACES must be applied by propagate the changes to all **"inheriting"**
 - 6895 | objects.
 - 6896 | ○ **repositorydetermined:** **Default value.** Indicates that the client leaves the behavior to
 - 6897 | the repository.

Formatted
Level: 2 +

6898 **2.2.10.2.2 Outputs**

- 6899 • **<Array> AccessControlEntryType:** The list of access control entries of the resulting ACL for the
- 6900 | object

6901 **Optional:**

- 6902 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
- 6903 | i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.
- 6904 | •~~String changeToken: See section 2.2.1.3 Change Tokens.~~

Formatted

6905 **2.2.10.2.3 Exceptions Thrown & Conditions**

- 6906 • See section 2.2.1.4.1 General Exceptions
- 6907 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
- 6908 | met:
 - 6909 | ○ The specified object's Object-Type definition's attribute for *controllableACL* is FALSE.
 - 6910 | ○ The value for *ACLPropagation* does not match the values as returned via
 - 6911 | *getACLCapabilities*.
 - 6912 | ○ At least one of the specified values for *permission* in ANY of the ACES does not match
 - 6913 | ANY of the *permissionNames* as returned by *getACLCapability* and is not a CMIS Basic
 - 6914 | *permission*

6915 **2.2.10.2.4 Notes**

6916 This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=manage*.

6917 How ACES are added or removed to or from the object is up to the repository – with respect to the

6918 | *ACLPropagation* provided by the client. For **"shared"** ACES (e.g. via inheritance), the repository MAY

6919 | merge the ACES provided with the ACES of the ACL already applied to the object (i.e. the ACES provided

6920 | MAY not be completely added or removed from the effective ACL for the object).

6921

6922

6923 **3 Restful AtomPub Binding**

6924 **3.1 Overview**

6925 This binding is based upon the Atom (RFC4287) and Atom Publishing Protocol (RFC5023).
6926 Implementations of CMIS MUST be compliant with RFC4287 and RFC5023.

6927
6928 In this binding, the client interacts with the repository by acquiring the service document. The client will
6929 request the service document by the URI provided by the vendor. The client will then choose a CMIS
6930 collection, and then start accessing the repository by following the references in the returned documents.

6931
6932 This binding consists of a service document specifying at least CMIS service collections, atom collections,
6933 feeds and entry documents. CMIS extends the Atom and AtomPub documents utilizing the Atom and
6934 AtomPub extension mechanism. CMIS also leverages link tags to specify additional resources related to
6935 the requested resource.

6936
6937 When requesting a resource, optional parameters may be specified to change default behavior via query
6938 parameters.

6939 **3.1.1 Namespaces**

6940 This specification uses the following namespaces and prefixes when referring to xml or xml schema
6941 elements in the text or examples:

- 6942 • CMIS-Core: <http://docs.oasis-open.org/ns/cmisis/core/200908/>
 - 6943 ○ Prefix: cmis
- 6944 • CMIS-RestAtom: <http://docs.oasis-open.org/ns/cmisis/restatom/200908/>
 - 6945 ○ Prefix: cmisra
- 6946 • Atom : <http://www.w3.org/2005/Atom>
 - 6947 ○ Prefix: atom
- 6948 • AtomPub: <http://www.w3.org/2007/app>
 - 6949 ○ Prefix: app

6950

6951 **3.1.2 Authentication**

6952 Authentication SHOULD be handled by the transport protocol. Please see AtomPub (RFC5023) section
6953 14.

6954

6955 **3.1.3 Response Formats**

6956 The client can specify, in HTTP the Accept header, which formats are acceptable to the client. With this
6957 mechanism the client can chose which response format the CMIS implementation should respond with.
6958 The CMIS compliant implementation MUST support the appropriate Media Types specified in this
6959 document.

6960 **3.1.4 Optional Arguments**

6961 The binding supports adding optional parameters to CMIS resources to modify the default behavior.
6962 CMIS implementations **MUST** support arguments being specified as HTTP query string parameters.

6963 Names and valid values for HTTP query string parameters are as described in the appropriate CMIS
6964 Service descriptions [see CMIS Domain Model]. Valid values of enumeration types are also represented
6965 in the CMIS Core XML Schema

6966 **3.1.5 Errors and Exceptions**

6967 Exceptions **MUST** be mapped to the appropriate HTTP status code.

6968 Repositories **SHOULD** provide sufficient information in the body of the HTTP response for a user to
6969 determine corrective action.

6970 See Section 3.2.4 HTTP Status Codes for more information.

6971 **3.1.6 Renditions**

6972 Each Rendition included in a CMIS AtomPub response is represented as an Atom link with relationship
6973 alternate.

6974

6975 The following attributes **SHOULD** be included on the link element:

- 6976 • href: URI to the rendition content stream
- 6977 • type: The Media Type of the Rendition
- 6978 • cmisra:renditionKind: The Rendition Kind for the Rendition

6979

6980 The following attributes **MAY** be included

- 6981 • title: The Filename (or name property if object) of Rendition
- 6982 • length: The length of the rendition

6983 **3.1.7 Content Streams**

6984 The content stream for a document **SHOULD** be referenced by the content src attribute as well as the
6985 edit-media link relation.

6986 A CMIS Repository **MAY** use different URIs for both content src attribute and the edit-media link relation
6987 for the same content stream.

6988 The following attributes **SHOULD** be included on the link element:

- 6989 • href: URI to the content stream
- 6990 • type: The Media Type of the content stream

6991 **3.1.8 Paging of Feeds**

6992 For paging, please see the AtomPub RFC. CMIS leverages first, next, previous, and last link relations to
6993 express paging.

6994 If the repository can include the number of items (numItems in CMIS Domain Model) in a feed, then the
6995 repository **SHOULD** include the cmisra:numItems extension element in the feed.

6996 **3.1.9 Services not Exposed**






6997 The following services are not exposed in this binding:

- 6998 • getRenditions: This is exposed as part of getObject
- 6999 • getProperties: This is exposed as part of getObject

- 7042 • createDocumentFromSource: This is not exposed in this binding except as the client saving the
7043 resource and resubmitting it without the cmis:objectId.
- 7044 • Setting ACL on Create or Checkin operations
 - 7045 ○ This is currently not possible with the REST binding. The Create or Checkin operation
7046 must be performed first. Then the dependent resource, ACL, must be retrieved and
7047 updated.
- 7048 • setContentStream: This does not return the new object id and change token as specified by the
7049 domain model. This is not possible without introducing a new HTTP header.
- 7050 • deleteContentStream: This does not return the new object id and change token as specified by
7051 the domain model. This is not possible without introducing a new HTTP header.
- 7052 • checkOut: This does not return whether or not content was copied. This is not possible without
7053 introducing a new HTTP header.

7054 3.1.9.1 removePolicy

7055 This service is exposed from the domain model in the RESTful Atom Binding. However, it is not as
7056 straightforward. To remove a policy from an object, one must do:

- 7057  Get the object.
- 7058  Fetch the policies collection of the object.
- 7059  Walk through the feed and find the policy object where cmis:objectId == policy id to remove.
- 7060  Get the self lin of this policy object.
- 7061  Perform a DELETE on this URL.

7062
7063 This is also the only case in the RESTful Atom Binding where an URI in a collection (policies) is specific
7064 to that collection.

7065 3.2 HTTP

7066 3.2.1 Entity Tag

7067 CMIS changeTokens are represented as Entity Tags and follow HTTP's use of Entity Tags. CMIS server
7068 implementations SHOULD support Entity Tags. ChangeTokens are also provided as properties and
7069 SHOULD be provided when the object is included inside an atom entry or feed.

7070
7071 On updates, perform the following checks (HTTP & CMIS levels):

- 7072
- 7073 1. If If-Match header is sent by client, ETag value is pulled from HTTP header If-Match per
7074 RFC2616. The supplied ETag is compared against the ETag on the server. If the match fails,
7075 then status code 412 is used.
- 7076 2.
7077 If cmis:changeToken property is supplied by the client, compare the supplied and the
7078 cmis:changeToken on the server. If the comparison fails, then return status code 409 per CMIS.
- 7079
- 7080 4-3. If ETag and cmis:changeToken are both specified, the HTTP If-Match check should be performed
7081 first.

7082 3.2.2 HTTP Range

7083 Repositories MAY support HTTP Range requests on Content Streams.

7084 **3.2.3 HTTP OPTIONS Method**

7085 The repository MAY support the HTTP OPTIONS method on all the resources defined in this
7086 specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP
7087 methods specified for that resource in the Allow header.

7088 **3.2.4 HTTP Status Codes**

7089 Please see the HTTP specification for more information on the HTTP status codes. These are provided
7090 as guidance from the HTTP specification. If any conflict arises, the HTTP specification is authoritative.

7091 **3.2.4.1 General CMIS Exceptions**

7092 The following listing defines the HTTP status codes that repositories MUST return for the various common
7093 exceptions defined in CMIS Domain Model.

7094	CMIS Services Exception	HTTP Status Code
7095	invalidArgument	400
7096	objectNotFound	404
7097	permissionDenied	403
7098	notSupported	405
7099	runtime	500
7100	constraint	409
7101	filterNotValid	400
7102	streamNotSupported	403
7103	storage	500
7104	contentAlreadyExists	409
7105	versioning	409
7106	updateConflict	409
7107	nameConstraintViolation	409

7108

7109 **3.2.4.2 Notable HTTP Status Codes**

- 7110 • 415 Unsupported Media Type
 - 7111 | ○ When a document is POST_{ed} to a collection that does not support the media type of the
 - 7112 | document, this status code MUST be returned
- 7113 • 422 Unprocessable Entity (Defined in RFC4918 Section 11.2)
 - 7114 | ○ When a request has been POST_{ed} but cannot be processed, this status code MUST be
 - 7115 | returned

7116

7117 Please see RFC2616 Section 10 for more information.

7118

7119 **3.3 Media Types**

7120 CMIS introduces new media types for:

- 7121 • a CMIS Query document (application/cmismquery+xml)
- 7122 • a CMIS AllowableActions document (application/cmismallowableactions+xml)
- 7123 • an Atom Document (Entry or Feed) with any CMIS Markup (application/cmismatom+xml)

- 7124 • an Atom Feed Document with CMIS Hierarchy extensions (application/cmistree+xml)
- 7125 • a CMIS ACL Document (application/cmisacl+xml)

7126

7127 In addition to those media types specified by CMIS, CMIS also leverages these media types:

- 7128 • AtomPub Service (application/atomsvc+xml)
- 7129 • Atom Entry (application/atom+xml;type=entry)
- 7130 • Atom Feed (application/atom+xml;type=feed)

7131 3.3.1 CMIS Atom

7132 Media Type: application/cmisatom+xml

7133 Starting tag: atom:feed or atom:entry

7134 Type Parameters:

- 7135 • type – the semantics of the type parameter MUST be the same as the media type parameter for
- 7136 atom documents.

7137

7138 This allows clients to differentiate between repositories that require atom media type with CMIS
 7139 extensions (application/cmisatom+xml) for creation and repositories that allow generic atom media type
 7140 without CMIS extensions (application/atom+xml).

7141

7142 This is only used for CMIS repositories to advertise what media types are accepted for adding to a
 7143 collection (e.g., creating resources in a collection). As such CMIS does not require specifying whether an
 7144 atom feed has CMIS markup. It is included to be consistent with the Atom media type.

7145

7146 All feeds and entries from a CMIS repository MUST utilize the atom media type for exposing Atom
 7147 resources. Please see the individual resources for more information on the media type. This provides
 7148 the interoperability with Atom clients.

7149

7150 Example:

7151

```

7152 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7153 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7154 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7155 xmlns:atom="http://www.w3.org/2005/Atom"
7156 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7157 open.org/ns/cmisis/restatom/200908/">
7158   <atom:author>
7159     <atom:name>Al Brown</atom:name>
7160   </atom:author>
7161   <atom:id>urn:uuid:efe0542e-8933-4b3e-93f2-4d1caa3fc2d9</atom:id>
7162   <atom:title type="text">CMIS Example Document</atom:title>
7163   <atom:updated>2010-01-25T10:20:58.318-08:00</atom:updated>
7164   <atom:content type="text">some text</atom:content>
7165   <cmisra:object>
7166     <cmis:properties>
7167       <cmis:propertyId localName="rep-cmisis:objectTypeId"
7168 propertyDefinitionId="cmis:objectTypeId">
7169         <cmis:value>invoice</cmis:value>
7170       </cmis:propertyId>
7171       <cmis:propertyString localName="rep-cmisis:name"
7172 propertyDefinitionId="cmis:name">
7173         <cmis:value>CMIS Example Document</cmis:value>
7174       </cmis:propertyString>

```

```
7175     </cmis:properties>
7176     </cmisra:object>
7177 </atom:entry>
7178
```

7179 3.3.2 CMIS Query

7180 Media Type: application/cmismquery+xml

7181 Starting tag: cmis:query

7182

7183 This document contains the representation of a query to be executed in a CMIS repository.

7184

7185 Example:

```
7186 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7187 <cmis:query xmlns:cmis="http://docs.oasis-open.org/ns/cmism/core/200908/"
7188 xmlns:cmism="http://docs.oasis-open.org/ns/cmism/messaging/200908/"
7189 xmlns:atom="http://www.w3.org/2005/Atom"
7190 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7191 open.org/ns/cmism/restatom/200908/">
7192   <cmis:statement>SELECT * FROM cmis:document</cmis:statement>
7193   <cmis:searchAllVersions>true</cmis:searchAllVersions>
7194   <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
7195   <cmis:includeRelationships>none</cmis:includeRelationships>
7196   <cmis:renditionFilter>*</cmis:renditionFilter>
7197   <cmis:maxItems>50</cmis:maxItems>
7198   <cmis:skipCount>0</cmis:skipCount>
7199 </cmis:query>
```

7200

7201 Please also see the example documents included with the schema.

7202

7203 3.3.3 CMIS Allowable Actions

7204 Media Type: application/cmismallowableactions+xml

7205 Starting tag: cmis:allowableActions

7206

7207 This document contains the representation of the allowable actions the user may perform on the
7208 referenced object.

7209

7210 Example:

```
7211 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7212 <cmis:allowableActions xmlns:cmis="http://docs.oasis-
7213 open.org/ns/cmism/core/200908/" xmlns:cmism="http://docs.oasis-
7214 open.org/ns/cmism/messaging/200908/"
7215 xmlns:atom="http://www.w3.org/2005/Atom"
7216 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7217 open.org/ns/cmism/restatom/200908/">
7218   <cmis:canDeleteObject>true</cmis:canDeleteObject>
7219   <cmis:canUpdateProperties>true</cmis:canUpdateProperties>
7220   <cmis:canGetProperties>true</cmis:canGetProperties>
7221   <cmis:canGetObjectRelationships>true</cmis:canGetObjectRelationships>
7222   <cmis:canGetObjectParents>true</cmis:canGetObjectParents>
7223   <cmis:canMoveObject>true</cmis:canMoveObject>
7224   <cmis:canDeleteContentStream>true</cmis:canDeleteContentStream>
7225   <cmis:canCheckOut>true</cmis:canCheckOut>
7226   <cmis:canCancelCheckOut>true</cmis:canCancelCheckOut>
```

```

7227 <cmis:canCheckIn>true</cmis:canCheckIn>
7228 <cmis:canSetContentStream>true</cmis:canSetContentStream>
7229 <cmis:canGetAllVersions>true</cmis:canGetAllVersions>
7230 <cmis:canAddObjectToFolder>true</cmis:canAddObjectToFolder>
7231 <cmis:canRemoveObjectFromFolder>true</cmis:canRemoveObjectFromFolder>
7232 <cmis:canGetContentStream>true</cmis:canGetContentStream>
7233 <cmis:canApplyPolicy>true</cmis:canApplyPolicy>
7234 <cmis:canGetAppliedPolicies>true</cmis:canGetAppliedPolicies>
7235 <cmis:canRemovePolicy>true</cmis:canRemovePolicy>
7236 <cmis:canCreateDocument>true</cmis:canCreateDocument>
7237 </cmis:allowableActions>

```

7238

7239 Please also see the example documents included with the schema.

7240

7241 3.3.4 CMIS Tree

7242 Media Type: application/cmistree+xml

7243 Starting tag: atom:feed

7244

7245 This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.

7246

7247 Please see Section 3.5.2.1 for more information.

7248

7249 Example:

```

7250 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7251 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7252 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7253 xmlns:atom="http://www.w3.org/2005/Atom"
7254 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7255 open.org/ns/cmisis/restatom/200908/"
7256 <atom:title type="text">Feed for folder1</atom:title>
7257 <atom:author>
7258 <atom:name>Al Brown</atom:name>
7259 <atom:uri>http://www.ibm.com/</atom:uri>
7260 <atom:email>albertcbrown@us.ibm.com</atom:email>
7261 </atom:author>
7262 <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
7263 <atom:id>urn:uuid:4a80905c-f774-4a9e-a57d-bf0dae5a796e</atom:id>
7264 <atom:link type="application/atom+xml;type=feed" rel="self"
7265 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
7266 41e0d92dfc71/3"/>
7267 <atom:link type="application/atomsvc+xml" rel="service"
7268 href="http://cmisexample.oasis-open.org/rep1//service"/>
7269 <atom:link type="application/atom+xml;type=entry" rel="via"
7270 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
7271 41e0d92dfc71"/>
7272 <atom:link type="application/atom+xml;type=feed"
7273 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7274 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
7275 41e0d92dfc71/foldertree"/>
7276 <atom:link type="application/atom+xml;type=feed" rel="down"
7277 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
7278 41e0d92dfc71/children"/>
7279 <atom:link type="application/atom+xml;type=entry" rel="up"
7280 href="http://cmisexample.oasis-open.org/rep1/bb11830c-7d1e-4b0f-9ff2-
7281 af4857c49200"/>
7282 <atom:entry>

```

```

7283     <atom:author>
7284         <atom:name>Al Brown</atom:name>
7285         <atom:uri>http://www.ibm.com/</atom:uri>
7286         <atom:email>albertcbrown@us.ibm.com</atom:email>
7287     </atom:author>
7288     <atom:content src=""http://cmisexample.oasis-open.org/rep1/63a9c18c-
7289 5e31-4590-8462-86d181e345a4""/>
7290     <atom:id>urn:uuid:63a9c18c-5e31-4590-8462-86d181e345a4</atom:id>
7291     <atom:title type=""text"">CMIS Example Folder as Customer
7292 type</atom:title>
7293     <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
7294     <atom:link rel=""self"" href=""http://cmisexample.oasis-
7295 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4""/>
7296     <atom:link rel=""edit"" href=""http://cmisexample.oasis-
7297 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4""/>
7298     <atom:link type=""application/cmismedia+xml;type=allowableActions""
7299 rel=""http://docs.oasis-open.org/ns/cmismedia/link/200908/allowableactions""
7300 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7301 86d181e345a4/allowableactions""/>
7302     <atom:link type=""application/atom+xml;type=entry""
7303 rel=""describedby"" href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-
7304 5e31-4590-8462-86d181e345a4/type""/>
7305     <atom:link type=""application/atomsvc+xml"" rel=""service""
7306 href=""http://cmisexample.oasis-open.org/rep1//service""/>
7307     <atom:published>2010-01-25T10:20:58.536-08:00</atom:published>
7308     <atom:summary type=""html"">HTML summary of Entry 63a9c18c-5e31-4590-
7309 8462-86d181e345a4</atom:summary>
7310     <atom:link type=""application/atom+xml;type=entry"" rel=""up""
7311 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7312 86d181e345a4/up""/>
7313     <atom:link type=""application/atom+xml;type=feed"" rel=""down""
7314 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7315 86d181e345a4/children""/>
7316     <atom:link type=""application/cmistree+xml"" rel=""down""
7317 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7318 86d181e345a4/tree""/>
7319     <atom:link type=""application/atom+xml;type=feed""
7320 rel=""http://docs.oasis-open.org/ns/cmismedia/link/200908/foldertree""
7321 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7322 86d181e345a4/foldertree""/>
7323     <atom:link type=""application/atom+xml;type=feed""
7324 rel=""http://docs.oasis-open.org/ns/cmismedia/link/200908/relationships""
7325 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7326 86d181e345a4/relationships""/>
7327     <atom:link type=""application/atom+xml;type=feed""
7328 rel=""http://docs.oasis-open.org/ns/cmismedia/link/200908/policies""
7329 href=""http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7330 86d181e345a4/policies""/>
7331     <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
7332 open.org/ns/cmismedia/link/200908/acl"" href=""http://cmisexample.oasis-
7333 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4/acl""/>
7334     <cmisra:object>
7335         <cmis:properties>
7336             <cmis:propertyId localName=""rep-cmis:objectId""
7337 propertyDefinitionId=""cmis:objectId"">
7338                 <cmis:value>63a9c18c-5e31-4590-8462-
7339 86d181e345a4</cmis:value>
7340             </cmis:propertyId>
7341         </cmis:properties>
7342     </cmisra:object>
7343     <cmisra:pathSegment>customer</cmisra:pathSegment>
7344     <cmisra:children>
7345     </atom:feed>

```

```

7346 |         <atom:title type="text">CMIS Example Folder as Customer
7347 | type</atom:title>
7348 |         <atom:author>
7349 |             <atom:name>Al Brown</atom:name>
7350 |             <atom:uri>http://www.ibm.com/</atom:uri>
7351 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
7352 |         </atom:author>
7353 |         <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
7354 |         <atom:id>urn:uuid:51b5c0cd-e473-4492-82b3-
7355 | 666fbf913cf0</atom:id>
7356 |         <atom:link type="application/atom+xml;type=feed"
7357 | rel="self" href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-
7358 | 8462-86d181e345a4/3"/>
7359 |         <atom:link type="application/atomsvc+xml" rel="service"
7360 | href="http://cmisexample.oasis-open.org/rep1//service"/>
7361 |         <atom:link type="application/atom+xml;type=entry"
7362 | rel="via" href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-
7363 | 8462-86d181e345a4"/>
7364 |         <atom:link type="application/atom+xml;type=feed"
7365 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7366 | href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
7367 | 86d181e345a4/foldertree"/>
7368 |         <atom:link type="application/atom+xml;type=feed"
7369 | rel="down" href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-
7370 | 8462-86d181e345a4/children"/>
7371 |         <atom:link type="application/atom+xml;type=entry" rel="up"
7372 | href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
7373 | 41e0d92dfc71"/>
7374 |         <atom:entry>
7375 |             <atom:author>
7376 |                 <atom:name>Al Brown</atom:name>
7377 |                 <atom:uri>http://www.ibm.com/</atom:uri>
7378 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
7379 |             </atom:author>
7380 |             <atom:content src="http://cmisexample.oasis-
7381 | open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
7382 |             <atom:id>urn:uuid:20cb7e68-0a7e-46ea-87e0-
7383 | 09fb8d85286e</atom:id>
7384 |             <atom:title type="text">CMIS Example Doc as Invoice
7385 | type</atom:title>
7386 |             <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
7387 |             <atom:link rel="self" href="http://cmisexample.oasis-
7388 | open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
7389 |             <atom:link rel="edit" href="http://cmisexample.oasis-
7390 | open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
7391 |             <atom:link
7392 | type="application/cmisis+xml;type=allowableActions" rel="http://docs.oasis-
7393 | open.org/ns/cmisis/link/200908/allowableactions"
7394 | href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
7395 | 09fb8d85286e/allowableactions"/>
7396 |             <atom:link type="application/atom+xml;type=entry"
7397 | rel="describedby" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-
7398 | 0a7e-46ea-87e0-09fb8d85286e/type"/>
7399 |             <atom:link type="application/atomsvc+xml"
7400 | rel="service" href="http://cmisexample.oasis-open.org/rep1//service"/>
7401 |             <atom:published>2010-01-25T10:20:58.536-
7402 | 08:00</atom:published>
7403 |             <atom:summary type="html">HTML summary of Entry
7404 | 20cb7e68-0a7e-46ea-87e0-09fb8d85286e</atom:summary>
7405 |             <atom:link rel="edit-media"
7406 | href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
7407 | 09fb8d85286e/edit-media"/>

```

```

7408         <atom:link rel="alternate"
7409 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
7410 09fb8d85286e/alternate"/>
7411         <atom:link type="application/atom+xml;type=feed"
7412 rel="up" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-
7413 87e0-09fb8d85286e/parents"/>
7414         <atom:link type="application/atom+xml;type=feed"
7415 rel="version-history" href="http://cmisexample.oasis-
7416 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/allversions"/>
7417         <atom:link type="application/atom+xml;type=entry"
7418 rel="current-version" href="http://cmisexample.oasis-
7419 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/latest"/>
7420         <atom:link type="application/atom+xml;type=feed"
7421 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
7422 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
7423 09fb8d85286e/relationships"/>
7424         <atom:link type="application/atom+xml;type=feed"
7425 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
7426 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
7427 09fb8d85286e/policies"/>
7428         <atom:link type="application/cmisacl+xml"
7429 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/acl"
7430 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
7431 09fb8d85286e/acl"/>
7432         <cmisra:object>
7433           <cmis:properties>
7434             <cmis:propertyId localName="rep-cmis:objectId"
7435 propertyDefinitionId="cmis:objectId">
7436 <cmis:value>20cb7e68-0a7e-46ea-87e0-09fb8d85286e</cmis:value>
7437           </cmis:propertyId>
7438         </cmis:properties>
7439       </cmisra:object>
7440       <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
7441     </atom:entry>
7442   </atom:feed>
7443 </cmisra:children>
7444 </atom:entry>
7445 <atom:entry>
7446   <atom:author>
7447     <atom:name>Al Brown</atom:name>
7448     <atom:uri>http://www.ibm.com/</atom:uri>
7449     <atom:email>albertcbrown@us.ibm.com</atom:email>
7450   </atom:author>
7451   <atom:content src="http://cmisexample.oasis-open.org/rep1/1de1d476-
7452 11fb-47bf-b136-8a8d0b4b030a"/>
7453   <atom:id>urn:uuid:1de1d476-11fb-47bf-b136-8a8d0b4b030a</atom:id>
7454   <atom:title type="text">CMIS Example Doc as Invoice
7455 type</atom:title>
7456   <atom:updated>2010-01-25T10:20:58.568-08:00</atom:updated>
7457   <atom:link rel="self" href="http://cmisexample.oasis-
7458 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a"/>
7459   <atom:link rel="edit" href="http://cmisexample.oasis-
7460 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a"/>
7461   <atom:link type="application/cmisis+xml;type=allowableActions"
7462 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7463 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
7464 8a8d0b4b030a/allowableactions"/>
7465   <atom:link type="application/atom+xml;type=entry"
7466 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/1de1d476-
7467 11fb-47bf-b136-8a8d0b4b030a/type"/>
7468   <atom:link type="application/atomsvc+xml" rel="service"
7469 href="http://cmisexample.oasis-open.org/rep1//service"/>
7470   <atom:published>2010-01-25T10:20:58.568-08:00</atom:published>

```



```

7471 | <atom:summary type="html">HTML summary of Entry 1de1d476-11fb-47bf-
7472 | b136-8a8d0b4b030a</atom:summary>
7473 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
7474 | open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/edit-media"/>
7475 | <atom:link rel="alternate" href="http://cmisexample.oasis-
7476 | open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/alternate"/>
7477 | <atom:link type="application/atom+xml;type=feed" rel="up"
7478 | href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
7479 | 8a8d0b4b030a/parents"/>
7480 | <atom:link type="application/atom+xml;type=feed" rel="version-
7481 | history" href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-
7482 | b136-8a8d0b4b030a/allversions"/>
7483 | <atom:link type="application/atom+xml;type=entry" rel="current-
7484 | version" href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-
7485 | b136-8a8d0b4b030a/latest"/>
7486 | <atom:link type="application/atom+xml;type=feed"
7487 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
7488 | href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
7489 | 8a8d0b4b030a/relationships"/>
7490 | <atom:link type="application/atom+xml;type=feed"
7491 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
7492 | href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
7493 | 8a8d0b4b030a/policies"/>
7494 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7495 | open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
7496 | open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/acl"/>
7497 | <cmisra:object>
7498 | <cmis:properties>
7499 | <cmis:propertyId localName="rep-cmis:objectId"
7500 | propertyDefinitionId="cmis:objectId">
7501 | <cmis:value>1de1d476-11fb-47bf-b136-
7502 | 8a8d0b4b030a</cmis:value>
7503 | </cmis:propertyId>
7504 | </cmis:properties>
7505 | </cmisra:object>
7506 | <cmisra:pathSegment>invoice3.pdf</cmisra:pathSegment>
7507 | </atom:entry>
7508 | </atom:feed>

```

7509

7510 | Note: This media type is used on links with relation down (see section 3.4.3.2 [Hierarchy Navigation](#)

7511 | [Internet Draft Link Relations Hierarchy Navigation Internet Draft Link Relations](#)). When the individual

7512 | resources are returned by the CMIS repository they will use the atom media type (application/atom+xml)

7513

7514 | Please also see the example documents included with the schema.

7515

7516 | 3.3.5 CMIS ACL

7517 | Media Type: application/cmisacl+xml

7518 | Starting tag: cmis:acl

7519

7520 | This document specifies an Access Control List based on the schema in CMIS Domain Model.

7521

7522 | Example:

7523 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```
7524 <cmis:acl xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7525 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7526 xmlns:atom="http://www.w3.org/2005/Atom"
7527 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7528 open.org/ns/cmis/restatom/200908/">
7529   <cmis:permission>
7530     <cmis:principal>
7531       <cmis:principalId>Al Brown</cmis:principalId>
7532     </cmis:principal>
7533     <cmis:permission>cmis:read</cmis:permission>
7534     <cmis:permission>cmis:write</cmis:permission>
7535     <cmis:permission>cmis:all</cmis:permission>
7536     <cmis:permission>publish</cmis:permission>
7537     <cmis:direct>>true</cmis:direct>
7538   </cmis:permission>
7539 </cmis:acl>
```

7540

7541 Please also see the example documents included with the schema.

7542

7543 **3.4 Atom Extensions for CMIS**

7544 **3.4.1 Atom Element Extensions**

7545 **3.4.1.1 AtomPub Workspace**

7546 **3.4.1.1.1 cmisra:collectionType**

7547 This element is included inside the app:collection element. This specifies the cmis collection type.

7548 **3.4.1.1.2 cmisra:repositoryInfo**

7549 This element is included inside the app:workspace element. This specifies information about the CMIS
7550 repository.

7551 **3.4.1.1.3 cmis:uritemplate**

7552 This element is included inside the app:workspace element. This specifies information about URI
7553 templates

7554 **3.4.1.2 Atom Feed**

7555 **3.4.1.2.1 cmisra:numItems**

7556 This element is included inside the atom:feed element. This specifies the number of items in the feed.

7557 **3.4.1.3 Atom Entry**

7558 **3.4.1.3.1 cmisra:children**

7559 This element is included inside the atom:entry element. This includes the children of the atom entry. This
7560 element **MUST** include an atom:feed element.

7561

7562 **3.4.1.3.2 cmisra:object**

7563 This element is included inside the atom:entry element for CMIS Document, Folder, Relationship and
7564 Policy objects. This specifies the CMIS object information for the atom entry.

7565

7566 **3.4.1.3.3 cmisra:pathSegment**

7567 This element is included inside the atom:entry element for CMIS Type Definitions that are filable. This
7568 specifies the pathSegment for this object in the folder representing the feed.

7569

7570 **3.4.1.3.4 cmisra:relativePathSegment**

7571 This element is included inside the atom:entry element. This specifies the relative pathSegment for the
7572 object in that particular folder. This MUST be used only inside an object parents feed.

7573 **3.4.1.3.5 cmisra:type**

7574 This element is included inside the atom:entry element for CMIS Type Definitions. This specifies the type
7575 definition the atom entry represents.

7576 **3.4.1.3.6 cmisra:content**

7577 This element specifies the content of the atom:entry element. The content is base64 encoded in the
7578 base64 element. The elements of a cmisra:content element are:

- 7579 • mediaType: This contains the media type of the content as described by RFC4288.
- 7580 • base64: This contains the base64 content of the file

7581

7582 This element MUST take precedence over atom:content on submission of an atom entry to a repository.

7583

7584 A repository MUST use the atom:content element to return back to the client the content of the document.

7585

7586 Note: This is required when the client has an XML document stored that is might not be well formed and
7587 thus would not be able to be included inside atom:content element.

7588

7589 **3.4.2 Attributes**

7590 These attributes are in the CMIS RestAtom namespace (cmisra).

7591 **3.4.2.1 cmisra:id**

7592 This attribute is used on the atom:link element to specify the cmis id of the resource. This attribute
7593 SHOULD be on all link relations that point to a CMIS object.

7594

7595 This attribute MAY also be on cmisra:type. The value of the attribute on cmis:type MUST be the same as
7596 the type definition id.

7597

7598 Example:

7599 | `<?xml version="1.0" encoding="UTF-8" standalone="yes"?>`

```
7600 | <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7601 | xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7602 | xmlns:atom="http://www.w3.org/2005/Atom"
7603 | xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7604 | open.org/ns/cmisis/restatom/200908/" type="application/atom+xml;type=feed"
7605 | rel="down" href="http://cmisexample.oasis-open.org/rep1//children/e170da7d-
7606 | d322-472d-b1eb-67bdb1ec18ca/1" cmisra:id="e170da7d-d322-472d-b1eb-
7607 | 67bdb1ec18ca"/>
```

7608

7609 Please also see the example documents included with the schema.

7610

7611 3.4.2.2 cmisra:renditionKind

7612 This attribute is used on the atom:link element with relation alternate to specify the renditionKind of the
7613 resource. This attribute SHOULD be on all link elements with relation alternate that are a CMIS rendition.

7614

7615 Example:

```
7616 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7617 | <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7618 | xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7619 | xmlns:atom="http://www.w3.org/2005/Atom"
7620 | xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7621 | open.org/ns/cmisis/restatom/200908/" type="text/html" rel="alternate"
7622 | href="http://cmisexample.oasis-open.org/rep1//rendition/e170da7d-d322-472d-
7623 | b1eb-67bdb1ec18ca/1" cmisra:renditionKind="cmis:thumbnail"/>
```

7624

7625 Please also see the example documents included with the schema.

7626 3.4.3 CMIS Link Relations

7627 The listing below outlines the different link relation types in CMIS. This is in addition to the link relations
7628 specified by Atom and Atom Publishing Protocol. The registry for link relations is located at
7629 <http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

7630

7631 The link element with a specified relation MUST be included if client can perform the operation. The
7632 repository SHOULD omit the link relation if the operation is not available. The operation may not be
7633 available due to a variety of reasons such as access control, administrative policies, or other
7634 mechanisms.

7635

7636 Links may have the following attribute in addition to the ones specified by Atom and Atom Publishing
7637 Protocol:

- 7638 • (CMIS) id: Specifies the CMIS ID of the resource referenced by the link. Repositories SHOULD
7639 include this attribute for elements such as atom:link that point to CMIS resources that have an id.

7640

7641 These are the link relation types specified by CMIS:

7642 3.4.3.1 Existing Link Relations

7643 Existing link relations should be used where appropriate by the implementation. In addition, the following
7644 link relations are leveraged for the CMIS specification:

- 7645 • self
 - 7646 ○ This link relation provides the URI to retrieve this resource again.

- 7647
 - Service: The appropriate service that generated the atom entry or feed.
- 7648
 - Resources: All except AllowableActions, ACL and Content Streams
- 7649
 - service
 - 7650
 - 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
 - 7651
 - represent the repository containing that resource.
 - 7652
 - Media Type: application/atomsvc+xml
 - 7653
 - Resources: All except AllowableActions, ACL and Content Streams
- 7654
 - Resources: All except AllowableActions, ACL and Content Streams
- 7655
 - describedby
 - 7656
 - When used on a CMIS resource, this link relation MUST point to an atom entry that
 - 7657
 - describes the type of that resource.
 - 7658
 - Service: getTypeDefinition on specified object
 - 7659
 - Media Type: application/atom+xml;type=entry
 - 7660
 - Resources: CMIS Document, CMIS Folder, CMIS Relationship, CMIS Policy objects and
 - 7661
 - CMIS Types
- 7662
 - via
 - 7663
 - When used on an Atom Feed document, this link relation MUST point to the atom entry
 - 7664
 - representing the CMIS resource from whom this feed is derived.
 - 7665
 - Media Type: application/atom+xml;type=entry
 - 7666
 - Resources: All CMIS Feeds and Collections
- 7667
 - edit-media
 - 7668
 - When used on a CMIS document resource, this link relation MUST point to the URI for
 - 7669
 - content stream of the CMIS document. This URI MUST be used to set or delete the
 - 7670
 - content stream. This URI MAY be used to retrieve the content stream for the document.
 - 7671
 - Service: setContentStream (PUT) , deleteContentStream (DELETE)
 - 7672
 - Media Type: Specific to resource
 - 7673
 - Resources: CMIS Document
- 7674
 - edit
 - 7675
 - When used on a CMIS resource, this link relation MUST provide an URI that can be used
 - 7676
 - with the HTTP PUT method to modify the atom:entry for the CMIS resource
 - 7677
 - Service: getObject (GET), updateProperties (PUT)
 - 7678
 - Media Type: application/atom+xml;type=entry
 - 7679
 - Resources: CMIS Documents, CMIS Folders, CMIS Relationships and CMIS Policies
- 7680
 - alternate
 - 7681
 - This is used to express Renditions on a CMIS resource. See section 3.1.6 Renditions.
 - 7682
 - Service: getContentStream for specified rendition
 - 7683
 - Resources: CMIS Document, CMIS Folder and CMIS Policies
- 7684
 - first
 - 7685
 - This is used for Paging. Please see the AtomPub specification.
 - 7686
 - Media Type: application/atom+xml;type=feed
 - 7687
 - Resources: All Feeds
- 7688
 - previous
 - 7689
 - This is used for Paging. Please see the AtomPub specification.
 - 7690
 - Media Type: application/atom+xml;type=feed

- 7691 ○ Resources: All Feeds
- 7692 ● next
 - 7693 ○ This is used for Paging. Please see the AtomPub specification.
 - 7694 ○ Media Type: application/atom+xml;type=feed
 - 7695 ○ Resources: All Feeds
- 7696 ● last
 - 7697 ○ This is used for Paging. Please see the AtomPub specification.
 - 7698 ○ Media Type: application/atom+xml;type=feed
 - 7699 ○ Resources: All Feeds

7700
7701

7702 Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on
7703 these link relations.

7704 3.4.3.2 Hierarchy Navigation Internet Draft Link Relations

7705 CMIS leverages the following link relations:

- 7706 ● up
 - 7707 ○ Service: getFolderParent, getObjectParents, getTypeDefinition, getObject
 - 7708 ○ Media Type: application/atom+xml;type=feed, application/atom+xml;type=entry
 - 7709 ○ Resources: CMIS Document, CMIS Folder, CMIS Type Definitions, CMIS Folder
7710 Children, CMIS Folder Descendants, CMIS FolderTree, CMIS Type Children, CMIS Type
7711 Descendants
 - 7712 ■ This link relation is not included on CMIS Base Type Definitions or the CMIS
7713 Root Folder
- 7714 ● down
 - 7715 ○ Service: getChildren, getDescendants, getTypeChildren, getTypeDescendants
 - 7716 ○ Media Type:
 - 7717 ■ For children: application/atom+xml;type=feed
 - 7718 ■ For descendants: application/cmistree+xml
 - 7719 ● The descendants feed resource when retrieved from the CMIS repository
7720 will use the Atom Feed Media Type (application/atom+xml;type=feed)
 - 7721 ○ Resources: CMIS Folder, Type

7722 3.4.3.3 Versioning Internet Draft Link Relations

7723 CMIS leverages the following link relations from the Internet Draft:

- 7724 ● version-history
 - 7725 ○ Service: getAllVersions
 - 7726 ○ Media Type: application/atom+xml;type=feed
 - 7727 ○ Resources: CMIS Document
- 7728 ● current-version
 - 7729 ○ Service: getObjectForLatestVersion
 - 7730 ○ Media Type: application/atom+xml;type=entry
 - 7731 ○ Resources: CMIS Document
- 7732 ● working-copy

- 7733 ○ Service: getObject for private-working-copy specified by
- 7734 cmis:versionSeriesCheckedOutId property
- 7735 ○ Media Type: application/atom+xml;type=entry
- 7736 ○ Resources: CMIS Document

7737 3.4.3.4 CMIS Specific Link Relations

7738 CMIS defines the following link relations:

- 7739 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>
 - 7740 ▪ This link relation MUST point to a resource containing a CMIS AllowableActions
 - 7741 document for the CMIS resource containing this link relation.
 - 7742 ▪ Service: getAllowableActions
 - 7743 ▪ Media Type: application/cmisallowableactions+xml
 - 7744 ▪ Resources: CMIS Documents, CMIS Folders, CMIS Policies, and CMIS
 - 7745 Relationships
- 7746 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>
 - 7747 ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS
 - 7748 relationship resources for the CMIS resource containing this link relation.
 - 7749 ▪ Service: getObjectRelationships
 - 7750 ▪ Media Type: application/atom+xml;type=feed
 - 7751 ▪ Resources: CMIS Documents, CMIS Folders, and CMIS Policies
- 7752 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/source>
 - 7753 ▪ When used on a CMIS Relationship resource, this link relation MUST point to an
 - 7754 atom entry document for the CMIS Resource specified by the cmis:sourceId
 - 7755 property on the relationship.
 - 7756 ▪ Source Link on Relationship
 - 7757 ▪ Media Type: application/atom+xml;type=entry
 - 7758 ▪ Resources: CMIS Relationships
- 7759 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/target>
 - 7760 ▪ When used on a CMIS Relationship resource, this link relation MUST point to an
 - 7761 atom entry document for the CMIS Resource specified by the cmis:targetId
 - 7762 property on the relationship.
 - 7763 ▪ Target Link on Relationship
 - 7764 ▪ Media Type: application/atom+xml;type=entry
 - 7765 ▪ Resources: CMIS Relationships
- 7766 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/policies>
 - 7767 ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS
 - 7768 Policy resources for the CMIS resource containing this link relation.
 - 7769 ▪ Service: getAppliedPolicies
 - 7770 ▪ Media Type: application/atom+xml;type=feed
 - 7771 ▪ Resources: CMIS Documents and CMIS Folders
- 7772 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/acl>
 - 7773 ▪ This link relation MUST point to a resource containing a CMIS ACL document for
 - 7774 the CMIS resource containing this link relation.
 - 7775 ▪ Service: getACL
 - 7776 ▪ Media Type: application/cmisacl+xml

- 7777 ▪ Resources: CMIS Documents, CMIS Folders, CMIS Relationships, and CMIS
- 7778 Policies that are securable
- 7779 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/changes>
- 7780 ▪ This link relation MUST point to an Atom Feed containing the set of changes
- 7781 ▪ Service: getContentChanges
- 7782 ▪ Media Type: application/atom+xml;type=feed
- 7783 ▪ Resources: AtomPub Workspace Element in Service Document
- 7784 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>
- 7785 ▪ Used in AtomPub Service Document to identify the folder tree for a specified
- 7786 folder
- 7787 ▪ Service: getFolderTree
- 7788 ▪ Media Type: application/atom+xml;type=feed
- 7789 ▪ Resources: CMIS Folder, also used in AtomPub Service Document for root folder
- 7790 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>
- 7791 ▪ Used in AtomPub Service Document to identify the base types descendants
- 7792 ▪ Service: getTypeDescendants
- 7793 ▪ Media Type: application/atom+xml;type=feed
- 7794 ▪ Resources: AtomPub Workspace Element in Service Document
- 7795 ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>
- 7796 ▪ Used in AtomPub Service Document to identify the root folder descendants
- 7797 ▪ Service: getDescendants for root folder
- 7798 ▪ Media Type: application/atom+xml;type=feed
- 7799 ▪ Resources: AtomPub Workspace Element in Service Document
- 7800

7801 3.5 Atom Resources

7802 For all Atom Resources used in this specification, the following MUST be followed:

7803 3.5.1 Feeds

7804 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmisis objects:

- 7805 • atom:updated SHOULD be the latest time the folder or its contents was updated. If unknown by
- 7806 the underlying repository, it MUST be the current time.
- 7807 • atom:author/atom:name MUST be the CMIS property cmisis:createdBy
- 7808 • atom:title MUST be the CMIS property cmisis:name
- 7809 • The atom:link with relation self MUST be generated to return the URI of the feed. If paging or any
- 7810 other mechanism is used to filter, sort, or change the representation of the feed, the URI MUST
- 7811 point back a resource with the same representation.
- 7812 • A feed SHOULD contain the element app:collection, describing the appropriate media types
- 7813 supported for creation of new entries in the feed
- 7814 • atom:id SHOULD be derived from cmisis:objectId. This id MUST be compliant with atom:ids
- 7815 specification and be a valid URI.
- 7816 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
- 7817 contained in the feed, then a link with the relation next MUST be included in the feed.
- 7818

- 7819 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis types:
- 7820 • atom:updated SHOULD be the latest time type definition was updated. If unknown by the
 - 7821 underlying repository, it MUST be the current time.
 - 7822 • atom:author/atom:name is repository specific
 - 7823 • atom:title MUST be the displayName attribute of the CMIS Type Definition.
 - 7824 • The atom:link with relation self MUST be generated to return the URI of the feed
 - 7825 • atom:id SHOULD be derived from the id attribute of the CMIS Type Definition. This id MUST be
 - 7826 compliant with atom:s specification and be a valid URI.
 - 7827 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
 - 7828 contained in the feed, then a link with the relation next MUST be included in the feed.

7829

7830 If on the root type, all fields are repository specific.

7831

7832 Ordering of entries in a feed is repository-specific if orderBy argument is not specified. If orderBy

7833 argument is specified, the order of the entries in the feed SHOULD conform to the ordering specified by

7834 the orderBy argument.

7835

7836 Note: Please see feedvalidator.org to validate Atom compliance.

7837 3.5.2 Entries

7838 At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry

7839 document and conform to the guidelines below for a cmis object:

- 7840 • atom:title MUST be the cmis:name property
- 7841 • app:edited MUST be cmis:~~lastModificationDate~~lastModifiedDate
- 7842 • atom:updated MUST be cmis:~~lastModificationDate~~lastModifiedDate
- 7843 • atom:published MUST be cmis:~~creationDate~~createdDate
- 7844 • atom:author/atom:name MUST be cmis:createdBy
- 7845 • All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are
- 7846 duplicated in an atom element
- 7847 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom:s
- 7848 specification and be a valid ~~URI~~URI.
- 7849 • The repository SHOULD populate the atom:summary tag with text that best represents a
- 7850 summary of the object. For example, an HTML table containing the properties and their values or
- 7851 the description of the document if available.

7852

7853 For Documents that support Content Streams:

7854 The repository SHOULD use the atom:content/src attribute to point to the content stream.

7855 The client SHOULD use cmisra:content if the content is not well-formed or would have

7856 trouble fitting inside an atom:content element. The repository MUST use the

7857 cmisra:content element if provided by the client over the atom:content element.

7858

7859 Other Objects (Folders, Relationships, and other Document Types that do not support Content

7860 Streams, etc):

7861 The repository MUST comply with the atom specification and have an atom:content

7862 element. This is repository specific. Any value in the content field MUST be ignored if the

7863 atom entry represents a non-document object by the CMIS repository when the atom
7864 entry is POST^{ed} to a collection or sent to the repository via a PUT.

7865
7866 When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding
7867 writable CMIS property. For example, atom:title will overwrite cmis:name.

7868
7869 At any point where an Atom document of CMIS Type is sent or returned, it must be a valid Atom Entry
7870 document and conform to the guidelines below for a cmis type definition:

- 7871 • atom:title MUST be the cmis:displayName
- 7872 • The repository SHOULD populate the atom:summary tag with text that best represents a
7873 summary of the object. For example, the type description if available.
- 7874 • The repository MUST comply with the atom specification and have an atom:content element. This
7875 is repository specific. Any value in the content field MUST be ignored if the atom entry represents
7876 a non-document object by the CMIS repository when the atom entry is POST^{ed} to a collection or
7877 sent to the repository via a PUT.

7878
7879
7880 Any atom element that is not specified is repository-specific.

7881 3.5.2.1 Hierarchical Atom Entries

7882 The repository SHOULD NOT provide any links to hierarchical objects if those capabilities are not
7883 supported with the exception of getTypeDescendants which is required

7884
7885 For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate
7886 a cmisra:children element in the atom:entry with the enclosing feed of its direct children. This pattern
7887 continues until the depth is satisfied.

7888
7889 The cmisra:children element MUST include an atom:feed element that contains the children entries of this
7890 resource.

7891
7892 If an entry does not contain cmisra:children element, then the entry MAY have children even though it is
7893 not represented in the atom entry.

7894
7895 For Example, here is a minimal Atom Entry with CMIS Children Extension Element:

```
7896 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
7897 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"  
7898 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"  
7899 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
7900 open.org/ns/cmisis/restatom/200908/">  
7901 <atom:author>  
7902 <atom:name>Al Brown</atom:name>  
7903 </atom:author>  
7904 <atom:content src="http://cmisexample.oasis-open.org/repl/af1d8c7f-b554-  
7905 4dfb-bfe1-1f41e4b34fef"/>  
7906 <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef</atom:id>  
7907 <atom:title type="text">CMIS Example Folder as Customer  
7908 type</atom:title>  
7909 <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>  
7910 <cmisra:object>  
7911 <cmis:properties>
```

```

7913 |         <cmis:propertyId localName=""rep-cmis:objectId""
7914 | propertyDefinitionId=""cmis:objectId"">
7915 |             <cmis:value>af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef</cmis:value>
7916 |         </cmis:propertyId>
7917 |     </cmis:properties>
7918 | </cmisra:object>
7919 | <cmisra:pathSegment>customer</cmisra:pathSegment>
7920 | <cmisra:children>
7921 |     <atom:feed>
7922 |         <atom:title type=""text"">CMIS Example Folder as Customer
7923 | type</atom:title>
7924 |         <atom:author>
7925 |             <atom:name>Al Brown</atom:name>
7926 |             <atom:uri>http://www.ibm.com/</atom:uri>
7927 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
7928 |         </atom:author>
7929 |         <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
7930 |         <atom:id>urn:uuid:ce2d65af-b246-454b-90ff-0986d9b05178</atom:id>
7931 |         <atom:link type=""application/atom+xml;type=feed"" rel=""self""
7932 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
7933 | 1f41e4b34fef/3""/>
7934 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
7935 | href=""http://cmisexample.oasis-open.org/rep1//service""/>
7936 |         <atom:link type=""application/atom+xml;type=entry"" rel=""via""
7937 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
7938 | 1f41e4b34fef""/>
7939 |         <atom:link type=""application/atom+xml;type=feed""
7940 | rel=""http://docs.oasis-open.org/ns/cmis/link/200908/foldertree""
7941 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
7942 | 1f41e4b34fef/foldertree""/>
7943 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
7944 | href=""http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
7945 | 1f41e4b34fef/children""/>
7946 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
7947 | href=""http://cmisexample.oasis-open.org/rep1/2eb09309-58f7-4627-b735-
7948 | 4d5cf4ba6554""/>
7949 |         <atom:entry>
7950 |             <atom:author>
7951 |                 <atom:name>Al Brown</atom:name>
7952 |             </atom:author>
7953 |             <atom:content src=""http://cmisexample.oasis-
7954 | open.org/rep1/af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef""/>
7955 |             <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-
7956 | 1f41e4b34fef</atom:id>
7957 |             <atom:title type=""text"">CMIS Example Child of
7958 | Folder</atom:title>
7959 |             <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
7960 |             <cmisra:object>
7961 |                 <cmis:properties>
7962 |                     <cmis:propertyId localName=""rep-cmis:objectId""
7963 | propertyDefinitionId=""cmis:objectId"">
7964 |                         <cmis:value>af1d8c7f-b554-4dfb-bfe1-
7965 | 1f41e4b34fef</cmis:value>
7966 |                     </cmis:propertyId>
7967 |                 </cmis:properties>
7968 |             </cmisra:object>
7969 |             <cmisra:pathSegment>document</cmisra:pathSegment>
7970 |         </atom:entry>
7971 |     </atom:feed>
7972 | </cmisra:children>
7973 | </atom:entry>
7974 |

```

7975 | Please also see the example documents included with the schema.

7976 3.6 AtomPub Service Document (Repository)

7977 The AtomPub Service Document contains the set of repositories that are available. Each repository is
7978 mapped to a app:workspace element in the AtomPub Service document.

7979

7980 CMIS Services exposed:

7981 GET: getRepositories, getRepositoryInfo

7982

7983 Media Type: application/atomsvc+xml

7984

7985 How the client will get the initial AtomPub (APP) service document or the URI for the service document is
7986 repository specific. Examples are via URI, or loading the service document from disk.

7987

7988 The service document will be available from Atom Entry and Atom Feed documents via a link relationship,
7989 service. That AtomPub service document MUST contain only one workspace element which MUST be
7990 the workspace representing the repository containing the Atom Entry or Atom Feed document.

7991

7992 A workspace element for a CMIS repository MUST have a collection element for each of following
7993 collections: Each collection MUST also contain a cmisra:collectionType element with the given value:

- 7994 • Root Folder Children Collection: Root folder of the Repository
 - 7995 ○ `"_root"` for the children collection of the root folder
 - 7996 ○ `cmisra:collectiontype="_root"`
- 7997 • Types Children Collection: Collection containing the base types in the repository
 - 7998 ○ `"_types"` for the children collection
 - 7999 ○ `cmisra:collectiontype="_types"`

8000

8001 The workspace element SHOULD contain these collections if the repository supports this functionality:

- 8002 • CheckedOut collection: collection containing all checked out documents user can see
 - 8003 ○ `"_checkedout"`
 - 8004 ○ `cmisra:collectiontype="_checkedout"`
- 8005 • Query collection: Collection for posting queries to be executed
 - 8006 ○ `"_query"`
 - 8007 ○ `cmisra:collectiontype="_query"`
- 8008 • Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
 - 8009 ○ `"_unfiled"`
 - 8010 ○ `cmisra:collectiontype="_unfiled"`

8011

8012 The repository MUST include the URI templates in the workspace elements.

8013

8014 The workspace element MUST also contain the following link element with the relation:

- 8015 • <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>: This link relation points to the
8016 types descendants for the base types in the repository.

8017

8018 The workspace element MUST contain the following link elements of the following relations for those
8019 services which are supported by the repository:

- 8020
- 8021
- <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.
- 8022
- 8023
- 8024
- 8025
- 8026

8027 The workspace element may include app:collection element for the collections that represent folders in
8028 the repository. However, an alternative approach, especially for a repository with many folders, is to not
8029 enumerate those collections here, but include the app:collection element per RFC5023 in the Atom Feed
8030 document.

8031 3.6.1 URI Templates

8032

8033 CMIS defines the following URI Templates:

- 8034
- objectbyid
- 8035
- objectbypath
- 8036
- query
- 8037
- typebyid
- 8038

8039 Repositories MUST provide the following URI Templates:

- 8040
- objectbyid
- 8041
- objectbypath
- 8042
- typebyid
- 8043

8044 Repositories MUST provide the URI Template query if the repository supports query.

8045

8046 URI Templates MUST only be used with HTTP GET.

8047

8048 Repositories MAY extend that set of templates. Those URI Template Types will be repository specific.
8049 Repositories MAY have more than one entry per URI Template type if the entries have different media
8050 types.

8051

8052 URI Templates are simple replacement of the template parameter with the specified value. If a client
8053 does not want to specify a value for some of these variables, then the client MUST substitute an empty
8054 string for the variable.

8055

8056 For example, if the URI template that supports the variable {id} is

8057 `http://example.org/repl/getbyid/{id}`

8058

8059 If the client wants to find the entry for an object with an id of "obj_1" then the URI would be:

8060 `http://example.org/repl/getbyid/obj_1`

8061

8062 Arguments that are substituted for URI template parameters MUST be percent escaped according to
8063 RFC3986. Please see that RFC for more information.

8064
8065 All variables MUST be in the template.

8066
8067 Structure of URI Template:

```
8068 <xs:complexType name="cmisUriTemplateType">
8069   <xs:sequence>
8070     <xs:element name="template" type="xs:string" />
8071     <xs:element name="type" type="xs:string" />
8072     <xs:element name="mediatype" type="xs:string" />
8073     <xs:any processContents="lax" namespace="##other"
8074 minOccurs="0"
8075           maxOccurs="unbounded" />
8076   </xs:sequence>
8077 </xs:complexType>
```

8078
8079
8080 Example of URI Template element in an AtomPub Workspace Element:

```
8081 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8082 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
8083 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8084 open.org/ns/cmisis/messaging/200908/"
8085 xmlns:atom="http://www.w3.org/2005/Atom"
8086 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8087 open.org/ns/cmisis/restatom/200908/">
8088   <cmisra:template>http://cmisexample.oasis-
8089 open.org/rep1/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
8090 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
8091 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
8092   <cmisra:type>objectbyid</cmisra:type>
8093   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
8094 </cmisra:uritemplate>
```

8095
8096 Please also see the example documents included with the schema.
8097

8098 3.6.1.1 Object By Id

8099 This URI template provides a method for creating an URI that directly accesses an atom entry
8100 representing documents, folders, policies or relationship objects. See section 3.10 for more information.

8101
8102 Type: objectbyid
8103 Media Type: application/atom+xml;type=entry

8104
8105 Service: getObjectById

8106
8107 Variables that are supported by the template:

- 8108 • {id}: Id of object
- 8109 • {filter}: Property Filter
- 8110 • {includeAllowableActions}
 - 8111 ○ Valid values: true, false
- 8112 • {includePolicyIds}: Include Policy Ids:
 - 8113 ○ Valid values: true, false

- 8114 • {includeRelationships}: Include relationships
 - 8115 ○ Valid values: See enumIncludeRelationships
- 8116 • {includeACL}: Include ACLs
 - 8117 ○ Valid values: true, false
- 8118 • {renditionFilter}
 - 8119 ○ Valid values: Please see renditionFilter in CMIS Domain Model

8120

8121 Example:

```

8122 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8123 <cmisra:writetemplate xmlns:cmis="http://docs.oasis-
8124 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8125 open.org/ns/cmisis/messaging/200908/"
8126 xmlns:atom="http://www.w3.org/2005/Atom"
8127 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8128 open.org/ns/cmisis/restatom/200908/">
8129   <cmisra:template>http://cmisexample.oasis-
8130 open.org/rep1/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
8131 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
8132 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
8133   <cmisra:type>objectbyid</cmisra:type>
8134   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
8135 </cmisra:writetemplate>
8136

```

8137

8138 Please also see the example documents included with the schema.

8139 3.6.1.2 Object By Path

8140 This URI template provides a method for creating an URI that directly accesses an atom entry
 8141 representing documents, folders or policy objects. See section 3.10 for more information.

8142

8143 Type: objectbypath

8144 Media Type: application/atom+xml;type=entry

8145

8146 Service: getObjectByPath

8147

8148 Variables that are supported by the template:

- 8149 • {path}: Path of Object
- 8150 • {filter}: Property Filter
- 8151 • {includeAllowableActions}: Boolean for include Allowable Actions
 - 8152 ○ Valid values: true, false
- 8153 • {includePolicyIds}: Include Policy Ids:
 - 8154 ○ Valid values: true, false
- 8155 • {includeRelationships}: Include relationships
 - 8156 ○ Valid values: See enumIncludeRelationships
- 8157 • {includeACL}: Include ACLs
 - 8158 ○ Valid values: true, false
- 8159 • {renditionFilter}

8160 o Valid values: Please see renditionFilter in CMIS Domain Model

8161

8162

8163 Example:

```
8164 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8165 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
8166 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8167 open.org/ns/cmisis/messaging/200908/"
8168 xmlns:atom="http://www.w3.org/2005/Atom"
8169 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8170 open.org/ns/cmisis/restatom/200908/">
8171   <cmisra:template>http://cmisexample.oasis-
8172 open.org/repl/objectbypath?p={path}&filter={filter}&includeAllowableAc
8173 tions={includeAllowableActions}&includePolicyIds={includePolicyIds}&in
8174 cludeRelationships={includeRelationships}&includeACL={includeACL}</cmisra:
8175 template>
8176   <cmisra:type>objectbypath</cmisra:type>
8177   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
8178 </cmisra:uritemplate>
8179
```

8180

8181 Please also see the example documents included with the schema.

8182

8183 3.6.1.3 Query

8184 Type: query

8185 Media Type: application/atom+xml;type=feed

8186

8187 Service: query

8188

8189 Variables that are supported by the template:

- 8190 • {q}: CMIS Query Statement
- 8191 • {searchAllVersions}: Boolean, true if to search all versions
- 8192 • {maxItems}: Integer, Max items to return
- 8193 • {skipCount}: Integer, Items to skip
- 8194 • {includeAllowableActions}: Boolean
- 8195 • {includeRelationships}: Boolean

8196

8197

8198 Example:

```
8199 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8200 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
8201 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8202 open.org/ns/cmisis/messaging/200908/"
8203 xmlns:atom="http://www.w3.org/2005/Atom"
8204 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8205 open.org/ns/cmisis/restatom/200908/">
```



```
8206     <cmisra:template>http://cmisexample.oasis-
8207 open.org/rep1/query?q={q}&searchAllVersions={searchAllVersions}&maxItems={maxItems}&skipCount={skipCount}&includeAllowableActions={includeAllowableActions}=&includeRelationships={includeRelationships}</cmisra:template>
8211     <cmisra:type>query</cmisra:type>
8212     <cmisra:mediatype>application/atom+xml;type=feed</cmisra:mediatype>
8213 </cmisra:uritemplate>
8214
```

8215
8216 Please also see the example documents included with the schema.
8217

8218 3.6.1.4 Type By Id

8219 Type: typebyid
8220 Media Type: application/atom+xml;type=entry
8221
8222 Service: getTypeDefinition
8223

8224 Variables that are supported by the template:

- 8225 • {id}: CMIS Type Id

8226
8227
8228 Example:

```
8229 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8230 <cmisra:uritemplate xmlns:cmisra="http://docs.oasis-
8231 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8232 open.org/ns/cmisis/messaging/200908/"
8233 xmlns:atom="http://www.w3.org/2005/Atom"
8234 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8235 open.org/ns/cmisis/restatom/200908/">
8236   <cmisra:template>http://cmisexample.oasis-
8237 open.org/rep1/type?id={id}</cmisra:template>
8238   <cmisra:type>query</cmisra:type>
8239   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
8240 </cmisra:uritemplate>
```

8241
8242 Please also see the example documents included with the schema.
8243
8244

8245 3.6.2 HTTP Methods

8246 3.6.2.1 GET

8247 This retrieves the AtomPub Service document for a specified repository. This exposes the capabilities defined in getRepositories and getRepositoryInfo in the Domain Model.

8248
8249
8250 The optional argument MAY be specified:

- 8251 • repositoryId:

- 8252 ○ This query parameter allows a client to specify a different repository than the one that is
8253 referenced by the URI.
- 8254 ○ If specified, the repository **MUST** return the AtomPub services document for the specified
8255 repository if that repository exists.
- 8256 ○ If not specified, the repository **MUST** return the service document for the repository that is
8257 referenced by URI.
- 8258

8259 **3.7 Service Collections**

8260 These are the collections that are included on an AtomPub Service document in the workspace element.
8261 For any HTTP verb not specified on a resource, each implementation **MAY** chose to implement that HTTP
8262 verb in a repository-specific manner.

8263 **3.7.1 Root Folder Collection**

8264 This is a collection described in the service document. Please see [Folder Children](#).

8265 **3.7.2 Query Collection**

8266 This is a collection for processing queries. If the implementation supports GET on this collection, then the
8267 implementation **SHOULD** at least return a feed consisting of zero or more atom entries. These atom
8268 entries should represent persisted objects related to query such as persisted queries, long running
8269 queries or search templates.

8270

8271 CMIS Services exposed via HTTP verbs:

8272 POST: Query

8273

8274 Media Type: application/atom+xml;type=feed

8275 Accept:

- 8276 • **MUST** support CMIS Query document,
- 8277 • **MAY** support other media type

8278

8279 Link Relations on resulting feed from Query Collection:

- 8280 • service: Points to service document containing the CMIS repository. The service document
8281 **MUST** contain only one workspace element.
 - 8282 ○ Media Type: application/atomsvc+xml
- 8283 • paging link relations as appropriate: first, next, previous, last

8284

8285 The following CMIS Atom extension element **MAY** be included inside the atom feed:

- 8286 • cmisra:numItems

8287

8288 The following CMIS Atom extension element **MUST** be included inside the atom entries:

- 8289 • cmisra:object inside atom:entry

8290

8291 **3.7.2.1 POST**

8292 This collection **MUST** accept CMIS Query documents (application/cmisquery+xml).

8293

8294 Upon submission (creation) of a query document, a response must be returned with a Location header
8295 representing the feed for that query. If the query cannot be performed and an atom feed returned, the
8296 repository MUST return the appropriate HTTP status code. In addition, the server SHOULD return the
8297 feed directly. If the server does so, the server should also return the Content-Location header.

8298

8299 The feed returned MUST contain a set of atom entries representing the result set from the query.

8300

8301 The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The
8302 atom entries MUST contain the CMIS extension element (cmisra:object) containing the properties
8303 specified by the query in the select clause of the query statement.

8304

8305 If all the selected properties can be mapped to the same type reference, then the repository MAY include
8306 additional information in the atom entry.

8307

8308 Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

8309

8310 Status Codes:

- 8311 • 201 Success

8312

8313 Headers returned:

- 8314 • Location Header
- 8315 • Content-Location Header

8316

8317 Link Relations on resulting feed from POST to Query Collection:

- 8318 • service: Points to service document containing the CMIS repository. The service document
8319 MUST contain only one workspace element.
 - 8320 ○ Media Type: application/atomsvc+xml
- 8321 • paging link relations as appropriate: first, next, previous, last

8322

8323 Example client request:

```
8324 POST /Query HTTP/1.1  
8325 Host: example.org  
8326 Content-Length: 756  
8327 Content-Type: application/cmismessage+xml
```

8328

8329

```
8330 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
8331 <cmis:query xmlns:app="http://www.w3.org/2007/app" xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
8332 open.org/ns/cmismessaging/200908/" xmlns:cmism="http://docs.oasis-  
8333 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-  
8334 open.org/ns/cmismessaging/200908/">  
8335 <cmis:statement>SELECT cmis:objectId FROM cmis:document</cmis:statement>  
8336 <cmis:searchAllVersions>true</cmis:searchAllVersions>  
8337 <cmis:includeAllowableActions>>false</cmis:includeAllowableActions>  
8338 <cmis:includeRelationships>none</cmis:includeRelationships>  
8339 <cmis:renditionFilter>*</cmis:renditionFilter>  
8340 <cmis:maxItems>50</cmis:maxItems>  
8341 <cmis:skipCount>0</cmis:skipCount>  
8342 </cmis:query>
```

8343

8344

8345

8346

8347 Example server response:

```

8348 HTTP/1.1 201 Created
8349 Date: Mon, 25 Jan 2010 10:21:00 -0800
8350 Content-Length: 1830
8351 Content-Type: application/atom+xml;type=feed
8352 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/44ce5b47-
8353 ebc3-4513-86e0-d3f46c77d0a8
8354 Location: http://cmisexample.oasis-open.org/rep1/queryresult/44ce5b47-ebc3-
8355 4513-86e0-d3f46c77d0a8
8356
8357
8358 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8359 <atom:feed xmlns:app="http://www.w3.org/2007/app"
8360 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8361 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8362 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8363 open.org/ns/cmisis/restatom/200908/">
8364   <atom:title type="text">CMIS Query Result for SELECT cmis:objectId FROM
8365 cmis:document</atom:title>
8366   <atom:author>
8367     <atom:name>Al Brown</atom:name>
8368     <atom:uri>http://www.ibm.com/</atom:uri>
8369     <atom:email>albertcbrown@us.ibm.com</atom:email>
8370   </atom:author>
8371   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
8372   <atom:id>urn:uuid:811b1b9b-80f5-4788-b46c-aa77564e294b</atom:id>
8373   <atom:link type="application/atom+xml;type=feed" rel="self"
8374 href="http://cmisexample.oasis-open.org/rep1/11355977-434b-4e71-b83a-
8375 77dea9878e04/3"/>
8376   <atom:link type="application/atomsvc+xml" rel="service"
8377 href="http://cmisexample.oasis-open.org/rep1//service"/>
8378   <atom:entry>
8379     <atom:author>
8380       <atom:name>Al Brown</atom:name>
8381     </atom:author>
8382     <atom:content src="http://cmisexample.oasis-open.org/rep1/a3386ea0-
8383 0477-4a74-96bd-70d3da1c483a"/>
8384     <atom:id>urn:uuid:a3386ea0-0477-4a74-96bd-70d3da1c483a</atom:id>
8385     <atom:title type="text">Resulting Document</atom:title>
8386     <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
8387     <cmisra:object>
8388       <cmis:properties>
8389         <cmis:propertyId queryName="cmis:objectId" localName="rep-
8390 cmis:objectId" propertyDefinitionId="cmis:objectId">
8391           <cmis:value>a3386ea0-0477-4a74-96bd-
8392 70d3da1c483a</cmis:value>
8393         </cmis:propertyId>
8394       </cmis:properties>
8395     </cmisra:object>
8396   </atom:entry>
8397 </atom:feed>
8398

```

8399

8400 Please also see the example documents included with the schema.

8401

8402 3.7.3 Checked Out Collection

8403 This is a collection described in the service document that contains the private working copies (PWCs) of
8404 the checkedout documents in the repository.

8405 CMIS Services:

8406 GET: getCheckedOutDocs

8407 POST: checkOut

8408 Media Type: application/atom+xml;type=feed

8409 Accept:

8410 • MUST support Atom Entry Documents with CMIS extensions

8411 ○ application/atom+xml;type=entry or

8412 ○ application/cmisaatom+xml

8413 • MAY support other media type

8414

8415 Link Relations:

8416 • service: Points to service document containing the CMIS repository. The service document
8417 MUST contain only one workspace element.

8418 ○ Media Type: application/atomsvc+xml

8419 • paging link relations as appropriate: first, next, previous, last

8420

8421 The following CMIS Atom extension element MAY be included inside the atom feed:

8422 • cmisra:numItems

8423

8424 The following CMIS Atom extension element MUST be included inside the atom entries:

8425 • cmisra:object inside atom:entry

8426

8427 3.7.3.1 GET

8428 The following arguments may be supplied. Please see the domain model for more information:

8429 • filter

8430 • folderId

8431 • maxItems

8432 • skipCount

8433 • renditionFilter

8434 • includeAllowableActions

8435 • includeRelationships

8436 3.7.3.2 POST

8437 | When an atom entry is POST^{ed} to this collection, the atom entry will be checked out. A Content-
8438 Location header MUST be returned containing the location of the private working copy.

8439

8440 Example client request:

8441 POST /CheckedOut HTTP/1.1

8442 Host: example.org

8443 Content-Length: 1044

8444 Content-Type: application/atom+xml;type=entry

8445

8446

8447 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>

```

8448 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
8449 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8450 | open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
8451 | open.org/ns/cmismessaging/200908/" xmlns:cmis="http://docs.oasis-
8452 | open.org/ns/cmismessaging/200908/">
8453 |   <atom:author>
8454 |     <atom:name>Al Brown</atom:name>
8455 |   </atom:author>
8456 |   <atom:content src="http://cmisexample.oasis-open.org/rep1/8d32d716-701b-
8457 | 4491-84e8-ad57c8230940"/>
8458 |   <atom:id>urn:uuid:8d32d716-701b-4491-84e8-ad57c8230940</atom:id>
8459 |   <atom:title type="text">CMIS Example Document to checkout</atom:title>
8460 |   <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
8461 |   <cmisra:object>
8462 |     <cmis:properties>
8463 |       <cmis:propertyId localName="rep-cmis:objectId"
8464 | propertyDefinitionId="cmis:objectId">
8465 |         <cmis:value>8d32d716-701b-4491-84e8-ad57c8230940</cmis:value>
8466 |       </cmis:propertyId>
8467 |     </cmis:properties>
8468 |   </cmisra:object>
8469 | </atom:entry>
8470 |

```

8471

8472 **Example server response:**

```

8473 | HTTP/1.1 201 Created
8474 | Date: Mon, 25 Jan 2010 10:21:00 -0800
8475 | Content-Length: 7846
8476 | Content-Type: application/atom+xml;type=entry
8477 | Content-Location: http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-
8478 | 8fab-4aa6e6797dbe
8479 | Location: http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8480 | 4aa6e6797dbe
8481 |
8482 |
8483 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8484 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
8485 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8486 | open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
8487 | open.org/ns/cmismessaging/200908/" xmlns:cmis="http://docs.oasis-
8488 | open.org/ns/cmismessaging/200908/">
8489 |   <atom:author>
8490 |     <atom:name>Al Brown</atom:name>
8491 |     <atom:uri>http://www.ibm.com/</atom:uri>
8492 |     <atom:email>albertcbrown@us.ibm.com</atom:email>
8493 |   </atom:author>
8494 |   <atom:content src="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-
8495 | 491c-8fab-4aa6e6797dbe"/>
8496 |   <atom:id>urn:uuid:6cce57fc-4e31-491c-8fab-4aa6e6797dbe</atom:id>
8497 |   <atom:title type="text">CMIS Example Child of Folder</atom:title>
8498 |   <atom:updated>2010-01-25T10:21:00.396-08:00</atom:updated>
8499 |   <atom:link rel="self" href="http://cmisexample.oasis-
8500 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
8501 |   <atom:link rel="edit" href="http://cmisexample.oasis-
8502 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
8503 |   <atom:link type="application/cmismessaging+xml;type=allowableActions"
8504 | rel="http://docs.oasis-open.org/ns/cmismessaging/200908/allowableactions"
8505 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8506 | 4aa6e6797dbe/allowableactions"/>
8507 |   <atom:link type="application/atom+xml;type=entry" rel="describedby"
8508 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8509 | 4aa6e6797dbe/type"/>

```

```

8510 | <atom:link type="application/atomsvc+xml" rel="service"
8511 | href="http://cmisexample.oasis-open.org/rep1//service"/>
8512 | <atom:published>2010-01-25T10:21:00.396-08:00</atom:published>
8513 | <atom:summary type="html">HTML summary of Entry 6cce57fc-4e31-491c-
8514 | 8fab-4aa6e6797dbe</atom:summary>
8515 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
8516 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/edit-media"/>
8517 | <atom:link rel="alternate" href="http://cmisexample.oasis-
8518 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/alternate"/>
8519 | <atom:link type="application/atom+xml;type=feed" rel="up"
8520 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8521 | 4aa6e6797dbe/parents"/>
8522 | <atom:link type="application/atom+xml;type=feed" rel="version-history"
8523 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8524 | 4aa6e6797dbe/allversions"/>
8525 | <atom:link type="application/atom+xml;type=entry" rel="current-
8526 | version" href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-
8527 | 8fab-4aa6e6797dbe/latest"/>
8528 | <atom:link type="application/atom+xml;type=feed"
8529 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
8530 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8531 | 4aa6e6797dbe/relationships"/>
8532 | <atom:link type="application/atom+xml;type=feed"
8533 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
8534 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8535 | 4aa6e6797dbe/policies"/>
8536 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8537 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
8538 | open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/acl"/>
8539 | <atom:link type="application/atom+xml;type=feed" rel="working-copy"
8540 | href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
8541 | 4aa6e6797dbe/pwc"/>
8542 | <cmisra:object>
8543 | <cmis:properties>
8544 | <cmis:propertyId localName="rep-cmis:objectId"
8545 | propertyDefinitionId="cmis:objectId">
8546 | <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
8547 | </cmis:propertyId>
8548 | <cmis:propertyId localName="rep-cmis:objectTypeId"
8549 | propertyDefinitionId="cmis:objectTypeId">
8550 | <cmis:value>customer</cmis:value>
8551 | </cmis:propertyId>
8552 | <cmis:propertyString localName="rep-cmis:name"
8553 | propertyDefinitionId="cmis:name">
8554 | <cmis:value>CMIS Example Child of Folder</cmis:value>
8555 | </cmis:propertyString>
8556 | <cmis:propertyDateTime localName="rep-cmis:creationDate"
8557 | propertyDefinitionId="cmis:creationDate">
8558 | <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
8559 | </cmis:propertyDateTime>
8560 | <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
8561 | propertyDefinitionId="cmis:lastModificationDate">
8562 | <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
8563 | </cmis:propertyDateTime>
8564 | <cmis:propertyId localName="rep-cmis:baseTypeId"
8565 | propertyDefinitionId="cmis:baseTypeId">
8566 | <cmis:value>cmis:document</cmis:value>
8567 | </cmis:propertyId>
8568 | <cmis:propertyString localName="rep-cmis:lastModifiedBy"
8569 | propertyDefinitionId="cmis:lastModifiedBy">
8570 | <cmis:value>Al Brown</cmis:value>
8571 | </cmis:propertyString>
8572 | <cmis:propertyString localName="rep-cmis:createdBy"
8573 | propertyDefinitionId="cmis:createdBy">

```

```

8574         <cmis:value>Al Brown</cmis:value>
8575     </cmis:propertyString>
8576     <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
8577 | propertyDefinitionId=""cmis:isLatestVersion"">
8578         <cmis:value>>true</cmis:value>
8579     </cmis:propertyBoolean>
8580     <cmis:propertyBoolean localName=""rep-
8581 | cmis:isVersionSeriesCheckedOut""
8582 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
8583         <cmis:value>>true</cmis:value>
8584     </cmis:propertyBoolean>
8585     <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
8586 | propertyDefinitionId=""cmis:isMajorVersion"">
8587         <cmis:value>>false</cmis:value>
8588     </cmis:propertyBoolean>
8589     <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
8590 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
8591         <cmis:value>>false</cmis:value>
8592     </cmis:propertyBoolean>
8593     <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
8594 | propertyDefinitionId=""cmis:isImmutable"">
8595         <cmis:value>>false</cmis:value>
8596     </cmis:propertyBoolean>
8597     <cmis:propertyString localName=""rep-cmis:checkinComment""
8598 | propertyDefinitionId=""cmis:checkinComment"">
8599         <cmis:value>Checkin comment</cmis:value>
8600     </cmis:propertyString>
8601     <cmis:propertyString localName=""rep-cmis:versionLabel""
8602 | propertyDefinitionId=""cmis:versionLabel"">
8603         <cmis:value>0.1</cmis:value>
8604     </cmis:propertyString>
8605     <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
8606 | propertyDefinitionId=""cmis:contentStreamMimeType"">
8607         <cmis:value>text/plain</cmis:value>
8608     </cmis:propertyString>
8609     <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
8610 | propertyDefinitionId=""cmis:contentStreamFileName"">
8611         <cmis:value>text.txt</cmis:value>
8612     </cmis:propertyString>
8613     <cmis:propertyInteger localName=""rep-cmis:contentStreamLength""
8614 | propertyDefinitionId=""cmis:contentStreamLength"">
8615         <cmis:value>4234</cmis:value>
8616     </cmis:propertyInteger>
8617     <cmis:propertyString displayName=""Keywords for Document""
8618 | localName=""keywords"" propertyDefinitionId=""keywords"">
8619         <cmis:value>document</cmis:value>
8620         <cmis:value>example</cmis:value>
8621         <cmis:value>sample</cmis:value>
8622         <cmis:value>cmis</cmis:value>
8623     </cmis:propertyString>
8624     <cmis:propertyId localName=""rep-cmis:versionSeriesCheckedOutId""
8625 | propertyDefinitionId=""cmis:versionSeriesCheckedOutId"">
8626         <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
8627     </cmis:propertyId>
8628     <cmis:propertyString localName=""rep-
8629 | cmis:versionSeriesCheckedOutBy""
8630 | propertyDefinitionId=""cmis:versionSeriesCheckedOutBy"">
8631         <cmis:value>Al Brown</cmis:value>
8632     </cmis:propertyString>
8633 </cmis:properties>
8634 </cmisra:object>
8635 </atom:entry>
8636

```

8637

8638 Please also see the example documents included with the schema.

8639

8640 **3.7.4 Unfiled Collection**

8641 This is a collection described in the service document that contains all the unfiled documents in the
8642 repository. This collection **MUST** be available if un-filing or multi-filing is supported by the repository.

8643 A repository that supports un-filing **MAY** provide read access (GET). If read access is not provided, the
8644 repository **SHOULD** respond to a read attempt with the HTTP status code 405 (notSupported).

8645 CMIS Services:

8646 POST: removeObjectFromFolder

8647 Media Type: application/atom+xml;type=feed

8648 Accept:

8649 • **MUST** support Atom Entry Documents with CMIS extensions

8650 ○ application/atom+xml;type=entry or

8651 ○ application/cmisatom+xml

8652 • **MAY** support other media type

8653

8654 Link Relations:

8655 • service: Points to service document containing the CMIS repository. The service document
8656 **MUST** contain only one workspace element.

8657 ○ Media Type: application/atomsvc+xml

8658 • paging link relations as appropriate: first, next, previous, last

8659

8660 The following CMIS Atom extension element **MAY** be included inside the atom feed:

8661 • cmisra:numItems

8662

8663 The following CMIS Atom extension element **MUST** be included inside the atom entries:

8664 • cmisra:object inside atom:entry

8665

8666 **3.7.4.1 POST**

8667 This removes the object from all folders in the repository by default. If the optional argument removeFrom
8668 is specified, the object will only be removed from that folder only.

8669

8670 | If the Atom Entry POST^{ed}, does not have the CMIS extensions with a valid cmis:objectId property, the
8671 document does not exist, or the document is not in that folder, the appropriate HTTP status code **MUST**
8672 be returned.

8673

8674 This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

8675 • HTTP Success: 201

8676 • Location Header

8677

8678 The following arguments may be supplied. Please see the domain model for more information:

- 8679 • removeFrom: For repositories which support multi-filing, this parameter identifies which folder to
8680 remove this object from. If specified, it indicates the folder from which the object shall be moved.
8681 If not specified, the object will be removed from all folders.

8682

8683 Example client request:

```
8684 POST /Unfiled HTTP/1.1
8685 Host: example.org
8686 Content-Length: 1043
8687 Content-Type: application/atom+xml;type=entry
8688
8689
8690 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8691 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8692 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8693 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8694 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8695 open.org/ns/cmisis/restatom/200908/">
8696   <atom:author>
8697     <atom:name>Al Brown</atom:name>
8698   </atom:author>
8699   <atom:content src="http://cmisexample.oasis-open.org/repl/12aa2bec-6f43-
8700 47d1-99ef-21797867173c"/>
8701   <atom:id>urn:uuid:12aa2bec-6f43-47d1-99ef-21797867173c</atom:id>
8702   <atom:title type="text">CMIS Example Document to unfiled</atom:title>
8703   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
8704   <cmisra:object>
8705     <cmis:properties>
8706       <cmis:propertyId localName="rep-cmis:objectId"
8707 propertyDefinitionId="cmis:objectId">
8708         <cmis:value>12aa2bec-6f43-47d1-99ef-21797867173c</cmis:value>
8709       </cmis:propertyId>
8710     </cmis:properties>
8711   </cmisra:object>
8712 </atom:entry>
8713
```

8714

8715 Example server response:

```
8716 HTTP/1.1 201 Created
8717 Date: Mon, 25 Jan 2010 10:21:00 -0800
8718 Content-Length: 7234
8719 Content-Type: application/atom+xml;type=entry
8720 Content-Location: http://cmisexample.oasis-open.org/repl/queryresult/15118373-
8721 8911-442b-9774-da3b102f224c
8722 Location: http://cmisexample.oasis-open.org/repl/queryresult/15118373-8911-
8723 442b-9774-da3b102f224c
8724
8725
8726 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8727 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8728 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8729 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8730 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8731 open.org/ns/cmisis/restatom/200908/">
8732   <atom:author>
8733     <atom:name>Al Brown</atom:name>
8734     <atom:uri>http://www.ibm.com/</atom:uri>
8735     <atom:email>albertcbrown@us.ibm.com</atom:email>
8736   </atom:author>
8737   <atom:content src="http://cmisexample.oasis-open.org/repl/15118373-8911-
8738 442b-9774-da3b102f224c"/>
8739   <atom:id>urn:uuid:15118373-8911-442b-9774-da3b102f224c</atom:id>
```

```

8740 | <atom:title type="text">CMIS Example Document to unfiled</atom:title>
8741 | <atom:updated>2010-01-25T10:21:00.443-08:00</atom:updated>
8742 | <atom:link rel="self" href="http://cmisexample.oasis-
8743 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c"/>
8744 | <atom:link rel="edit" href="http://cmisexample.oasis-
8745 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c"/>
8746 | <atom:link type="application/cmismedia+xml;type=allowableActions"
8747 | rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/allowableactions"
8748 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
8749 | da3b102f224c/allowableactions"/>
8750 | <atom:link type="application/atom+xml;type=entry" rel="describedby"
8751 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
8752 | da3b102f224c/type"/>
8753 | <atom:link type="application/atomsvc+xml" rel="service"
8754 | href="http://cmisexample.oasis-open.org/rep1//service"/>
8755 | <atom:published>2010-01-25T10:21:00.443-08:00</atom:published>
8756 | <atom:summary type="html">HTML summary of Entry 15118373-8911-442b-
8757 | 9774-da3b102f224c</atom:summary>
8758 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
8759 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c/edit-media"/>
8760 | <atom:link rel="alternate" href="http://cmisexample.oasis-
8761 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c/alternate"/>
8762 | <atom:link type="application/atom+xml;type=feed" rel="up"
8763 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
8764 | da3b102f224c/parents"/>
8765 | <atom:link type="application/atom+xml;type=feed" rel="version-history"
8766 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
8767 | da3b102f224c/allversions"/>
8768 | <atom:link type="application/atom+xml;type=entry" rel="current-
8769 | version" href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-
8770 | 9774-da3b102f224c/latest"/>
8771 | <atom:link type="application/atom+xml;type=feed"
8772 | rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/relationships"
8773 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
8774 | da3b102f224c/relationships"/>
8775 | <atom:link type="application/atom+xml;type=feed"
8776 | rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/policies"
8777 | href="http://cmisexample.oasis-open.org/rep1/15118373-8911-442b-9774-
8778 | da3b102f224c/policies"/>
8779 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8780 | open.org/ns/cmismedia/link/200908/acl" href="http://cmisexample.oasis-
8781 | open.org/rep1/15118373-8911-442b-9774-da3b102f224c/acl"/>
8782 | <cmisra:object>
8783 | <cmis:properties>
8784 | <cmis:propertyId localName="rep-cmis:objectId"
8785 | propertyDefinitionId="cmis:objectId">
8786 | <cmis:value>15118373-8911-442b-9774-da3b102f224c</cmis:value>
8787 | </cmis:propertyId>
8788 | <cmis:propertyId localName="rep-cmis:objectTypeId"
8789 | propertyDefinitionId="cmis:objectTypeId">
8790 | <cmis:value>customer</cmis:value>
8791 | </cmis:propertyId>
8792 | <cmis:propertyString localName="rep-cmis:name"
8793 | propertyDefinitionId="cmis:name">
8794 | <cmis:value>CMIS Example Document to unfiled</cmis:value>
8795 | </cmis:propertyString>
8796 | <cmis:propertyDateTime localName="rep-cmis:creationDate"
8797 | propertyDefinitionId="cmis:creationDate">
8798 | <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
8799 | </cmis:propertyDateTime>
8800 | <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
8801 | propertyDefinitionId="cmis:lastModificationDate">
8802 | <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
8803 | </cmis:propertyDateTime>

```

```

8804 |         <cmis:propertyId localName=""rep-cmis:baseTypeId""
8805 | propertyDefinitionId=""cmis:baseTypeId"">
8806 |             <cmis:value>cmis:document</cmis:value>
8807 |         </cmis:propertyId>
8808 |         <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
8809 | propertyDefinitionId=""cmis:lastModifiedBy"">
8810 |             <cmis:value>Al Brown</cmis:value>
8811 |         </cmis:propertyString>
8812 |         <cmis:propertyString localName=""rep-cmis:createdBy""
8813 | propertyDefinitionId=""cmis:createdBy"">
8814 |             <cmis:value>Al Brown</cmis:value>
8815 |         </cmis:propertyString>
8816 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
8817 | propertyDefinitionId=""cmis:isLatestVersion"">
8818 |             <cmis:value>>true</cmis:value>
8819 |         </cmis:propertyBoolean>
8820 |         <cmis:propertyBoolean localName=""rep-
8821 | cmis:isVersionSeriesCheckedOut""
8822 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
8823 |             <cmis:value>>false</cmis:value>
8824 |         </cmis:propertyBoolean>
8825 |         <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
8826 | propertyDefinitionId=""cmis:isMajorVersion"">
8827 |             <cmis:value>>false</cmis:value>
8828 |         </cmis:propertyBoolean>
8829 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
8830 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
8831 |             <cmis:value>>false</cmis:value>
8832 |         </cmis:propertyBoolean>
8833 |         <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
8834 | propertyDefinitionId=""cmis:isImmutable"">
8835 |             <cmis:value>>false</cmis:value>
8836 |         </cmis:propertyBoolean>
8837 |         <cmis:propertyString localName=""rep-cmis:checkinComment""
8838 | propertyDefinitionId=""cmis:checkinComment"">
8839 |             <cmis:value>Checkin comment</cmis:value>
8840 |         </cmis:propertyString>
8841 |         <cmis:propertyString localName=""rep-cmis:versionLabel""
8842 | propertyDefinitionId=""cmis:versionLabel"">
8843 |             <cmis:value>0.1</cmis:value>
8844 |         </cmis:propertyString>
8845 |         <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
8846 | propertyDefinitionId=""cmis:contentStreamMimeType"">
8847 |             <cmis:value>text/plain</cmis:value>
8848 |         </cmis:propertyString>
8849 |         <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
8850 | propertyDefinitionId=""cmis:contentStreamFileName"">
8851 |             <cmis:value>text.txt</cmis:value>
8852 |         </cmis:propertyString>
8853 |         <cmis:propertyInteger localName=""rep-cmis:contentStreamLength""
8854 | propertyDefinitionId=""cmis:contentStreamLength"">
8855 |             <cmis:value>4234</cmis:value>
8856 |         </cmis:propertyInteger>
8857 |         <cmis:propertyString displayName=""Keywords for Document""
8858 | localName=""keywords"" propertyDefinitionId=""keywords"">
8859 |             <cmis:value>document</cmis:value>
8860 |             <cmis:value>example</cmis:value>
8861 |             <cmis:value>sample</cmis:value>
8862 |             <cmis:value>cmis</cmis:value>
8863 |         </cmis:propertyString>
8864 |     </cmis:properties>
8865 | </cmisra:object>
8866 | </atom:entry>
8867 |

```

8868

8869 Please also see the example documents included with the schema.

8870

8871 **3.7.5 Types Children Collection**

8872 This is a collection described in the service document that contains the types in the repository under the
8873 specified parent type. If no parent type is specified, then the base types are returned in the feed. This
8874 feed does not include any nesting and is a flat feed.

8875 CMIS Services:

8876 GET: getTypeChildren

8877 Media Type: application/atom+xml;type=feed

8878

8879 Link Relations:

- 8880 • service: Points to service document containing the CMIS repository. The service document
8881 MUST contain only one workspace element.
 - 8882 ○ Media Type: application/atomsvc+xml
- 8883 • via: points to the type definition entry whose children represent this feed
- 8884 • down: points to the atom feed document representing the descendents collection for this same
8885 type with media type of application/cmistree+xml
- 8886 • paging link relations as appropriate: first, next, previous, last
- 8887 • up: points to the parent type definition
 - 8888 ○ If this is a children feed for a base object type, this link is not present.

8889

8890 This feed contains a set of atom entries for each child type definition.

8891

8892 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8893 • cmisra:numItems

8894

8895 The following CMIS Atom extension element MUST be included inside the atom entries:

- 8896 • cmisra:type inside atom:entry

8897

8898

8899 **3.7.5.1 GET**

8900 The following arguments may be supplied. Please see the domain model for more information:

- 8901 • includePropertyDefinitions
- 8902 • maxItems
- 8903 • skipCount
- 8904 • typeId

8905 **3.8 Collections**

8906 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
8907 verb in a repository-specific manner.

8908

8909 3.8.1 Relationships Collection

8910 This is the set of relationships available (either source or target or both) from a specific item such as a
8911 document, folder or policy.

8912 CMIS Services:

8913 GET: getObjectRelationships

8914 POST: createRelationship

8915 Media Type: application/atom+xml;type=feed

8916 Accept:

- 8917 • MUST support Atom Entry Documents with CMIS extensions
 - 8918 ○ application/atom+xml;type=entry or
 - 8919 ○ application/cmisaatom+xml
- 8920 • MAY support other media type

8921

8922 Link Relations:

- 8923 • service: Points to service document containing the CMIS repository. The service document
8924 MUST contain only one workspace element.
 - 8925 ○ Media Type: application/atomsvc+xml
- 8926 • paging link relations as appropriate: first, next, previous, last

8927

8928 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8929 • cmisra:numItems

8930

8931 The following CMIS Atom extension element MUST be included inside the atom entries:

- 8932 • cmisra:object inside atom:entry

8933

8934 3.8.1.1 GET

8935 The following arguments may be supplied. Please see the domain model for more information:

- 8936 • typeId
- 8937 • includeSubRelationshipTypes
- 8938 • relationshipDirection
- 8939 • maxItems
- 8940 • skipCount
- 8941 • filter
- 8942 • includeAllowableActions

8943 3.8.1.2 POST

8944 When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new
8945 CMIS relationship, then that relationship will be created.

8946 The server MUST return the appropriate HTTP status code if the source is different than the sourceId or
8947 target different than the targetId for the source and targets specified in this collection.

8948 The server MUST return the appropriate status code if the cmis:objectTypeId is not specified.

8949

8950 Example client request:

```
8951 POST /relationships/source/dbf0316c-47b5-47c9-a2fa-f005eb93f0a4 HTTP/1.1
8952 Host: example.org
8953 Content-Length: 1432
8954 Content-Type: application/atom+xml;type=entry
8955
8956
8957 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8958 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8959 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8960 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
8961 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8962 open.org/ns/cmisis/restatom/200908/">
8963   <atom:author>
8964     <atom:name>Al Brown</atom:name>
8965   </atom:author>
8966   <atom:content src="http://cmisexample.oasis-open.org/repl/dab97641-8c94-
8967 4a12-a604-7532980f05cb"/>
8968   <atom:id>urn:uuid:dab97641-8c94-4a12-a604-7532980f05cb</atom:id>
8969   <atom:title type="text">New Relationship</atom:title>
8970   <atom:updated>2010-01-25T10:20:58.864-08:00</atom:updated>
8971   <cmisra:object>
8972     <cmis:properties>
8973       <cmis:propertyId localName="rep-cmis:objectTypeId"
8974 propertyDefinitionId="cmis:objectTypeId">
8975         <cmis:value>customerRelationships</cmis:value>
8976       </cmis:propertyId>
8977       <cmis:propertyId localName="rep-cmis:sourceId"
8978 propertyDefinitionId="cmis:sourceId">
8979         <cmis:value>dbf0316c-47b5-47c9-a2fa-f005eb93f0a4</cmis:value>
8980       </cmis:propertyId>
8981       <cmis:propertyId localName="rep-cmis:targetId"
8982 propertyDefinitionId="cmis:targetId">
8983         <cmis:value>b9baac7d-7584-445e-bcd1-29af9b25bf2f</cmis:value>
8984       </cmis:propertyId>
8985     </cmis:properties>
8986   </cmisra:object>
8987 </atom:entry>
8988
```

8989

8990 Example server response:

```
8991 HTTP/1.1 201 Created
8992 Date: Mon, 25 Jan 2010 10:20:58 -0800
8993 Content-Length: 4684
8994 Content-Type: application/atom+xml;type=entry
8995 Content-Location: http://cmisexample.oasis-open.org/repl/b3006a8f-345b-4c27-
8996 86df-3f4b157bb495
8997 Location: http://cmisexample.oasis-open.org/repl/b3006a8f-345b-4c27-86df-
8998 3f4b157bb495
8999
9000
9001 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9002 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9003 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9004 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9005 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9006 open.org/ns/cmisis/restatom/200908/">
9007   <atom:author>
9008     <atom:name>Al Brown</atom:name>
9009     <atom:uri>http://www.ibm.com/</atom:uri>
9010     <atom:email>albertcbrown@us.ibm.com</atom:email>
9011   </atom:author>
```

```

9012 |     <atom:content src="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-
9013 | 4c27-86df-3f4b157bb495">/>
9014 |     <atom:id>urn:uuid:b3006a8f-345b-4c27-86df-3f4b157bb495</atom:id>
9015 |     <atom:title type="text">New Relationship</atom:title>
9016 |     <atom:updated>2010-01-25T10:20:58.880-08:00</atom:updated>
9017 |     <atom:link rel="self" href="http://cmisexample.oasis-
9018 | open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495">/>
9019 |     <atom:link rel="edit" href="http://cmisexample.oasis-
9020 | open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495">/>
9021 |     <atom:link type="application/cmism+xml;type=allowableActions"
9022 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
9023 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
9024 | 3f4b157bb495/allowableactions">/>
9025 |     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9026 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
9027 | 3f4b157bb495/type">/>
9028 |     <atom:link type="application/atomsvc+xml" rel="service"
9029 | href="http://cmisexample.oasis-open.org/rep1//service">/>
9030 |     <atom:published>2010-01-25T10:20:58.880-08:00</atom:published>
9031 |     <atom:summary type="html">HTML summary of Entry b3006a8f-345b-4c27-
9032 | 86df-3f4b157bb495</atom:summary>
9033 |     <atom:link type="application/atom+xml;type=entry"
9034 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/source"
9035 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
9036 | 3f4b157bb495/source">/>
9037 |     <atom:link type="application/atom+xml;type=entry"
9038 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/target"
9039 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
9040 | 3f4b157bb495/target">/>
9041 |     <atom:link type="application/atom+xml;type=feed"
9042 | rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
9043 | href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
9044 | 3f4b157bb495/policies">/>
9045 |     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9046 | open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
9047 | open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/acl">/>
9048 |     <cmisra:object>
9049 |       <cmis:properties>
9050 |         <cmis:propertyId localName="rep-cmis:objectId"
9051 | propertyDefinitionId="cmis:objectId">
9052 |           <cmis:value>b3006a8f-345b-4c27-86df-3f4b157bb495</cmis:value>
9053 |         </cmis:propertyId>
9054 |         <cmis:propertyId localName="rep-cmis:objectTypeId"
9055 | propertyDefinitionId="cmis:objectTypeId">
9056 |           <cmis:value>customerRelationships</cmis:value>
9057 |         </cmis:propertyId>
9058 |         <cmis:propertyString localName="rep-cmis:name"
9059 | propertyDefinitionId="cmis:name">
9060 |           <cmis:value>New Relationship</cmis:value>
9061 |         </cmis:propertyString>
9062 |         <cmis:propertyDateTime localName="rep-cmis:creationDate"
9063 | propertyDefinitionId="cmis:creationDate">
9064 |           <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
9065 |         </cmis:propertyDateTime>
9066 |         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
9067 | propertyDefinitionId="cmis:lastModificationDate">
9068 |           <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
9069 |         </cmis:propertyDateTime>
9070 |         <cmis:propertyId localName="rep-cmis:baseTypeId"
9071 | propertyDefinitionId="cmis:baseTypeId">
9072 |           <cmis:value>cmis:relationship</cmis:value>
9073 |         </cmis:propertyId>
9074 |         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
9075 | propertyDefinitionId="cmis:lastModifiedBy">

```



```

9076         <cmis:value>Al Brown</cmis:value>
9077     </cmis:propertyString>
9078     <cmis:propertyString localName=""_rep-cmis:createdBy""
9079 | propertyDefinitionId=""_cmis:createdBy"">
9080         <cmis:value>Al Brown</cmis:value>
9081     </cmis:propertyString>
9082     <cmis:propertyId localName=""_rep-cmis:sourceId""
9083 | propertyDefinitionId=""_cmis:sourceId"">
9084         <cmis:value>d4551c6d-30bd-4fc2-9c84-a55f11559e89</cmis:value>
9085     </cmis:propertyId>
9086     <cmis:propertyId localName=""_rep-cmis:targetId""
9087 | propertyDefinitionId=""_cmis:targetId"">
9088         <cmis:value>fe7e056f-c4bf-42f1-a03e-3ababcf2491d</cmis:value>
9089     </cmis:propertyId>
9090 </cmis:properties>
9091 </cmisra:object>
9092 </atom:entry>
9093

```

9094

9095 Please also see the example documents included with the schema.

9096

9097 3.8.2 Folder Children Collection

9098 This is a collection comprised of all the direct children of a particular folder represented as a feed.

9099 CMIS Services:

9100 GET: getChildren

9101 POST:

9102 createDocument

9103 or createFolder

9104 or createPolicy

9105 or moveObject

9106 or addObjectToFolder

9107

9108 Media Type: application/atom+xml;type=feed

9109

9110 Accept:

- 9111 • MUST support Atom Entry Documents with CMIS extensions

- 9112 • MAY support other media type

9113

9114 Link Relations:

- 9115 • service: Points to service document containing the CMIS repository. The service document
- 9116 MUST contain only one workspace element.

- 9117 ○ Media Type: application/atomsvc+xml

- 9118 • via: points to the atom entry of the folder generating this collection

- 9119 | • up: points to the atom entry document for this folder's parent

- 9120 ○ If the root folder, this link relation MUST NOT be included.

- 9121 ○ Media Type: application/atom+xml;type=entry

- 9122 • down: points to the atom feed document representing the descendents feed with a media type of
- 9123 application/cmistree+xml

- 9124 ○ If a repository does not support capabilityGetDescendants, then this link SHOULD NOT
9125 be included.
- 9126 • <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>: Points to the folder tree for this folder.
9127 This is represented as a feed with CMIS hierarchy extensions.
- 9128 ○ Media Type: application/atom+xml;type=feed
- 9129 • paging link relations as appropriate: first, next, previous, last

9130

9131 The following CMIS Atom extension element MAY be included inside the atom feed:

- 9132 • cmisra:numItems

9133

9134 The following CMIS Atom extension element MUST be included inside the atom entries:

- 9135 • cmisra:object inside atom:entry
- 9136 • cmisra:pathSegment inside atom:entry if pathSegment is not false

9137

9138 **3.8.2.1 GET**

9139 HTTP Code:

- 9140 • 200 OK (Success)

9141

9142 The following arguments may be supplied. Please see the domain model for more information:

- 9143 • maxItems
- 9144 • skipCount
- 9145 • filter
- 9146 • includeAllowableActions
- 9147 • includeRelationships
- 9148 • renditionFilter
- 9149 ○ If specified, renditions will be returned as links with relation alternate.
- 9150 • orderBy
- 9151 • includePathSegment

9152 **3.8.2.2 POST**

9153 CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please
9154 see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 9155 • HTTP Success: 201
- 9156 • Location Header

9157

9158 The following arguments MAY be supplied.

- 9159 • sourceFolderId: This parameter indicates the folder from which the object shall be moved from to
9160 the current specified folder. This parameter is not allowed for create operations.
- 9161 ○ If specified moveObject will be performed.
- 9162 ○ If not specified, addObjectToFolder will be performed.
- 9163 • versioningState: The optional argument versioningState MAY specify additional versioning
9164 behavior such as checkIn as major or minor. Please see CMIS Domain Model for more
9165 information on this parameter.

9166

9167 POSTing an Atom Entry document with CMIS markup:

9168 Adding a document to a folder:
9169 If the atom entry has a cmis property cmis:objectId that is valid for the repository, the object will
9170 be added to the folder.

9171
9172 When an object is added to the folder, in repositories that do not support multi-filing it will be
9173 removed from the previous folder and the operation treated as move. If the repository supports
9174 multiple folders, it will be added to the new folder.

9175 If the optional argument sourceFolderId is specified, then the object will be removed from the
9176 folder specified.

9177
9178 If atom:content is missing from the request, the repository MUST treat the missing atom:content
9179 element as an empty atom:content element.

9180 Example client request:

```
9181 POST /obj/1cd0d82f-d579-4897-9b0a-ad0917595445?sourceFolderId=313fd58d-2eab-
9182 41af-9517-06dadb010d49 HTTP/1.1
9183 Host: example.org
9184 Content-Length: 1227
9185 Content-Type: application/atom+xml;type=entry
```

```
9186
9187
9188 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9189 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9190 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9191 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9192 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9193 open.org/ns/cmis/restatom/200908/">
9194   <atom:author>
9195     <atom:name>Al Brown</atom:name>
9196   </atom:author>
9197   <atom:id>urn:uuid:1cd0d82f-d579-4897-9b0a-ad0917595445</atom:id>
9198   <atom:title type="text">Document - To Be Moved</atom:title>
9199   <atom:updated>2010-01-25T10:20:58.708-08:00</atom:updated>
9200   <atom:content src="http://cmisexample.oasis-
9201 open.org/rep1/content/1cd0d82f-d579-4897-9b0a-ad0917595445"/>
9202   <cmisra:object>
9203     <cmis:properties>
9204       <cmis:propertyId localName="rep-cmis:objectId"
9205 propertyDefinitionId="cmis:objectId">
9206         <cmis:value>1cd0d82f-d579-4897-9b0a-ad0917595445</cmis:value>
9207       </cmis:propertyId>
9208       <cmis:propertyId localName="rep-cmis:objectTypeId"
9209 propertyDefinitionId="cmis:objectTypeId">
9210         <cmis:value>invoice</cmis:value>
9211       </cmis:propertyId>
9212     </cmis:properties>
9213   </cmisra:object>
9214 </atom:entry>
```

9216
9217 Example server response:

```
9218 HTTP/1.1 201 Created
9219 Date: Mon, 25 Jan 2010 10:20:58 -0800
9220 Content-Length: 7213
9221 Content-Type: application/atom+xml;type=entry
9222 Content-Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-
9223 8141-4aed91d28b5b
9224 Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
9225 4aed91d28b5b
```

9226
9227
9228
9229
9230
9231
9232
9233
9234
9235
9236
9237
9238
9239
9240
9241
9242
9243
9244
9245
9246
9247
9248
9249
9250
9251
9252
9253
9254
9255
9256
9257
9258
9259
9260
9261
9262
9263
9264
9265
9266
9267
9268
9269
9270
9271
9272
9273
9274
9275
9276
9277
9278
9279
9280
9281
9282
9283
9284
9285
9286
9287
9288
9289

```
<?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:cmis="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/atom+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+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>
```

```

9290 |         <cmis:propertyId localName=""rep-cmis:objectId""
9291 | propertyDefinitionId=""cmis:objectId"">
9292 |             <cmis:value>invoice</cmis:value>
9293 |         </cmis:propertyId>
9294 |         <cmis:propertyString localName=""rep-cmis:name""
9295 | propertyDefinitionId=""cmis:name"">
9296 |             <cmis:value>Document - To Be Moved</cmis:value>
9297 |         </cmis:propertyString>
9298 |         <cmis:propertyDateTime localName=""rep-cmis:creationDate""
9299 | propertyDefinitionId=""cmis:creationDate"">
9300 |             <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
9301 |         </cmis:propertyDateTime>
9302 |         <cmis:propertyDateTime localName=""rep-cmis:lastModificationDate""
9303 | propertyDefinitionId=""cmis:lastModificationDate"">
9304 |             <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
9305 |         </cmis:propertyDateTime>
9306 |         <cmis:propertyId localName=""rep-cmis:baseTypeId""
9307 | propertyDefinitionId=""cmis:baseTypeId"">
9308 |             <cmis:value>cmis:document</cmis:value>
9309 |         </cmis:propertyId>
9310 |         <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
9311 | propertyDefinitionId=""cmis:lastModifiedBy"">
9312 |             <cmis:value>Al Brown</cmis:value>
9313 |         </cmis:propertyString>
9314 |         <cmis:propertyString localName=""rep-cmis:createdBy""
9315 | propertyDefinitionId=""cmis:createdBy"">
9316 |             <cmis:value>Al Brown</cmis:value>
9317 |         </cmis:propertyString>
9318 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
9319 | propertyDefinitionId=""cmis:isLatestVersion"">
9320 |             <cmis:value>>true</cmis:value>
9321 |         </cmis:propertyBoolean>
9322 |         <cmis:propertyBoolean localName=""rep-
9323 | cmis:isVersionSeriesCheckedOut""
9324 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
9325 |             <cmis:value>>false</cmis:value>
9326 |         </cmis:propertyBoolean>
9327 |         <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
9328 | propertyDefinitionId=""cmis:isMajorVersion"">
9329 |             <cmis:value>>false</cmis:value>
9330 |         </cmis:propertyBoolean>
9331 |         <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
9332 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
9333 |             <cmis:value>>false</cmis:value>
9334 |         </cmis:propertyBoolean>
9335 |         <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
9336 | propertyDefinitionId=""cmis:isImmutable"">
9337 |             <cmis:value>>false</cmis:value>
9338 |         </cmis:propertyBoolean>
9339 |         <cmis:propertyString localName=""rep-cmis:checkinComment""
9340 | propertyDefinitionId=""cmis:checkinComment"">
9341 |             <cmis:value>Checkin comment</cmis:value>
9342 |         </cmis:propertyString>
9343 |         <cmis:propertyString localName=""rep-cmis:versionLabel""
9344 | propertyDefinitionId=""cmis:versionLabel"">
9345 |             <cmis:value>0.1</cmis:value>
9346 |         </cmis:propertyString>
9347 |         <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
9348 | propertyDefinitionId=""cmis:contentStreamMimeType"">
9349 |             <cmis:value>text/plain</cmis:value>
9350 |         </cmis:propertyString>
9351 |         <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
9352 | propertyDefinitionId=""cmis:contentStreamFileName"">
9353 |             <cmis:value>text.txt</cmis:value>

```

```

9405         </cmis:propertyString>
9406         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
9407 propertyDefinitionId="cmis:contentStreamLength"
9408         <cmis:value>4234</cmis:value>
9409         </cmis:propertyInteger>
9410         <cmis:propertyString displayName="Keywords for Document"
9411 localName="keywords" propertyDefinitionId="keywords">
9412         <cmis:value>document</cmis:value>
9413         <cmis:value>example</cmis:value>
9414         <cmis:value>sample</cmis:value>
9415         <cmis:value>cmis</cmis:value>
9416         </cmis:propertyString>
9417     </cmis:properties>
9418 </cmisra:object>
9419 </atom:entry>
9420

```

9421

9422 Please also see the example documents included with the schema.

9423

9424 Creating a CMIS Object (in that folder):

9425 If the cmis:objectId property is missing, the object will be created and then added to the folder. If
9426 the cmis:objectId property is present but not a valid object Id, the repository MUST return the
9427 appropriate HTTP status code.

9428

9429 For Documents:

9430 If Content Stream is not provided and it is required by the type definition, the repository
9431 MUST return the appropriate HTTP status code.

9432

9433 Content Streams MAY be provided by any of the following mechanisms:

- 9434 ○ As part of the atom entry via the src attribute on the content element (AtomPub)
 - 9435 ■ src attribute: Implementers MAY support external references to content
 - 9436 ■ If the URI in the src attribute is not reachable, then an appropriate http
9437 status code should be returned.
- 9438 ○ As part of the atom entry inlining via the content element (AtomPub)
 - 9439 ■ Please see the AtomPub specification RFC5023 for the processing
9440 model of the content element.
- 9441 ○ If the cmisra:content is provided by the client inside the atom:entry, the
9442 cmisra:content element MUST take precedence over the atom:content element.
9443 (CMIS)
 - 9444 ● This element cmisra:content is base64 encoded
- 9445 ○ At a later time (AtomPub)
 - 9446 ■ At a later time by replacing the edit-media link with a new content

9447

9448 The optional argument versioningState MAY specify additional versioning behavior such
9449 as checkin.

9450

9451 Example client request:

```

9452 POST /obj/bb2b208b-3acd-4abe-9788-8078a239f228 HTTP/1.1
9453 Host: example.org
9454 Content-Length: 1190
9455 Content-Type: application/atom+xml;type=entry

```

Formatted
Level: 2 +
Indent at:
at 1"

Formatted
Level: 2 +
Indent at:
at 1"

```

9456
9457
9458 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9459 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9460 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9461 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9462 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9463 open.org/ns/cmisis/restatom/200908/"
9464 <atom:author>
9465 <atom:name>Al Brown</atom:name>
9466 </atom:author>
9467 <atom:id>urn:uuid:bb2b208b-3acd-4abe-9788-8078a239f228</atom:id>
9468 <atom:title type="text">New Invoice</atom:title>
9469 <atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
9470 <atom:content type="text">this is the content of the new
9471 document</atom:content>
9472 <cmisra:object>
9473 <cmis:properties>
9474 <cmis:propertyId localName="rep-cmis:objectId"
9475 propertyDefinitionId="cmis:objectId">
9476 <cmis:value>bb2b208b-3acd-4abe-9788-8078a239f228</cmis:value>
9477 </cmis:propertyId>
9478 <cmis:propertyId localName="rep-cmis:objectTypeId"
9479 propertyDefinitionId="cmis:objectTypeId">
9480 <cmis:value>invoice</cmis:value>
9481 </cmis:propertyId>
9482 </cmis:properties>
9483 </cmisra:object>
9484 </atom:entry>
9485

```

9486
9487 **Example server response:**

```

9488 HTTP/1.1 201 Created
9489 Date: Mon, 25 Jan 2010 10:20:58 -0800
9490 Content-Length: 7191
9491 Content-Type: application/atom+xml;type=entry
9492 Content-Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-
9493 ad0b-10ea94c4b93d
9494 Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9495 10ea94c4b93d
9496
9497
9498 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9499 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9500 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9501 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9502 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9503 open.org/ns/cmisis/restatom/200908/"
9504 <atom:author>
9505 <atom:name>Al Brown</atom:name>
9506 <atom:uri>http://www.ibm.com/</atom:uri>
9507 <atom:email>albertcbrown@us.ibm.com</atom:email>
9508 </atom:author>
9509 <atom:content src="http://cmisexample.oasis-open.org/rep1/13475008-6a20-
9510 4454-ad0b-10ea94c4b93d"/>
9511 <atom:id>urn:uuid:13475008-6a20-4454-ad0b-10ea94c4b93d</atom:id>
9512 <atom:title type="text">New Invoice</atom:title>
9513 <atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
9514 <atom:link rel="self" href="http://cmisexample.oasis-
9515 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>
9516 <atom:link rel="edit" href="http://cmisexample.oasis-
9517 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>

```

```

9518 |     <atom:link type="application/cmism+xml;type=allowableActions"
9519 |     rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
9520 |     href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9521 |     10ea94c4b93d/allowableactions"/>
9522 |     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9523 |     href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9524 |     10ea94c4b93d/type"/>
9525 |     <atom:link type="application/atomsvc+xml" rel="service"
9526 |     href="http://cmismexample.oasis-open.org/rep1//service"/>
9527 |     <atom:published>2010-01-25T10:20:58.833-08:00</atom:published>
9528 |     <atom:summary type="html">HTML summary of Entry 13475008-6a20-4454-
9529 |     ad0b-10ea94c4b93d</atom:summary>
9530 |     <atom:link rel="edit-media" href="http://cmismexample.oasis-
9531 |     open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/edit-media"/>
9532 |     <atom:link rel="alternate" href="http://cmismexample.oasis-
9533 |     open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/alternate"/>
9534 |     <atom:link type="application/atom+xml;type=feed" rel="up"
9535 |     href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9536 |     10ea94c4b93d/parents"/>
9537 |     <atom:link type="application/atom+xml;type=feed" rel="version-history"
9538 |     href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9539 |     10ea94c4b93d/allversions"/>
9540 |     <atom:link type="application/atom+xml;type=entry" rel="current-
9541 |     version" href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-
9542 |     ad0b-10ea94c4b93d/latest"/>
9543 |     <atom:link type="application/atom+xml;type=feed"
9544 |     rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
9545 |     href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9546 |     10ea94c4b93d/relationships"/>
9547 |     <atom:link type="application/atom+xml;type=feed"
9548 |     rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
9549 |     href="http://cmismexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
9550 |     10ea94c4b93d/policies"/>
9551 |     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9552 |     open.org/ns/cmism/link/200908/acl" href="http://cmismexample.oasis-
9553 |     open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/acl"/>
9554 |     <cmisra:object>
9555 |       <cmis:properties>
9556 |         <cmis:propertyId localName="rep-cmis:objectId"
9557 |         propertyDefinitionId="cmis:objectId">
9558 |           <cmis:value>13475008-6a20-4454-ad0b-10ea94c4b93d</cmis:value>
9559 |         </cmis:propertyId>
9560 |         <cmis:propertyId localName="rep-cmis:objectTypeId"
9561 |         propertyDefinitionId="cmis:objectTypeId">
9562 |           <cmis:value>invoice</cmis:value>
9563 |         </cmis:propertyId>
9564 |         <cmis:propertyString localName="rep-cmis:name"
9565 |         propertyDefinitionId="cmis:name">
9566 |           <cmis:value>New Invoice</cmis:value>
9567 |         </cmis:propertyString>
9568 |         <cmis:propertyDateTime localName="rep-cmis:creationDate"
9569 |         propertyDefinitionId="cmis:creationDate">
9570 |           <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
9571 |         </cmis:propertyDateTime>
9572 |         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
9573 |         propertyDefinitionId="cmis:lastModificationDate">
9574 |           <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
9575 |         </cmis:propertyDateTime>
9576 |         <cmis:propertyId localName="rep-cmis:baseTypeId"
9577 |         propertyDefinitionId="cmis:baseTypeId">
9578 |           <cmis:value>cmis:document</cmis:value>
9579 |         </cmis:propertyId>
9580 |         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
9581 |         propertyDefinitionId="cmis:lastModifiedBy">

```



```

9582         <cmis:value>Al Brown</cmis:value>
9583     </cmis:propertyString>
9584     <cmis:propertyString localName=""rep-cmis:createdBy""
9585 | propertyDefinitionId=""cmis:createdBy"">
9586         <cmis:value>Al Brown</cmis:value>
9587     </cmis:propertyString>
9588     <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
9589 | propertyDefinitionId=""cmis:isLatestVersion"">
9590         <cmis:value>>true</cmis:value>
9591     </cmis:propertyBoolean>
9592     <cmis:propertyBoolean localName=""rep-
9593 | cmis:isVersionSeriesCheckedOut""
9594 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
9595         <cmis:value>>false</cmis:value>
9596     </cmis:propertyBoolean>
9597     <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
9598 | propertyDefinitionId=""cmis:isMajorVersion"">
9599         <cmis:value>>false</cmis:value>
9600     </cmis:propertyBoolean>
9601     <cmis:propertyBoolean localName=""rep-cmis:isLatestMajorVersion""
9602 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
9603         <cmis:value>>false</cmis:value>
9604     </cmis:propertyBoolean>
9605     <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
9606 | propertyDefinitionId=""cmis:isImmutable"">
9607         <cmis:value>>false</cmis:value>
9608     </cmis:propertyBoolean>
9609     <cmis:propertyString localName=""rep-cmis:checkinComment""
9610 | propertyDefinitionId=""cmis:checkinComment"">
9611         <cmis:value>Checkin comment</cmis:value>
9612     </cmis:propertyString>
9613     <cmis:propertyString localName=""rep-cmis:versionLabel""
9614 | propertyDefinitionId=""cmis:versionLabel"">
9615         <cmis:value>0.1</cmis:value>
9616     </cmis:propertyString>
9617     <cmis:propertyString localName=""rep-cmis:contentStreamMimeType""
9618 | propertyDefinitionId=""cmis:contentStreamMimeType"">
9619         <cmis:value>text/plain</cmis:value>
9620     </cmis:propertyString>
9621     <cmis:propertyString localName=""rep-cmis:contentStreamFileName""
9622 | propertyDefinitionId=""cmis:contentStreamFileName"">
9623         <cmis:value>text.txt</cmis:value>
9624     </cmis:propertyString>
9625     <cmis:propertyInteger localName=""rep-cmis:contentStreamLength""
9626 | propertyDefinitionId=""cmis:contentStreamLength"">
9627         <cmis:value>4234</cmis:value>
9628     </cmis:propertyInteger>
9629     <cmis:propertyString displayName=""Keywords for Document""
9630 | localName=""keywords"" propertyDefinitionId=""keywords"">
9631         <cmis:value>document</cmis:value>
9632         <cmis:value>example</cmis:value>
9633         <cmis:value>sample</cmis:value>
9634         <cmis:value>cmis</cmis:value>
9635     </cmis:propertyString>
9636 </cmis:properties>
9637 </cmisra:object>
9638 </atom:entry>
9639

```

9640

9641 Please also see the example documents included with the schema.

9642

9643 POSTing other document formats: (AtomPub)

9644 The behavior is repository specific when a non Atom entry or an atom document without the
9645 CMIS elements is posted to a folder collection.
9646 For example, the repository MAY auto-create a document with a specific type (document) the
9647 client could edit.
9648 If the repository does not support this scenario or another exception occurs, then the repository
9649 MUST return the appropriate HTTP status code.

9650

9651 Optional arguments:

- 9652 • versioningState (for createDocument)
- 9653 • sourceFolderId (for moveObject)

9654

9655 3.8.3 Policies Collection

9656 This is an atom feed of all the policy objects currently applied to a specific object. This is the only
9657 collection where the URIs of the objects in the collection MUST be specific to that collection. A DELETE
9658 on the policy object in the collection is a removal of the policy from the object NOT a deletion of the policy
9659 object itself.

9660

9661 CMIS Services:

- 9662 GET: getAppliedPolicies
- 9663 POST: applyPolicy (to object representing this collection of policies)
- 9664 DELETE: removePolicy

9665 Media Type: application/atom+xml;type=feed

9666 Accept:

- 9667 • MUST support Atom Entry Documents with CMIS extensions
 - 9668 ○ application/atom+xml;type=entry or
 - 9669 ○ application/cmisaatom+xml
- 9670 • MAY support other media type

9671

9672 Link Relations:

- 9673 • service: Points to service document containing the CMIS repository. The service document
9674 MUST contain only one workspace element.
 - 9675 ○ Media Type: application/atomsvc+xml
- 9676 • via: points to the atom entry of the resource generating this collection
- 9677 • paging link relations as appropriate: first, next, previous, last

9678

9679 The policy entries displayed here are specific to the object generating this collection. A DELETE method
9680 on those URIs will invoke removePolicy().

9681

9682 The following CMIS Atom extension element MAY be included inside the atom feed:

- 9683 • cmisra:numItems

9684

9685 The following CMIS Atom extension element MUST be included inside the atom entries:

- 9686 • cmisra:object inside atom:entry

9687

9688 3.8.3.1 GET

9689 The following arguments may be supplied. Please see the domain model for more information:

- 9690 • filter

9691 3.8.3.2 POST

9692 When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the
9693 object.

9694

9695 Example client request:

```
9696 POST /policies/f3670f66-62ee-487f-b733-999a69237024 HTTP/1.1
9697 Host: example.org
9698 Content-Length: 1039
9699 Content-Type: application/atom+xml;type=entry
9700
9701
9702 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9703 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9704 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9705 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9706 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9707 open.org/ns/cmisis/restatom/200908/">
9708   <atom:author>
9709     <atom:name>Al Brown</atom:name>
9710   </atom:author>
9711   <atom:content src="http://cmisexample.oasis-open.org/rep1/f3670f66-62ee-
9712 487f-b733-999a69237024"/>
9713   <atom:id>urn:uuid:f3670f66-62ee-487f-b733-999a69237024</atom:id>
9714   <atom:title type="text">Security Policy for Invoices</atom:title>
9715   <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
9716   <cmisra:object>
9717     <cmis:properties>
9718       <cmis:propertyId localName="rep-cmis:objectId"
9719 propertyDefinitionId="cmis:objectId">
9720         <cmis:value>f3670f66-62ee-487f-b733-999a69237024</cmis:value>
9721       </cmis:propertyId>
9722     </cmis:properties>
9723   </cmisra:object>
9724 </atom:entry>
9725
```

9726

9727 Example server response:

```
9728 HTTP/1.1 201 Created
9729 Date: Mon, 25 Jan 2010 10:20:58 -0800
9730 Content-Length: 4043
9731 Content-Type: application/atom+xml;type=entry
9732 Content-Location: http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-
9733 bdf6-690761576116
9734 Location: http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-bdf6-
9735 690761576116
9736
9737
9738 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

9739 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
9740 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9741 | open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9742 | open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9743 | open.org/ns/cmisis/restatom/200908/">
9744 |   <atom:author>
9745 |     <atom:name>Al Brown</atom:name>
9746 |     <atom:uri>http://www.ibm.com/</atom:uri>
9747 |     <atom:email>albertcbrown@us.ibm.com</atom:email>
9748 |   </atom:author>
9749 |   <atom:content src="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-
9750 | 4354-bdfe-690761576116"/>
9751 |   <atom:id>urn:uuid:55cca51b-6cfa-4354-bdfe-690761576116</atom:id>
9752 |   <atom:title type="text">Security Policy for Invoices</atom:title>
9753 |   <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
9754 |   <atom:link rel="self" href="http://cmisexample.oasis-
9755 | open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116"/>
9756 |   <atom:link rel="edit" href="http://cmisexample.oasis-
9757 | open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116"/>
9758 |   <atom:link type="application/cmisis+xml;type=allowableActions"
9759 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9760 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
9761 | 690761576116/allowableactions"/>
9762 |   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9763 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
9764 | 690761576116/type"/>
9765 |   <atom:link type="application/atomsvc+xml" rel="service"
9766 | href="http://cmisexample.oasis-open.org/repl//service"/>
9767 |   <atom:published>2010-01-25T10:20:58.849-08:00</atom:published>
9768 |   <atom:summary type="html">HTML summary of Entry 55cca51b-6cfa-4354-
9769 | bdfe-690761576116</atom:summary>
9770 |   <atom:link type="application/atom+xml;type=feed" rel="up"
9771 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
9772 | 690761576116/parents"/>
9773 |   <atom:link type="application/atom+xml;type=feed"
9774 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
9775 | href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
9776 | 690761576116/relationships"/>
9777 |   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
9778 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9779 | open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116/acl"/>
9780 |   <cmisra:object>
9781 |     <cmis:properties>
9782 |       <cmis:propertyId localName="rep-cmisis:objectId"
9783 | propertyDefinitionId="cmisis:objectId">
9784 |         <cmis:value>55cca51b-6cfa-4354-bdfe-690761576116</cmis:value>
9785 |       </cmis:propertyId>
9786 |       <cmis:propertyId localName="rep-cmisis:objectTypeId"
9787 | propertyDefinitionId="cmisis:objectTypeId">
9788 |         <cmis:value>generalSecurityPolicy</cmis:value>
9789 |       </cmis:propertyId>
9790 |       <cmis:propertyString localName="rep-cmisis:name"
9791 | propertyDefinitionId="cmisis:name">
9792 |         <cmis:value>Security Policy for Invoices</cmis:value>
9793 |       </cmis:propertyString>
9794 |       <cmis:propertyDateTime localName="rep-cmisis:creationDate"
9795 | propertyDefinitionId="cmisis:creationDate">
9796 |         <cmis:value>2010-01-25T10:20:58.849-08:00</cmis:value>
9797 |       </cmis:propertyDateTime>
9798 |       <cmis:propertyDateTime localName="rep-cmisis:lastModificationDate"
9799 | propertyDefinitionId="cmisis:lastModificationDate">
9800 |         <cmis:value>2010-01-25T10:20:58.864-08:00</cmis:value>
9801 |       </cmis:propertyDateTime>

```

```

9802 |         <cmis:propertyId localName=""rep-cmis:baseTypeId"""
9803 | propertyDefinitionId=""cmis:baseTypeId""">
9804 |             <cmis:value>cmis:policy</cmis:value>
9805 |         </cmis:propertyId>
9806 |         <cmis:propertyString localName=""rep-cmis:lastModifiedBy"""
9807 | propertyDefinitionId=""cmis:lastModifiedBy""">
9808 |             <cmis:value>Al Brown</cmis:value>
9809 |         </cmis:propertyString>
9810 |         <cmis:propertyString localName=""rep-cmis:createdBy"""
9811 | propertyDefinitionId=""cmis:createdBy""">
9812 |             <cmis:value>Al Brown</cmis:value>
9813 |         </cmis:propertyString>
9814 |     </cmis:properties>
9815 | </cmisra:object>
9816 | </atom:entry>
9817 |

```

9818

9819 Please also see the example documents included with the schema.

9820 **3.8.3.3 DELETE**

9821 | This is the only collection where the URI's of the objects in the collection MUST be specific to that
9822 | collection. A DELETE on the policy object in the collection is a removal of the policy from the object NOT
9823 | a deletion of the policy object itself.

9824

9825 **3.9 Feeds**

9826 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
9827 | verb in a repository-specific manner.

9828 **3.9.1 Object Parents Feed**

9829 This is the set of parents for a specific object.

9830 CMIS Services:

9831 GET: getObjectParents

9832 Media Type: application/atom+xml;type=feed

9833

9834 Link Relations:

- 9835 • service: Points to service document containing the CMIS repository. The service document
9836 | MUST contain only one workspace element.
 - 9837 | ○ Media Type: application/atomsvc+xml
- 9838 | • via: points to the atom entry of object who's parents are represented by this collection

9839

9840 This feed contains a set of atom entries for each parent of the object that MUST contain:

- 9841 • cmisra:object inside atom:entry
- 9842 • cmisra:relativePathSegment inside atom:entry for the name of the object inside the folder

9843

9844 Example:

```

9845 | <?xml version=""1.0""" encoding=""UTF-8""" standalone=""yes"""?>

```

```

9846 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisispec/200908/"
9847 xmlns:cmism="http://docs.oasis-open.org/ns/cmisispec/messaging/200908/"
9848 xmlns:atom="http://www.w3.org/2005/Atom"
9849 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
9850 open.org/ns/cmisispec/restatom/200908/"
9851 <atom:title type="text">Parent Feed for 268d30b5-91a0-47f0-b985-
9852 6765e178f0bb</atom:title>
9853 <atom:author>
9854 <atom:name>Al Brown</atom:name>
9855 <atom:uri>http://www.ibm.com/</atom:uri>
9856 <atom:email>albertcbrown@us.ibm.com</atom:email>
9857 </atom:author>
9858 <atom:updated>2010-01-25T10:20:59.818-08:00</atom:updated>
9859 <atom:id>urn:uuid:6f541940-4abf-471b-99f0-8e6f66d53789</atom:id>
9860 <atom:link type="application/atom+xml;type=feed" rel="self"
9861 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
9862 6765e178f0bb/3"/>
9863 <atom:link type="application/atomsvc+xml" rel="service"
9864 href="http://cmisexample.oasis-open.org/rep1//service"/>
9865 <atom:link type="application/atom+xml;type=entry" rel="via"
9866 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
9867 6765e178f0bb"/>
9868 <atom:link type="application/atom+xml;type=feed" rel="first"
9869 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
9870 6765e178f0bb/first"/>
9871 <atom:link type="application/atom+xml;type=feed" rel="next"
9872 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
9873 6765e178f0bb/4"/>
9874 <atom:link type="application/atom+xml;type=feed" rel="previous"
9875 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
9876 6765e178f0bb/2"/>
9877 <atom:link type="application/atom+xml;type=feed" rel="last"
9878 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
9879 6765e178f0bb/last"/>
9880 <cmisra:numItems>1</cmisra:numItems>
9881 <atom:entry>
9882 <atom:author>
9883 <atom:name>Al Brown</atom:name>
9884 <atom:uri>http://www.ibm.com/</atom:uri>
9885 <atom:email>albertcbrown@us.ibm.com</atom:email>
9886 </atom:author>
9887 <atom:content src="http://cmisexample.oasis-open.org/rep1/661d6945-
9888 8f75-4dea-8799-7ba07b0e510e"/>
9889 <atom:id>urn:uuid:661d6945-8f75-4dea-8799-7ba07b0e510e</atom:id>
9890 <atom:title type="text">Customer Folder</atom:title>
9891 <atom:updated>2010-01-25T10:20:59.833-08:00</atom:updated>
9892 <atom:link rel="self" href="http://cmisexample.oasis-
9893 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
9894 <atom:link rel="edit" href="http://cmisexample.oasis-
9895 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
9896 <atom:link type="application/cmisispec+xml;type=allowableActions"
9897 rel="http://docs.oasis-open.org/ns/cmisispec/link/200908/allowableactions"
9898 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9899 7ba07b0e510e/allowableactions"/>
9900 <atom:link type="application/atom+xml;type=entry"
9901 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/661d6945-
9902 8f75-4dea-8799-7ba07b0e510e/type"/>
9903 <atom:link type="application/atomsvc+xml" rel="service"
9904 href="http://cmisexample.oasis-open.org/rep1//service"/>
9905 <atom:published>2010-01-25T10:20:59.833-08:00</atom:published>
9906 <atom:summary type="html">HTML summary of Entry 661d6945-8f75-4dea-
9907 8799-7ba07b0e510e</atom:summary>

```

```

9908     <atom:link type="application/atom+xml;type=entry" rel="up"
9909 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9910 7ba07b0e510e/up" />
9911     <atom:link type="application/atom+xml;type=feed" rel="down"
9912 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9913 7ba07b0e510e/children" />
9914     <atom:link type="application/cmistree+xml" rel="down"
9915 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9916 7ba07b0e510e/tree" />
9917     <atom:link type="application/atom+xml;type=feed"
9918 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
9919 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9920 7ba07b0e510e/foldertree" />
9921     <atom:link type="application/atom+xml;type=feed"
9922 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
9923 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9924 7ba07b0e510e/relationships" />
9925     <atom:link type="application/atom+xml;type=feed"
9926 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
9927 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
9928 7ba07b0e510e/policies" />
9929     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9930 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9931 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e/acl" />
9932     <cmisra:object>
9933         <cmis:properties>
9934             <cmis:propertyId localName="rep-cmis:objectId"
9935 propertyDefinitionId="cmis:objectId">
9936                 <cmis:value>661d6945-8f75-4dea-8799-
9937 7ba07b0e510e</cmis:value>
9938             </cmis:propertyId>
9939             <cmis:propertyId localName="rep-cmis:objectTypeId"
9940 propertyDefinitionId="cmis:objectTypeId">
9941                 <cmis:value>customer</cmis:value>
9942             </cmis:propertyId>
9943             <cmis:propertyString localName="rep-cmis:name"
9944 propertyDefinitionId="cmis:name">
9945                 <cmis:value>Customer Folder</cmis:value>
9946             </cmis:propertyString>
9947             <cmis:propertyDateTime localName="rep-cmis:creationDate"
9948 propertyDefinitionId="cmis:creationDate">
9949                 <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
9950             </cmis:propertyDateTime>
9951             <cmis:propertyDateTime localName="rep-
9952 cmis:lastModificationDate"
9953 propertyDefinitionId="cmis:lastModificationDate">
9954                 <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
9955             </cmis:propertyDateTime>
9956             <cmis:propertyId localName="rep-cmis:baseTypeId"
9957 propertyDefinitionId="cmis:baseTypeId">
9958                 <cmis:value>cmis:folder</cmis:value>
9959             </cmis:propertyId>
9960             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
9961 propertyDefinitionId="cmis:lastModifiedBy">
9962                 <cmis:value>Al Brown</cmis:value>
9963             </cmis:propertyString>
9964             <cmis:propertyString localName="rep-cmis:createdBy"
9965 propertyDefinitionId="cmis:createdBy">
9966                 <cmis:value>Al Brown</cmis:value>
9967             </cmis:propertyString>
9968             <cmis:propertyId localName="rep-cmis:parentId"
9969 propertyDefinitionId="cmis:parentId">
9970                 <cmis:value>661d6945-8f75-4dea-8799-
9971 7ba07b0e510eup</cmis:value>

```

```
9972         </cmis:propertyId>
9973         </cmis:properties>
9974     </cmisra:object>
9975     <cmisra:relativePathSegment>customer1</cmisra:relativePathSegment>
9976 </atom:entry>
9977 </atom:feed>
```

9978
9979 Please also see the example documents included with the schema.

9980 **3.9.1.1 GET**

9981 The following arguments may be supplied. Please see the domain model for more information:

- 9982 • filter
- 9983 • includeAllowableActions
- 9984 • includeRelationships
- 9985 • renditionFilter
- 9986 • includeRelativePathSegment
- 9987 ○ If true, then the cmisra:relativePathSegment element MUST be included in the response.

9988 **3.9.2 Changes**

9989 This is a link relationship described in the service document that contains the changes in the repository in
9990 the workspace element. The link relation pointing to this feed is [http://docs.oasis-
open.org/ns/cmis/link/200908/changes](http://docs.oasis-
9991 open.org/ns/cmis/link/200908/changes).

9992
9993 *The ChangeLog Token is specified in the URI specified by the paging link notations. Through this binding
9994 it is not possible to retrieve the ChangeLog Token from the URIs.*

9995
9996 CMIS Services:

9997 GET: getContentChanges()

9998 Media Type: application/atom+xml;type=feed

9999 Link Relations:

- 10000 • service: Points to service document containing the CMIS repository. The service document
10001 MUST contain only one workspace element.
 - 10002 ○ Media Type: application/atomsvc+xml
- 10003 • paging link relations as appropriate: first, next, previous, last
 - 10004 ○ ChangeLogToken is incorporated into the URI specified by the next link relation

10005
10006 This feed MUST be ordered from oldest first to newest.

10007
10008 If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is
10009 not available, the client should revisit the feed in the future and look for new items and the next link
10010 relation.

10011
10012 The following CMIS Atom extension element MAY be included inside the atom feed:

- 10013 • cmisra:numItems

10014
10015 The following CMIS Atom extension element MUST be included inside the atom entries:

- 10016 • cmisra:object inside atom:entry

10017

10018 Example:

```

10019 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
10020 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
10021 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
10022 xmlns:atom="http://www.w3.org/2005/Atom"
10023 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
10024 open.org/ns/cmisis/restatom/200908/">
10025   <atom:title type="text">changelog feed</atom:title>
10026   <atom:author>
10027     <atom:name>Al Brown</atom:name>
10028     <atom:uri>http://www.ibm.com/</atom:uri>
10029     <atom:email>albertcbrown@us.ibm.com</atom:email>
10030   </atom:author>
10031   <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
10032   <atom:id>urn:uuid:0bfc5306-fc76-4cd8-a0c0-7653dd43c0ff</atom:id>
10033   <atom:link type="application/atom+xml;type=feed" rel="self"
10034 href="http://cmisexample.oasis-open.org/rep1/oId/3"/>
10035   <atom:link type="application/atomsvc+xml" rel="service"
10036 href="http://cmisexample.oasis-open.org/rep1//service"/>
10037   <atom:link type="application/atom+xml;type=feed" rel="first"
10038 href="http://cmisexample.oasis-open.org/rep1/oId/first"/>
10039   <atom:link type="application/atom+xml;type=feed" rel="next"
10040 href="http://cmisexample.oasis-open.org/rep1/oId/4"/>
10041   <atom:link type="application/atom+xml;type=feed" rel="previous"
10042 href="http://cmisexample.oasis-open.org/rep1/oId/2"/>
10043   <atom:link type="application/atom+xml;type=feed" rel="last"
10044 href="http://cmisexample.oasis-open.org/rep1/oId/last"/>
10045   <cmisra:numItems>2</cmisra:numItems>
10046   <atom:entry>
10047     <atom:author>
10048       <atom:name>Al Brown</atom:name>
10049       <atom:uri>http://www.ibm.com/</atom:uri>
10050       <atom:email>albertcbrown@us.ibm.com</atom:email>
10051     </atom:author>
10052     <atom:content src="http://cmisexample.oasis-open.org/rep1/3f724c1d-
10053 12c8-43f2-919f-674df52b6ebd"/>
10054     <atom:id>urn:uuid:3f724c1d-12c8-43f2-919f-674df52b6ebd</atom:id>
10055     <atom:title type="text">CMIS Example Folder as Customer Policy
10056 type</atom:title>
10057     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
10058     <atom:link rel="self" href="http://cmisexample.oasis-
10059 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
10060     <atom:link rel="edit" href="http://cmisexample.oasis-
10061 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
10062     <atom:link type="application/cmisis+xml;type=allowableActions"
10063 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
10064 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10065 674df52b6ebd/allowableactions"/>
10066     <atom:link type="application/atom+xml;type=entry"
10067 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/3f724c1d-
10068 12c8-43f2-919f-674df52b6ebd/type"/>
10069     <atom:link type="application/atomsvc+xml" rel="service"
10070 href="http://cmisexample.oasis-open.org/rep1//service"/>
10071     <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
10072     <atom:summary type="html">HTML summary of Entry 3f724c1d-12c8-43f2-
10073 919f-674df52b6ebd</atom:summary>
10074     <atom:link type="application/atom+xml;type=entry" rel="up"
10075 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10076 674df52b6ebd/up"/>

```

```

10077 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
10078 | href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10079 | 674df52b6ebd/children""/>
10080 |         <atom:link type=""application/cmistree+xml"" rel=""down""
10081 | href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10082 | 674df52b6ebd/tree""/>
10083 |         <atom:link type=""application/atom+xml;type=feed""
10084 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree""
10085 | href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10086 | 674df52b6ebd/foldertree""/>
10087 |         <atom:link type=""application/atom+xml;type=feed""
10088 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships""
10089 | href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10090 | 674df52b6ebd/relationships""/>
10091 |         <atom:link type=""application/atom+xml;type=feed""
10092 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies""
10093 | href=""http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
10094 | 674df52b6ebd/policies""/>
10095 |         <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
10096 | open.org/ns/cmisis/link/200908/acl"" href=""http://cmisexample.oasis-
10097 | open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd/acl""/>
10098 |         <cmisra:object>
10099 |             <cmis:properties>
10100 |                 <cmis:propertyId localName=""rep-cmis:objectId""
10101 | propertyDefinitionId=""cmis:objectId"">
10102 |                     <cmis:value>3f724c1d-12c8-43f2-919f-
10103 | 674df52b6ebd</cmis:value>
10104 |                 </cmis:propertyId>
10105 |                 <cmis:propertyId localName=""rep-cmis:objectTypeId""
10106 | propertyDefinitionId=""cmis:objectTypeId"">
10107 |                     <cmis:value>customerpolicy</cmis:value>
10108 |                 </cmis:propertyId>
10109 |                 <cmis:propertyString localName=""rep-cmis:name""
10110 | propertyDefinitionId=""cmis:name"">
10111 |                     <cmis:value>CMIS Example Folder as Customer Policy
10112 | type</cmis:value>
10113 |                 </cmis:propertyString>
10114 |                 <cmis:propertyDateTime localName=""rep-cmis:creationDate""
10115 | propertyDefinitionId=""cmis:creationDate"">
10116 |                     <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
10117 |                 </cmis:propertyDateTime>
10118 |                 <cmis:propertyDateTime localName=""rep-
10119 | cmis:lastModificationDate""
10120 | propertyDefinitionId=""cmis:lastModificationDate"">
10121 |                     <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
10122 |                 </cmis:propertyDateTime>
10123 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
10124 | propertyDefinitionId=""cmis:baseTypeId"">
10125 |                     <cmis:value>cmis:folder</cmis:value>
10126 |                 </cmis:propertyId>
10127 |                 <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
10128 | propertyDefinitionId=""cmis:lastModifiedBy"">
10129 |                     <cmis:value>Al Brown</cmis:value>
10130 |                 </cmis:propertyString>
10131 |                 <cmis:propertyString localName=""rep-cmis:createdBy""
10132 | propertyDefinitionId=""cmis:createdBy"">
10133 |                     <cmis:value>Al Brown</cmis:value>
10134 |                 </cmis:propertyString>
10135 |                 <cmis:propertyId localName=""rep-cmis:parentId""
10136 | propertyDefinitionId=""cmis:parentId"">
10137 |                     <cmis:value>3f724c1d-12c8-43f2-919f-
10138 | 674df52b6ebdup</cmis:value>
10139 |                 </cmis:propertyId>
10140 |             </cmis:properties>

```

```

10141         <cmis:changeEventInfo>
10142             <cmis:changeType>updated</cmis:changeType>
10143             <cmis:changeTime>2010-01-25T10:20:59.255-
10144 08:00</cmis:changeTime>
10145         </cmis:changeEventInfo>
10146     </cmisra:object>
10147     <cmisra:pathSegment>policy</cmisra:pathSegment>
10148 </atom:entry>
10149 <atom:entry>
10150     <atom:author>
10151         <atom:name>Al Brown</atom:name>
10152         <atom:uri>http://www.ibm.com/</atom:uri>
10153         <atom:email>albertcbrown@us.ibm.com</atom:email>
10154     </atom:author>
10155     <atom:content src=""http://cmisexample.oasis-open.org/rep1/6e27bada-
10156 b5a2-4a39-be2c-269806eb0d42"">
10157     <atom:id>urn:uuid:6e27bada-b5a2-4a39-be2c-269806eb0d42</atom:id>
10158     <atom:title type=""text"">CMIS Example Document</atom:title>
10159     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
10160     <atom:link rel=""self"" href=""http://cmisexample.oasis-
10161 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"">
10162     <atom:link rel=""edit"" href=""http://cmisexample.oasis-
10163 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"">
10164     <atom:link type=""application/cmisacl+xml"" type=allowableActions""
10165 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions""
10166 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
10167 269806eb0d42/allowableactions"">
10168     <atom:link type=""application/atom+xml;type=entry""
10169 rel=""describedby"" href=""http://cmisexample.oasis-open.org/rep1/6e27bada-
10170 b5a2-4a39-be2c-269806eb0d42/type"">
10171     <atom:link type=""application/atomsvc+xml"" rel=""service""
10172 href=""http://cmisexample.oasis-open.org/rep1//service"">
10173     <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
10174     <atom:summary type=""html"">HTML summary of Entry 6e27bada-b5a2-4a39-
10175 be2c-269806eb0d42</atom:summary>
10176     <atom:link rel=""edit-media"" href=""http://cmisexample.oasis-
10177 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/edit-media"">
10178     <atom:link rel=""alternate"" href=""http://cmisexample.oasis-
10179 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/alternate"">
10180     <atom:link type=""application/atom+xml;type=feed"" rel=""up""
10181 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
10182 269806eb0d42/parents"">
10183     <atom:link type=""application/atom+xml;type=feed"" rel=""version-
10184 history"" href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-
10185 be2c-269806eb0d42/allversions"">
10186     <atom:link type=""application/atom+xml;type=entry"" rel=""current-
10187 version"" href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-
10188 be2c-269806eb0d42/latest"">
10189     <atom:link type=""application/atom+xml;type=feed""
10190 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/relationships""
10191 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
10192 269806eb0d42/relationships"">
10193     <atom:link type=""application/atom+xml;type=feed""
10194 rel=""http://docs.oasis-open.org/ns/cmisacl/link/200908/policies""
10195 href=""http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
10196 269806eb0d42/policies"">
10197     <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
10198 open.org/ns/cmisacl/link/200908/acl"" href=""http://cmisexample.oasis-
10199 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/acl"">
10200     <cmisra:object>
10201         <cmis:properties>
10202             <cmis:propertyId localName=""rep-cmis:objectId""
10203 propertyDefinitionId=""cmis:objectId"">

```

```

10204         <cmis:value>6e27bada-b5a2-4a39-be2c-
10205 269806eb0d42</cmis:value>
10206     </cmis:propertyId>
10207     <cmis:propertyId localName=""rep-cmis:objectId""
10208 propertyDefinitionId=""cmis:objectId"">
10209     <cmis:value>document</cmis:value>
10210     </cmis:propertyId>
10211     <cmis:propertyString localName=""rep-cmis:name""
10212 propertyDefinitionId=""cmis:name"">
10213     <cmis:value>CMIS Example Document</cmis:value>
10214     </cmis:propertyString>
10215     <cmis:propertyDateTime localName=""rep-cmis:creationDate""
10216 propertyDefinitionId=""cmis:creationDate"">
10217     <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
10218     </cmis:propertyDateTime>
10219     <cmis:propertyDateTime localName=""rep-
10220 cmis:lastModificationDate""
10221 propertyDefinitionId=""cmis:lastModificationDate"">
10222     <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
10223     </cmis:propertyDateTime>
10224     <cmis:propertyId localName=""rep-cmis:baseTypeId""
10225 propertyDefinitionId=""cmis:baseTypeId"">
10226     <cmis:value>cmis:document</cmis:value>
10227     </cmis:propertyId>
10228     <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
10229 propertyDefinitionId=""cmis:lastModifiedBy"">
10230     <cmis:value>Al Brown</cmis:value>
10231     </cmis:propertyString>
10232     <cmis:propertyString localName=""rep-cmis:createdBy""
10233 propertyDefinitionId=""cmis:createdBy"">
10234     <cmis:value>Al Brown</cmis:value>
10235     </cmis:propertyString>
10236     <cmis:propertyBoolean localName=""rep-cmis:isLatestVersion""
10237 propertyDefinitionId=""cmis:isLatestVersion"">
10238     <cmis:value>>true</cmis:value>
10239     </cmis:propertyBoolean>
10240     <cmis:propertyBoolean localName=""rep-
10241 cmis:isVersionSeriesCheckedOut""
10242 propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
10243     <cmis:value>>false</cmis:value>
10244     </cmis:propertyBoolean>
10245     <cmis:propertyBoolean localName=""rep-cmis:isMajorVersion""
10246 propertyDefinitionId=""cmis:isMajorVersion"">
10247     <cmis:value>>false</cmis:value>
10248     </cmis:propertyBoolean>
10249     <cmis:propertyBoolean localName=""rep-
10250 cmis:isLatestMajorVersion""
10251 propertyDefinitionId=""cmis:isLatestMajorVersion"">
10252     <cmis:value>>false</cmis:value>
10253     </cmis:propertyBoolean>
10254     <cmis:propertyBoolean localName=""rep-cmis:isImmutable""
10255 propertyDefinitionId=""cmis:isImmutable"">
10256     <cmis:value>>false</cmis:value>
10257     </cmis:propertyBoolean>
10258     <cmis:propertyString localName=""rep-cmis:checkinComment""
10259 propertyDefinitionId=""cmis:checkinComment"">
10260     <cmis:value>Checkin comment</cmis:value>
10261     </cmis:propertyString>
10262     <cmis:propertyString localName=""rep-cmis:versionLabel""
10263 propertyDefinitionId=""cmis:versionLabel"">
10264     <cmis:value>0.1</cmis:value>
10265     </cmis:propertyString>

```

```

10266 |         <cmis:propertyString localName="__rep-
10267 | cmis:contentStreamMimeType__"
10268 | propertyDefinitionId="__cmis:contentStreamMimeType__">
10269 |         <cmis:value>text/plain</cmis:value>
10270 |     </cmis:propertyString>
10271 |     <cmis:propertyString localName="__rep-
10272 | cmis:contentStreamFileName__"
10273 | propertyDefinitionId="__cmis:contentStreamFileName__">
10274 |         <cmis:value>text.txt</cmis:value>
10275 |     </cmis:propertyString>
10276 |     <cmis:propertyInteger localName="__rep-
10277 | cmis:contentStreamLength__" propertyDefinitionId="__cmis:contentStreamLength__">
10278 |         <cmis:value>4234</cmis:value>
10279 |     </cmis:propertyInteger>
10280 |     <cmis:propertyString displayName="__Keywords for Document__"
10281 | localName="__keywords__" propertyDefinitionId="__keywords__">
10282 |         <cmis:value>document</cmis:value>
10283 |         <cmis:value>example</cmis:value>
10284 |         <cmis:value>sample</cmis:value>
10285 |         <cmis:value>cmis</cmis:value>
10286 |     </cmis:propertyString>
10287 | </cmis:properties>
10288 | <cmis:changeEventInfo>
10289 |     <cmis:changeType>updated</cmis:changeType>
10290 |     <cmis:changeTime>2010-01-25T10:20:59.271-
10291 | 08:00</cmis:changeTime>
10292 | </cmis:changeEventInfo>
10293 | </cmisra:object>
10294 |     <cmisra:pathSegment>invoice.pdf</cmisra:pathSegment>
10295 | </atom:entry>
10296 | </atom:feed>

```

10297

10298 Please also see the example documents included with the schema.

10299 **3.9.2.1 GET**

10300 The following optional parameters may be supplied:

- 10301 • filter
- 10302 • maxItems
- 10303 • includeACL
- 10304 • includePolicyIds
- 10305 • includeProperties
- 10306 • changeLogToken: If this parameter is specified, start the changes from the specified token. The
- 10307 changeLogToken is embedded in the paging link relations for normal iteration through the change
- 10308 list.

10309 **3.9.3 Folder Descendants**

10310 This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available
10311 via the link relation down with the application/cmistree+xml media type. Please see the Hierarchical Atom
10312 Entries for more information on format.

10313

10314 If a repository does not support capabilityGetDescendants, then these resources SHOULD NOT be
10315 exposed.

10316

10317 CMIS Services:

- 10318 GET: getDescendants
- 10319 DELETE: deleteTree
- 10320 Media Type: application/atom+xml;type=feed
- 10321
- 10322 Link Relations:
- 10323 • service: Points to service document containing the CMIS repository. The service document
 - 10324 MUST contain only one workspace element.
 - 10325 ○ Media Type: application/atomsvc+xml
 - 10326 • via: points to the atom entry of the folder generating this collection
 - 10327 • up: points to the atom entry document for this folder's parent
 - 10328 ○ Media Type: application/atom+xml;type=entry
 - 10329 ○ If the root folder, this link relation MUST not be included.
 - 10330 • down:
 - 10331 ○ points to the atom feed document representing the children feed for this same folder with
 - 10332 media type of application/atom+xml;type=entry
 - 10333 ○ Since this is the descendants, the descendants link SHOULD NOT be included
 - 10334 • paging link relations MAY be included as appropriate: first, next, previous, last
 - 10335 ○ Repositories may support these paging link relations on a particular cmisra:children
 - 10336 element.
 - 10337 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

10339 The following CMIS Atom extension element MAY be included inside the atom feed:

- 10340 • cmisra:numItems

10341

10342 The following CMIS Atom extension element MUST be included inside the atom entries:

- 10343 • cmisra:object inside atom:entry
- 10344 • cmisra:pathSegment inside atom:entry
- 10345 • cmisra:children inside atom:entry

10346

10347 Example:

```

10348 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
10349 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
10350 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
10351 xmlns:atom="http://www.w3.org/2005/Atom"
10352 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
10353 open.org/ns/cmis/restatom/200908/">
10354   <atom:title type="text">Feed for folder1</atom:title>
10355   <atom:author>
10356     <atom:name>Al Brown</atom:name>
10357     <atom:uri>http://www.ibm.com/</atom:uri>
10358     <atom:email>albertcbrown@us.ibm.com</atom:email>
10359   </atom:author>
10360   <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
10361   <atom:id>urn:uuid:cb0a47d4-8d09-46f9-9b09-584acad684af</atom:id>
10362   <atom:link type="application/atom+xml;type=feed" rel="self"
10363 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
10364 040ec0c7c05a/3"/>
10365   <atom:link type="application/atomsvc+xml" rel="service"
10366 href="http://cmisexample.oasis-open.org/rep1//service"/>

```

```

10367 |     <atom:link type=""application/atom+xml;type=entry" rel=""via"
10368 | href=""http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
10369 | 040ec0c7c05a"/>
10370 |     <atom:link type=""application/atom+xml;type=feed"
10371 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
10372 | href=""http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
10373 | 040ec0c7c05a/foldertree"/>
10374 |     <atom:link type=""application/atom+xml;type=feed" rel=""down"
10375 | href=""http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
10376 | 040ec0c7c05a/children"/>
10377 |     <atom:link type=""application/atom+xml;type=entry" rel=""up"
10378 | href=""http://cmisexample.oasis-open.org/rep1/03dcf5b8-5f82-45a1-b276-
10379 | 44d88069eec3"/>
10380 |     <cmisra:numItems>1</cmisra:numItems>
10381 |     <atom:entry>
10382 |         <atom:author>
10383 |             <atom:name>Al Brown</atom:name>
10384 |             <atom:uri>http://www.ibm.com/</atom:uri>
10385 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
10386 |         </atom:author>
10387 |         <atom:content src=""http://cmisexample.oasis-open.org/rep1/8e5a512c-
10388 | 8f2d-4387-a283-f3f30bbc312f"/>
10389 |         <atom:id>urn:uuid:8e5a512c-8f2d-4387-a283-f3f30bbc312f</atom:id>
10390 |         <atom:title type=""text">CMIS Example Folder as Customer
10391 | type</atom:title>
10392 |         <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
10393 |         <atom:link rel=""self" href=""http://cmisexample.oasis-
10394 | open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
10395 |         <atom:link rel=""edit" href=""http://cmisexample.oasis-
10396 | open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
10397 |         <atom:link type=""application/cmisis+xml;type=allowableActions"
10398 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
10399 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10400 | f3f30bbc312f/allowableactions"/>
10401 |         <atom:link type=""application/atom+xml;type=entry"
10402 | rel=""describedby" href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-
10403 | 8f2d-4387-a283-f3f30bbc312f/type"/>
10404 |         <atom:link type=""application/atomsvc+xml" rel=""service"
10405 | href=""http://cmisexample.oasis-open.org/rep1//service"/>
10406 |         <atom:published>2010-01-25T10:20:59.380-08:00</atom:published>
10407 |         <atom:summary type=""html">HTML summary of Entry 8e5a512c-8f2d-4387-
10408 | a283-f3f30bbc312f</atom:summary>
10409 |         <atom:link type=""application/atom+xml;type=entry" rel=""up"
10410 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10411 | f3f30bbc312f/up"/>
10412 |         <atom:link type=""application/atom+xml;type=feed" rel=""down"
10413 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10414 | f3f30bbc312f/children"/>
10415 |         <atom:link type=""application/cmistree+xml" rel=""down"
10416 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10417 | f3f30bbc312f/tree"/>
10418 |         <atom:link type=""application/atom+xml;type=feed"
10419 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
10420 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10421 | f3f30bbc312f/foldertree"/>
10422 |         <atom:link type=""application/atom+xml;type=feed"
10423 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
10424 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10425 | f3f30bbc312f/relationships"/>
10426 |         <atom:link type=""application/atom+xml;type=feed"
10427 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
10428 | href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10429 | f3f30bbc312f/policies"/>

```

```

10430 |         <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
10431 | open.org/ns/cmisacl/link/200908/acl"" href=""http://cmisexample.oasis-
10432 | open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f/acl"" />
10433 |         <cmisra:object>
10434 |             <cmis:properties>
10435 |                 <cmis:propertyId localName=""rep-cmis:objectId""
10436 | propertyDefinitionId=""cmis:objectId"">
10437 |                     <cmis:value>8e5a512c-8f2d-4387-a283-
10438 | f3f30bbc312f</cmis:value>
10439 |                 </cmis:propertyId>
10440 |                 <cmis:propertyId localName=""rep-cmis:objectTypeId""
10441 | propertyDefinitionId=""cmis:objectTypeId"">
10442 |                     <cmis:value>customer</cmis:value>
10443 |                 </cmis:propertyId>
10444 |                 <cmis:propertyString localName=""rep-cmis:name""
10445 | propertyDefinitionId=""cmis:name"">
10446 |                     <cmis:value>CMIS Example Folder as Customer
10447 | type</cmis:value>
10448 |                 </cmis:propertyString>
10449 |                 <cmis:propertyDateTime localName=""rep-cmis:creationDate""
10450 | propertyDefinitionId=""cmis:creationDate"">
10451 |                     <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
10452 |                 </cmis:propertyDateTime>
10453 |                 <cmis:propertyDateTime localName=""rep-
10454 | cmis:lastModificationDate""
10455 | propertyDefinitionId=""cmis:lastModificationDate"">
10456 |                     <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
10457 |                 </cmis:propertyDateTime>
10458 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
10459 | propertyDefinitionId=""cmis:baseTypeId"">
10460 |                     <cmis:value>cmis:folder</cmis:value>
10461 |                 </cmis:propertyId>
10462 |                 <cmis:propertyString localName=""rep-cmis:lastModifiedBy""
10463 | propertyDefinitionId=""cmis:lastModifiedBy"">
10464 |                     <cmis:value>Al Brown</cmis:value>
10465 |                 </cmis:propertyString>
10466 |                 <cmis:propertyString localName=""rep-cmis:createdBy""
10467 | propertyDefinitionId=""cmis:createdBy"">
10468 |                     <cmis:value>Al Brown</cmis:value>
10469 |                 </cmis:propertyString>
10470 |                 <cmis:propertyId localName=""rep-cmis:parentId""
10471 | propertyDefinitionId=""cmis:parentId"">
10472 |                     <cmis:value>8e5a512c-8f2d-4387-a283-
10473 | f3f30bbc312fup</cmis:value>
10474 |                 </cmis:propertyId>
10475 |             </cmis:properties>
10476 |         </cmisra:object>
10477 |         <cmisra:pathSegment>customer</cmisra:pathSegment>
10478 |         <cmisra:children>
10479 |             <atom:feed>
10480 |                 <atom:title type=""text"">CMIS Example Folder as Customer
10481 | type</atom:title>
10482 |                 <atom:author>
10483 |                     <atom:name>Al Brown</atom:name>
10484 |                     <atom:uri>http://www.ibm.com/</atom:uri>
10485 |                     <atom:email>albertcbrown@us.ibm.com</atom:email>
10486 |                 </atom:author>
10487 |                 <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
10488 |                 <atom:id>urn:uuid:67ee5e9f-d2e3-457d-9dec-
10489 | be718e780452</atom:id>
10490 |                 <atom:link type=""application/atom+xml;type=feed""
10491 | rel=""self"" href=""http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-
10492 | a283-f3f30bbc312f/3"" />

```



```

10493 |         <atom:link type="application/atomsvc+xml" rel="service"
10494 | href="http://cmisexample.oasis-open.org/rep1//service"/>
10495 |         <atom:link type="application/atom+xml;type=entry"
10496 | rel="via" href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-
10497 | a283-f3f30bbc312f"/>
10498 |         <atom:link type="application/atom+xml;type=feed"
10499 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
10500 | href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
10501 | f3f30bbc312f/foldertree"/>
10502 |         <atom:link type="application/atom+xml;type=feed"
10503 | rel="down" href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-
10504 | a283-f3f30bbc312f/children"/>
10505 |         <atom:link type="application/atom+xml;type=entry" rel="up"
10506 | href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
10507 | 040ec0c7c05a"/>
10508 |         <cmisra:numItems>1</cmisra:numItems>
10509 |         <atom:entry>
10510 |             <atom:author>
10511 |                 <atom:name>Al Brown</atom:name>
10512 |                 <atom:uri>http://www.ibm.com/</atom:uri>
10513 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
10514 |             </atom:author>
10515 |             <atom:content src="http://cmisexample.oasis-
10516 | open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
10517 |             <atom:id>urn:uuid:8c2dbba5-ea33-469f-a052-
10518 | 9f01e636c72a</atom:id>
10519 |             <atom:title type="text">CMIS Example Doc as Invoice
10520 | type</atom:title>
10521 |             <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
10522 |             <atom:link rel="self" href="http://cmisexample.oasis-
10523 | open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
10524 |             <atom:link rel="edit" href="http://cmisexample.oasis-
10525 | open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
10526 |             <atom:link
10527 | type="application/cmisis+xml;type=allowableActions" rel="http://docs.oasis-
10528 | open.org/ns/cmisis/link/200908/allowableactions"
10529 | href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
10530 | 9f01e636c72a/allowableactions"/>
10531 |             <atom:link type="application/atom+xml;type=entry"
10532 | rel="describedby" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-
10533 | ea33-469f-a052-9f01e636c72a/type"/>
10534 |             <atom:link type="application/atomsvc+xml"
10535 | rel="service" href="http://cmisexample.oasis-open.org/rep1//service"/>
10536 |             <atom:published>2010-01-25T10:20:59.380-
10537 | 08:00</atom:published>
10538 |             <atom:summary type="html">HTML summary of Entry
10539 | 8c2dbba5-ea33-469f-a052-9f01e636c72a</atom:summary>
10540 |             <atom:link rel="edit-media"
10541 | href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
10542 | 9f01e636c72a/edit-media"/>
10543 |             <atom:link rel="alternate"
10544 | href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
10545 | 9f01e636c72a/alternate"/>
10546 |             <atom:link type="application/atom+xml;type=feed"
10547 | rel="up" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-
10548 | a052-9f01e636c72a/parents"/>
10549 |             <atom:link type="application/atom+xml;type=feed"
10550 | rel="version-history" href="http://cmisexample.oasis-
10551 | open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a/allversions"/>
10552 |             <atom:link type="application/atom+xml;type=entry"
10553 | rel="current-version" href="http://cmisexample.oasis-
10554 | open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a/latest"/>

```

```

10555 |         <atom:link type=""application/atom+xml;type=feed""
10556 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships""
10557 | href=""http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
10558 | 9f01e636c72a/relationships""/>
10559 |         <atom:link type=""application/atom+xml;type=feed""
10560 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies""
10561 | href=""http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
10562 | 9f01e636c72a/policies""/>
10563 |         <atom:link type=""application/cmisisacl+xml""
10564 | rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/acl""
10565 | href=""http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
10566 | 9f01e636c72a/acl""/>
10567 |         <cmisra:object>
10568 |             <cmis:properties>
10569 |                 <cmis:propertyId localName=""rep-cmis:objectId""
10570 | propertyDefinitionId=""cmis:objectId"">
10571 | <cmis:value>8c2dbba5-ea33-469f-a052-9f01e636c72a</cmis:value>
10572 |                 </cmis:propertyId>
10573 |                 <cmis:propertyId localName=""rep-
10574 | cmis:objectTypeId"" propertyDefinitionId=""cmis:objectTypeId"">
10575 | <cmis:value>invoice</cmis:value>
10576 |                 </cmis:propertyId>
10577 |                 <cmis:propertyString localName=""rep-cmis:name""
10578 | propertyDefinitionId=""cmis:name"">
10579 | <cmis:value>CMIS Example Doc as Invoice type</cmis:value>
10580 |                 </cmis:propertyString>
10581 |                 <cmis:propertyDateTime localName=""rep-
10582 | cmis:creationDate"" propertyDefinitionId=""cmis:creationDate"">
10583 | <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
10584 |                 </cmis:propertyDateTime>
10585 |                 <cmis:propertyDateTime localName=""rep-
10586 | cmis:lastModificationDate""
10587 | propertyDefinitionId=""cmis:lastModificationDate"">
10588 | <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
10589 |                 </cmis:propertyDateTime>
10590 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
10591 | propertyDefinitionId=""cmis:baseTypeId"">
10592 | <cmis:value>cmis:document</cmis:value>
10593 |                 </cmis:propertyId>
10594 |                 <cmis:propertyString localName=""rep-
10595 | cmis:lastModifiedBy"" propertyDefinitionId=""cmis:lastModifiedBy"">
10596 | <cmis:value>Al Brown</cmis:value>
10597 |                 </cmis:propertyString>
10598 |                 <cmis:propertyString localName=""rep-
10599 | cmis:createdBy"" propertyDefinitionId=""cmis:createdBy"">
10600 | <cmis:value>Al Brown</cmis:value>
10601 |                 </cmis:propertyString>
10602 |                 <cmis:propertyBoolean localName=""rep-
10603 | cmis:isLatestVersion"" propertyDefinitionId=""cmis:isLatestVersion"">
10604 | <cmis:value>>true</cmis:value>
10605 |                 </cmis:propertyBoolean>
10606 |                 <cmis:propertyBoolean localName=""rep-
10607 | cmis:isVersionSeriesCheckedOut""
10608 | propertyDefinitionId=""cmis:isVersionSeriesCheckedOut"">
10609 | <cmis:value>>false</cmis:value>
10610 |                 </cmis:propertyBoolean>
10611 |                 <cmis:propertyBoolean localName=""rep-
10612 | cmis:isMajorVersion"" propertyDefinitionId=""cmis:isMajorVersion"">
10613 | <cmis:value>>false</cmis:value>
10614 |                 </cmis:propertyBoolean>
10615 |                 <cmis:propertyBoolean localName=""rep-
10616 | cmis:isLatestMajorVersion""
10617 | propertyDefinitionId=""cmis:isLatestMajorVersion"">
10618 | <cmis:value>>false</cmis:value>

```

```

10619         </cmis:propertyBoolean>
10620         <cmis:propertyBoolean localName=""_rep-
10621 cmis:isImmutable" propertyDefinitionId=""_cmis:isImmutable">
10622 <cmis:value>>false</cmis:value>
10623         </cmis:propertyBoolean>
10624         <cmis:propertyString localName=""_rep-
10625 cmis:checkinComment" propertyDefinitionId=""_cmis:checkinComment">
10626 <cmis:value>Checkin comment</cmis:value>
10627         </cmis:propertyString>
10628         <cmis:propertyString localName=""_rep-
10629 cmis:versionLabel" propertyDefinitionId=""_cmis:versionLabel">
10630 <cmis:value>0.1</cmis:value>
10631         </cmis:propertyString>
10632         <cmis:propertyString localName=""_rep-
10633 cmis:contentStreamMimeType"
10634 propertyDefinitionId=""_cmis:contentStreamMimeType">
10635 <cmis:value>text/plain</cmis:value>
10636         </cmis:propertyString>
10637         <cmis:propertyString localName=""_rep-
10638 cmis:contentStreamFileName"
10639 propertyDefinitionId=""_cmis:contentStreamFileName">
10640 <cmis:value>text.txt</cmis:value>
10641         </cmis:propertyString>
10642         <cmis:propertyInteger localName=""_rep-
10643 cmis:contentStreamLength" propertyDefinitionId=""_cmis:contentStreamLength">
10644 <cmis:value>4234</cmis:value>
10645         </cmis:propertyInteger>
10646         <cmis:propertyString displayName=""_Keywords for
10647 Document" localName=""_keywords" propertyDefinitionId=""_keywords">
10648 <cmis:value>document</cmis:value>
10649 <cmis:value>example</cmis:value>
10650 <cmis:value>sample</cmis:value>
10651 <cmis:value>cmis</cmis:value>
10652         </cmis:propertyString>
10653     </cmis:properties>
10654 </cmisra:object>
10655     <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
10656 </atom:entry>
10657 </atom:feed>
10658 </cmisra:children>
10659 </atom:entry>
10660 </atom:feed>

```

10661

10662 Please also see the example documents included with the schema.

10663 **3.9.3.1 GET**

10664 The following arguments may be supplied. Please see the domain model for more information:

- 10665 • filter
- 10666 • depth
- 10667 • includeAllowableActions
- 10668 • includeRelationships
- 10669 • renditionFilter
- 10670 • includePathSegment

10671 **3.9.3.2 DELETE**

10672 This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the
10673 domain model for more information:

- 10674 • continueOnFailure
10675 • unfileObjects

10676
10677 Status Code:

- 10678 • 200 OK if successful. Body contains entity describing the status
10679 • 202 Accepted, if accepted but deletion not yet taking place
10680 • 204 No Content, if successful with no content
10681 • 403 Forbidden, if permission is denied
10682 • 401 Unauthorized, if not authenticated
10683 • 500 Internal Server Error. The body SHOULD contain an entity describing the status

10684
10685 If the delete method does not delete all items, invoking GET with infinite depth on this URI will return the
10686 items not deleted. Subsequent DELETE methods can be invoked on this URI.

10687 Note: If the repository does not implement get on this resource, or the canGetDescendants is false, there
10688 is no mechanism to identify the resources that were not removed.

10689 **3.9.4 Folder Tree**

10690 This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link
10691 relation foldertree with media type application/atom+xml;type=feed. Please see the Hierarchical Atom
10692 Entries for more information on format.

10693
10694 CMIS Services:

- 10695 GET: getFolderTree
10696 DELETE: deleteTree

10697 Media Type: application/atom+xml;type=feed

10698
10699 Link Relations:

- 10700 • service: Points to service document containing the CMIS repository. The service document
10701 MUST contain only one workspace element.
- 10702 ○ Media Type: application/atomsvc+xml
- 10703 • via: points to the atom entry of the folder generating this collection
- 10704 • up: points to the atom entry document of this folder's parent
- 10705 ○ If the root folder, this link relation MUST not be included.
 - 10706 ○ Media Type: application/atom+xml;type=entry
- 10707 • down:
- 10708 ○ application/atom+xml : Points to the atom feed document representing the children feed
10709 for this same folder
 - 10710 ○ application/cmistree+xml: Points to the descendants feed of the same folder. If a
10711 repository does not support capabilityGetDescendants, then this link SHOULD NOT be
10712 included.
- 10713 • paging link relations MAY be included as appropriate: first, next, previous, last
- 10714 ○ Repositories may support these paging link relations on a particular cmisra:children
10715 element.

10716

10717 This feed contains a set of atom entries for each sub-folder in the folder.

10718

10719 The following CMIS Atom extension element MAY be included inside the atom feed:

- 10720 • cmisra:numItems

10721

10722 The following CMIS Atom extension element MUST be included inside the atom entries:

- 10723 • cmisra:object inside atom:entry
- 10724 • cmisra:pathSegment inside atom:entry
- 10725 • cmisra:children inside atom:entry

10726

10727 Example:

```
10728 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
10729 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
10730 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
10731 xmlns:atom="http://www.w3.org/2005/Atom"
10732 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
10733 open.org/ns/cmisis/restatom/200908/">
10734   <atom:title type="text">FolderTree Feed of Folder1</atom:title>
10735   <atom:author>
10736     <atom:name>Al Brown</atom:name>
10737     <atom:uri>http://www.ibm.com/</atom:uri>
10738     <atom:email>albertcbrown@us.ibm.com</atom:email>
10739   </atom:author>
10740   <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
10741   <atom:id>urn:uuid:f87e5678-dd24-4214-9e71-635f060beb7d</atom:id>
10742   <atom:link type="application/atom+xml;type=feed" rel="self"
10743 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
10744 b65edc3e1854/3"/>
10745   <atom:link type="application/atomsvc+xml" rel="service"
10746 href="http://cmisexample.oasis-open.org/rep1//service"/>
10747   <atom:link type="application/atom+xml;type=entry" rel="via"
10748 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
10749 b65edc3e1854"/>
10750   <atom:link type="application/cmistree+xml" rel="down"
10751 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
10752 b65edc3e1854/tree"/>
10753   <atom:link type="application/atom+xml;type=feed" rel="down"
10754 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
10755 b65edc3e1854/children"/>
10756   <atom:link type="application/atom+xml;type=entry" rel="up"
10757 href="http://cmisexample.oasis-open.org/rep1/3056c4d7-4e16-49cb-a750-
10758 ad7a3844alaa"/>
10759   <cmisra:numItems>1</cmisra:numItems>
10760   <atom:entry>
10761     <atom:author>
10762       <atom:name>Al Brown</atom:name>
10763       <atom:uri>http://www.ibm.com/</atom:uri>
10764       <atom:email>albertcbrown@us.ibm.com</atom:email>
10765     </atom:author>
10766     <atom:content src="http://cmisexample.oasis-open.org/rep1/c7b5a83e-
10767 37b6-4f5a-b646-50892252a180"/>
10768     <atom:id>urn:uuid:c7b5a83e-37b6-4f5a-b646-50892252a180</atom:id>
10769     <atom:title type="text">Customer Folder</atom:title>
10770     <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
10771     <atom:link rel="self" href="http://cmisexample.oasis-
10772 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
10773     <atom:link rel="edit" href="http://cmisexample.oasis-
10774 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
```

```

10775 |         <atom:link type=""application/cmism+xml;type=allowableActions""
10776 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions""
10777 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10778 | 50892252a180/allowableactions""/>
10779 |         <atom:link type=""application/atom+xml;type=entry""
10780 | rel=""describedby"" href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-
10781 | 37b6-4f5a-b646-50892252a180/type""/>
10782 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
10783 | href=""http://cmisexample.oasis-open.org/rep1//service""/>
10784 |         <atom:published>2010-01-25T10:20:59.521-08:00</atom:published>
10785 |         <atom:summary type=""html"">HTML summary of Entry c7b5a83e-37b6-4f5a-
10786 | b646-50892252a180</atom:summary>
10787 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
10788 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10789 | 50892252a180/up""/>
10790 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
10791 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10792 | 50892252a180/children""/>
10793 |         <atom:link type=""application/cmistree+xml"" rel=""down""
10794 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10795 | 50892252a180/tree""/>
10796 |         <atom:link type=""application/atom+xml;type=feed""
10797 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/foldertree""
10798 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10799 | 50892252a180/foldertree""/>
10800 |         <atom:link type=""application/atom+xml;type=feed""
10801 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/relationships""
10802 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10803 | 50892252a180/relationships""/>
10804 |         <atom:link type=""application/atom+xml;type=feed""
10805 | rel=""http://docs.oasis-open.org/ns/cmism/link/200908/policies""
10806 | href=""http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
10807 | 50892252a180/policies""/>
10808 |         <atom:link type=""application/cmisacl+xml"" rel=""http://docs.oasis-
10809 | open.org/ns/cmism/link/200908/acl"" href=""http://cmisexample.oasis-
10810 | open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180/acl""/>
10811 |         <cmisra:object>
10812 |             <cmis:properties>
10813 |                 <cmis:propertyId localName=""rep-cmis:objectId""
10814 | propertyDefinitionId=""cmis:objectId"">
10815 |                     <cmis:value>c7b5a83e-37b6-4f5a-b646-
10816 | 50892252a180</cmis:value>
10817 |                 </cmis:propertyId>
10818 |                 <cmis:propertyId localName=""rep-cmis:objectTypeId""
10819 | propertyDefinitionId=""cmis:objectTypeId"">
10820 |                     <cmis:value>customer</cmis:value>
10821 |                 </cmis:propertyId>
10822 |                 <cmis:propertyString localName=""rep-cmis:name""
10823 | propertyDefinitionId=""cmis:name"">
10824 |                     <cmis:value>Customer Folder</cmis:value>
10825 |                 </cmis:propertyString>
10826 |                 <cmis:propertyDateTime localName=""rep-cmis:creationDate""
10827 | propertyDefinitionId=""cmis:creationDate"">
10828 |                     <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
10829 |                 </cmis:propertyDateTime>
10830 |                 <cmis:propertyDateTime localName=""rep-
10831 | cmis:lastModificationDate""
10832 | propertyDefinitionId=""cmis:lastModificationDate"">
10833 |                     <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
10834 |                 </cmis:propertyDateTime>
10835 |                 <cmis:propertyId localName=""rep-cmis:baseTypeId""
10836 | propertyDefinitionId=""cmis:baseTypeId"">
10837 |                     <cmis:value>cmis:folder</cmis:value>
10838 |                 </cmis:propertyId>

```

```

10839 |         <cmis:propertyString localName="__rep-cmis:lastModifiedBy__"
10840 | propertyDefinitionId="__cmis:lastModifiedBy__">
10841 |             <cmis:value>Al Brown</cmis:value>
10842 |         </cmis:propertyString>
10843 |         <cmis:propertyString localName="__rep-cmis:createdBy__"
10844 | propertyDefinitionId="__cmis:createdBy__">
10845 |             <cmis:value>Al Brown</cmis:value>
10846 |         </cmis:propertyString>
10847 |         <cmis:propertyId localName="__rep-cmis:parentId__"
10848 | propertyDefinitionId="__cmis:parentId__">
10849 |             <cmis:value>c7b5a83e-37b6-4f5a-b646-
10850 | 50892252a180up</cmis:value>
10851 |         </cmis:propertyId>
10852 |     </cmis:properties>
10853 | </cmisra:object>
10854 | <cmisra:pathSegment>customer</cmisra:pathSegment>
10855 | </atom:entry>
10856 | </atom:feed>

```

10857

10858 Please also see the example documents included with the schema.

10859 **3.9.4.1 GET**

10860 The following arguments may be supplied. Please see the domain model for more information:

- 10861 • filter
- 10862 • depth
- 10863 • includeAllowableActions
- 10864 • includeRelationships
- 10865 • renditionFilter

10866 **3.9.4.2 DELETE**

10867 This is the same as DELETE on Folder Descendants. Please see that section.

10868 **3.9.5 AllVersions Feed**

10869 This is a feed comprised of all the versions of the given document.

10870 CMIS Services:

- 10871 GET: getAllVersions
- 10872 DELETE: deleteAllVersions

10873 Media Type: application/atom+xml;type=feed

10874

10875 The feed SHOULD contain the newest versions at the beginning of the feed.

10876

10877 Link Relations:

- 10878 • service: Points to service document containing the CMIS repository. The service document
 - 10879 MUST contain only one workspace element.
 - 10880 ○ Media Type: application/atomsvc+xml
- 10881 • via: points to the atom entry of the resource generating this collection
- 10882 • paging link relations as appropriate: first, next, previous, last

10883

10884 This feed contains a set of atom entries for each version in the version series

- 10885 • cmisra:object inside atom:entry
- 10886 • cmisra:children inside atom:entry if atom:entry represents a CMIS Folder

10887
10888 Example:

```

10889 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
10890 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
10891 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
10892 xmlns:atom="http://www.w3.org/2005/Atom"
10893 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
10894 open.org/ns/cmisis/restatom/200908/"
10895 <atom:title type="text">AllVersions for Document e8abd7cd-b9ec-4dba-
10896 9eaa-1bce2ae9977f</atom:title>
10897 <atom:author>
10898 <atom:name>Al Brown</atom:name>
10899 <atom:uri>http://www.ibm.com/</atom:uri>
10900 <atom:email>albertcbrown@us.ibm.com</atom:email>
10901 </atom:author>
10902 <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
10903 <atom:id>urn:uuid:5dc3d1c1-3e85-4720-acf8-cf98c96a5830</atom:id>
10904 <atom:link type="application/atom+xml;type=feed" rel="self"
10905 href="http://cmisexample.oasis-open.org/rep1/e8abd7cd-b9ec-4dba-9eaa-
10906 1bce2ae9977f/3"/>
10907 <atom:link type="application/atomsvc+xml" rel="service"
10908 href="http://cmisexample.oasis-open.org/rep1//service"/>
10909 <atom:link type="application/atom+xml;type=entry" rel="via"
10910 href="http://cmisexample.oasis-open.org/rep1/e8abd7cd-b9ec-4dba-9eaa-
10911 1bce2ae9977f"/>
10912 <cmisra:numItems>1</cmisra:numItems>
10913 <atom:entry>
10914 <atom:author>
10915 <atom:name>Al Brown</atom:name>
10916 <atom:uri>http://www.ibm.com/</atom:uri>
10917 <atom:email>albertcbrown@us.ibm.com</atom:email>
10918 </atom:author>
10919 <atom:content src="http://cmisexample.oasis-open.org/rep1/197033f2-
10920 ac11-4911-b5a3-60781fa5c281"/>
10921 <atom:id>urn:uuid:197033f2-ac11-4911-b5a3-60781fa5c281</atom:id>
10922 <atom:title type="text">Invoice (Version1)</atom:title>
10923 <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
10924 <atom:link rel="self" href="http://cmisexample.oasis-
10925 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281"/>
10926 <atom:link rel="edit" href="http://cmisexample.oasis-
10927 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281"/>
10928 <atom:link type="application/cmisis+xml;type=allowableActions"
10929 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
10930 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
10931 60781fa5c281/allowableactions"/>
10932 <atom:link type="application/atom+xml;type=entry"
10933 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/197033f2-
10934 ac11-4911-b5a3-60781fa5c281/type"/>
10935 <atom:link type="application/atomsvc+xml" rel="service"
10936 href="http://cmisexample.oasis-open.org/rep1//service"/>
10937 <atom:published>2010-01-25T10:20:58.896-08:00</atom:published>
10938 <atom:summary type="html">HTML summary of Entry 197033f2-ac11-4911-
10939 b5a3-60781fa5c281</atom:summary>
10940 <atom:link rel="edit-media" href="http://cmisexample.oasis-
10941 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/edit-media"/>
10942 <atom:link rel="alternate" href="http://cmisexample.oasis-
10943 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/alternate"/>
10944 <atom:link type="application/atom+xml;type=feed" rel="up"
10945 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
10946 60781fa5c281/parents"/>

```



```

10947     <atom:link type=""application/atom+xml;type=feed"" rel=""version-
10948 history"" href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-
10949 b5a3-60781fa5c281/allversions"" />
10950     <atom:link type=""application/atom+xml;type=entry"" rel=""current-
10951 version"" href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-
10952 b5a3-60781fa5c281/latest"" />
10953     <atom:link type=""application/atom+xml;type=feed""
10954 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/relationships""
10955 href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
10956 60781fa5c281/relationships"" />
10957     <atom:link type=""application/atom+xml;type=feed""
10958 rel=""http://docs.oasis-open.org/ns/cmisis/link/200908/policies""
10959 href=""http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
10960 60781fa5c281/policies"" />
10961     <atom:link type=""application/cmisisacl+xml"" rel=""http://docs.oasis-
10962 open.org/ns/cmisis/link/200908/acl"" href=""http://cmisexample.oasis-
10963 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/acl"" />
10964     <cmisisra:object>
10965         <cmisisra:properties>
10966             <cmisisra:propertyId localName=""rep-cmisis:objectId""
10967 propertyDefinitionId=""cmisis:objectId"">
10968                 <cmisisra:value>197033f2-ac11-4911-b5a3-
10969 60781fa5c281</cmisisra:value>
10970             </cmisisra:propertyId>
10971         </cmisisra:properties>
10972     </cmisisra:object>
10973 </atom:entry>
10974 </atom:feed>

```

10975

10976 Please also see the example documents included with the schema.

10977 **3.9.5.1 GET**

10978 The following arguments may be supplied. Please see the domain model for more information:

- 10979 • filter
- 10980 • includeAllowableActions

10981 **3.9.5.2 DELETE**

10982 This removes the entire version history of the document.

10983

10984 Success HTTP code: 204

10985 **3.9.6 Type Descendants Feed**

10986 This is a feed described in the service document that contains all the types under a specific type in the
10987 repository to a specific depth. If no parent type is specified, then the base types and their descendants
10988 are returned in the feed which is equivalent to all types in the repository if depth is infinite. The link
10989 relation is <http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>.

10990

10991 Types are nested using the CMIS hierarchy extension. Please see section 3.4.3.2 [Hierarchy Navigation](#)
10992 [Internet Draft Link Relations](#)~~[Hierarchy Navigation](#)~~ [Internet Draft Link Relations](#).

10993

10994 CMIS Services:

10995 GET: getTypeDescendants

10996

10997 Media Type: application/atom+xml;type=feed

10998

10999 Link Relations:

- 11000 • service: Points to service document containing the CMIS repository. The service document
- 11001 MUST contain only one workspace element.
 - 11002 ○ Media Type: application/atomsvc+xml
- 11003 • via: points to the type definition whose descendants represent this feed. This link is not present if
- 11004 no parent type is specified.
- 11005 • down: points to the children feed for the same type
- 11006 • up: points to the parent type definition
 - 11007 ○ If this is a descendants feed for a base object type, this link is not present.

11008

11009 The following CMIS Atom extension element MAY be included inside the atom feed:

- 11010 • cmisra:numItems

11011

11012 Example:

```
11013 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
11014 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
11015 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
11016 xmlns:atom="http://www.w3.org/2005/Atom"
11017 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
11018 open.org/ns/cmisis/restatom/200908/">
11019   <atom:title type="text">Base Types</atom:title>
11020   <atom:author>
11021     <atom:name>Al Brown</atom:name>
11022     <atom:uri>http://www.ibm.com/</atom:uri>
11023     <atom:email>albertcbrown@us.ibm.com</atom:email>
11024   </atom:author>
11025   <atom:updated>2010-01-25T10:20:59.911-08:00</atom:updated>
11026   <atom:id>urn:uuid:c5d3d357-33ec-47c1-8436-563e0d94d2e5</atom:id>
11027   <atom:link type="application/atom+xml;type=feed" rel="self"
11028 href="http://cmisexample.oasis-open.org/rep1//3"/>
11029   <atom:link type="application/atomsvc+xml" rel="service"
11030 href="http://cmisexample.oasis-open.org/rep1//service"/>
11031   <atom:link type="application/atom+xml;type=entry" rel="via"
11032 href="http://cmisexample.oasis-open.org/rep1/">
11033   <atom:link type="application/atom+xml;type=feed" rel="down"
11034 href="http://cmisexample.oasis-open.org/rep1//children"/>
11035   <cmisra:numItems>1</cmisra:numItems>
11036   <atom:entry>
11037     <atom:author>
11038       <atom:name>Al Brown</atom:name>
11039       <atom:uri>http://www.ibm.com/</atom:uri>
11040       <atom:email>albertcbrown@us.ibm.com</atom:email>
11041     </atom:author>
11042     <atom:content>Type Definition for cmis:document</atom:content>
11043     <atom:id>http://cmisexample.oasis-
11044 open.org/rep1/type/cmisis:document</atom:id>
11045     <atom:link type="application/atom+xml;type=entry" rel="self"
11046 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document"/>
11047     <atom:link type="application/atomsvc+xml" rel="service"
11048 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document"/>
11049     <atom:link type="application/atom+xml;type=entry"
11050 rel="describedby" href="http://cmisexample.oasis-
11051 open.org/rep1/type/cmisis:document"/>
11052     <atom:link type="application/atom+xml;type=entry" rel="up"
11053 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document/parent"/>
```

```

11054 |         <atom:link type=""application/atom+xml;type=feed"" rel=""down""
11055 | href=""http://cmisexample.oasis-
11056 | open.org/rep1/type/cmisis:document/children/flat"">
11057 |         <atom:link type=""application/cmistree+xml"" rel=""down""
11058 | href=""http://cmisexample.oasis-
11059 | open.org/rep1/type/cmisis:document/children/tree"">
11060 |         <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
11061 |         <atom:summary type=""html"">HTML summary of Type Definition
11062 | cmis:document</atom:summary>
11063 |         <atom:title type=""text"">Type Definition - cmis:document</atom:title>
11064 | <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
11065 | <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
11066 | <cmisra:type xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
11067 | xsi:type=""cmis:cmisTypeDocumentDefinitionType"" cmisra:id=""cmis:document"">
11068 |     <cmis:id>dtcmis:document</cmis:id>
11069 |     <cmis:localName>myrepname-cmis:document</cmis:localName>
11070 |     <cmis:localNamespace xsi:nil=""true"">
11071 |     <cmis:displayName>cmis:document</cmis:displayName>
11072 |     <cmis:queryName>cmis:document</cmis:queryName>
11073 |     <cmis:description>Description for type definition
11074 | cmis:document</cmis:description>
11075 |     <cmis:baseId>cmis:document</cmis:baseId>
11076 |     <cmis:parentId>parent</cmis:parentId>
11077 |     <cmis:creatable>true</cmis:creatable>
11078 |     <cmis:fileable>true</cmis:fileable>
11079 |     <cmis:queryable>false</cmis:queryable>
11080 |     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11081 |
11082 | <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11083 |     <cmis:controllablePolicy>true</cmis:controllablePolicy>
11084 |     <cmis:controllableACL>true</cmis:controllableACL>
11085 |     <cmis:versionable>true</cmis:versionable>
11086 |     <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
11087 | </cmisra:type>
11088 | <cmisra:children>
11089 |     <atom:feed>
11090 |         <atom:title type=""text"">Children for Document</atom:title>
11091 |         <atom:author>
11092 |             <atom:name>Al Brown</atom:name>
11093 |             <atom:uri>http://www.ibm.com/</atom:uri>
11094 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
11095 |         </atom:author>
11096 |         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
11097 |         <atom:id>urn:uuid:6f1cdc44-bd89-41c0-8fad-
11098 | 89f3ad0f8f30</atom:id>
11099 |         <atom:link type=""application/atom+xml;type=feed""
11100 | rel=""self"" href=""http://cmisexample.oasis-open.org/rep1/cmisis:document/3"">
11101 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
11102 | href=""http://cmisexample.oasis-open.org/rep1//service"">
11103 |         <atom:link type=""application/atom+xml;type=entry""
11104 | rel=""via"" href=""http://cmisexample.oasis-open.org/rep1/cmisis:document"">
11105 |         <atom:link type=""application/atom+xml;type=feed""
11106 | rel=""down"" href=""http://cmisexample.oasis-
11107 | open.org/rep1/cmisis:document/children"">
11108 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
11109 | href=""http://cmisexample.oasis-open.org/rep1/document"">
11110 |         <cmisra:numItems>1</cmisra:numItems>
11111 |         <atom:entry>
11112 |             <atom:author>
11113 |                 <atom:name>Al Brown</atom:name>
11114 |                 <atom:uri>http://www.ibm.com/</atom:uri>
11115 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
11116 |             </atom:author>

```

```

11117         <atom:content>Type Definition for invoice-
11118 document</atom:content>
11119         <atom:id>http://cmisexample.oasis-
11120 open.org/repl/type/invoice-document</atom:id>
11121         <atom:link type="application/atom+xml;type=entry"
11122 rel="self" href="http://cmisexample.oasis-open.org/repl/type/invoice-
11123 document"/>
11124         <atom:link type="application/atomsvc+xml"
11125 rel="service" href="http://cmisexample.oasis-open.org/repl/type/invoice-
11126 document"/>
11127         <atom:link type="application/atom+xml;type=entry"
11128 rel="describedby" href="http://cmisexample.oasis-
11129 open.org/repl/type/cmis:document"/>
11130         <atom:link type="application/atom+xml;type=entry"
11131 rel="up" href="http://cmisexample.oasis-open.org/repl/type/invoice-
11132 document/parent"/>
11133         <atom:link type="application/atom+xml;type=feed"
11134 rel="down" href="http://cmisexample.oasis-open.org/repl/type/invoice-
11135 document/children/flat"/>
11136         <atom:link type="application/cmistree+xml" rel="down"
11137 href="http://cmisexample.oasis-open.org/repl/type/invoice-
11138 document/children/tree"/>
11139         <atom:published>2010-01-25T10:20:59.927-
11140 08:00</atom:published>
11141         <atom:summary type="html">HTML summary of Type
11142 Definition invoice-document</atom:summary>
11143         <atom:title type="text">Type Definition - invoice-
11144 document</atom:title>
11145         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
11146         <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
11147         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
11148 instance" xsi:type="cmis:cmisTypeDocumentDefinitionType"
11149 cmisra:id="invoice-document">
11150         <cmis:id>dtinvoice-document</cmis:id>
11151         <cmis:localName>myrepname-invoice-
11152 document</cmis:localName>
11153         <cmis:localNamespace xsi:nil="true"/>
11154         <cmis:displayName>invoice-document</cmis:displayName>
11155         <cmis:queryName>invoice-document</cmis:queryName>
11156         <cmis:description>Description for type definition
11157 invoice-document</cmis:description>
11158         <cmis:baseId>cmis:document</cmis:baseId>
11159         <cmis:parentId>parent</cmis:parentId>
11160         <cmis:creatable>true</cmis:creatable>
11161         <cmis:fileable>true</cmis:fileable>
11162         <cmis:queryable>false</cmis:queryable>
11163         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11164
11165         <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11166
11167         <cmis:controllablePolicy>true</cmis:controllablePolicy>
11168         <cmis:controllableACL>true</cmis:controllableACL>
11169         <cmis:versionable>true</cmis:versionable>
11170
11171         <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
11172         </cmisra:type>
11173     </atom:entry>
11174 </atom:feed>
11175 </cmisra:children>
11176 </atom:entry>
11177 <atom:entry>
11178   <atom:author>
11179     <atom:name>Al Brown</atom:name>
11180     <atom:uri>http://www.ibm.com/</atom:uri>

```

```

11181         <atom:email>albertcbrown@us.ibm.com</atom:email>
11182     </atom:author>
11183     <atom:content>Type Definition for cmis:folder</atom:content>
11184     <atom:id>http://cmisexample.oasis-
11185 open.org/repl/type/cmis:folder</atom:id>
11186     <atom:link type="application/atom+xml;type=entry" rel="self"
11187 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
11188     <atom:link type="application/atomsvc+xml" rel="service"
11189 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
11190     <atom:link type="application/atom+xml;type=entry"
11191 rel="describedby" href="http://cmisexample.oasis-
11192 open.org/repl/type/cmis:folder"/>
11193     <atom:link type="application/atom+xml;type=entry" rel="up"
11194 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/parent"/>
11195     <atom:link type="application/atom+xml;type=feed" rel="down"
11196 href="http://cmisexample.oasis-
11197 open.org/repl/type/cmis:folder/children/flat"/>
11198     <atom:link type="application/cmistree+xml" rel="down"
11199 href="http://cmisexample.oasis-
11200 open.org/repl/type/cmis:folder/children/tree"/>
11201     <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
11202     <atom:summary type="html">HTML summary of Type Definition
11203 cmis:folder</atom:summary>
11204     <atom:title type="text">Type Definition - cmis:folder</atom:title>
11205     <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
11206     <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
11207     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11208 xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="cmis:folder">
11209         <cmis:id>dtcmis:folder</cmis:id>
11210         <cmis:localName>myrepname-cmis:folder</cmis:localName>
11211         <cmis:localNamespace xsi:nil="true"/>
11212         <cmis:displayName>cmis:folder</cmis:displayName>
11213         <cmis:queryName>cmis:folder</cmis:queryName>
11214         <cmis:description>Description for type definition
11215 cmis:folder</cmis:description>
11216         <cmis:baseId>cmis:folder</cmis:baseId>
11217         <cmis:parentId>parent</cmis:parentId>
11218         <cmis:creatable>true</cmis:creatable>
11219         <cmis:fileable>true</cmis:fileable>
11220         <cmis:queryable>false</cmis:queryable>
11221         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11222
11223     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11224     <cmis:controllablePolicy>true</cmis:controllablePolicy>
11225     <cmis:controllableACL>true</cmis:controllableACL>
11226 </cmisra:type>
11227 <cmisra:children>
11228     <atom:feed>
11229         <atom:title type="text">Children for Folder</atom:title>
11230         <atom:author>
11231             <atom:name>Al Brown</atom:name>
11232             <atom:uri>http://www.ibm.com/</atom:uri>
11233             <atom:email>albertcbrown@us.ibm.com</atom:email>
11234         </atom:author>
11235         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
11236         <atom:id>urn:uuid:361a3ac1-f7f7-47cb-b941-
11237 ae1200213fe0</atom:id>
11238         <atom:link type="application/atom+xml;type=feed"
11239 rel="self" href="http://cmisexample.oasis-open.org/repl/cmis:folder/3"/>
11240         <atom:link type="application/atomsvc+xml" rel="service"
11241 href="http://cmisexample.oasis-open.org/repl//service"/>
11242         <atom:link type="application/atom+xml;type=entry"
11243 rel="via" href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>

```

```

11244 |         <atom:link type=""application/atom+xml;type=feed""
11245 | rel=""down"" href=""http://cmisexample.oasis-
11246 | open.org/repl/cmisa:folder/children""/>
11247 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
11248 | href=""http://cmisexample.oasis-open.org/repl/cmisa:folder""/>
11249 |         <cmisra:numItems>1</cmisra:numItems>
11250 |         <atom:entry>
11251 |             <atom:author>
11252 |                 <atom:name>Al Brown</atom:name>
11253 |                 <atom:uri>http://www.ibm.com/</atom:uri>
11254 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
11255 |             </atom:author>
11256 |             <atom:content>Type Definition for customer-
11257 | folder</atom:content>
11258 |             <atom:id>http://cmisexample.oasis-
11259 | open.org/repl/type/customer-folder</atom:id>
11260 |             <atom:link type=""application/atom+xml;type=entry""
11261 | rel=""self"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11262 | folder""/>
11263 |             <atom:link type=""application/atomsvc+xml""
11264 | rel=""service"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11265 | folder""/>
11266 |             <atom:link type=""application/atom+xml;type=entry""
11267 | rel=""describedby"" href=""http://cmisexample.oasis-
11268 | open.org/repl/type/cmisa:folder""/>
11269 |             <atom:link type=""application/atom+xml;type=entry""
11270 | rel=""up"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11271 | folder/parent""/>
11272 |             <atom:link type=""application/atom+xml;type=feed""
11273 | rel=""down"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11274 | folder/children/flat""/>
11275 |             <atom:link type=""application/cmistree+xml"" rel=""down""
11276 | href=""http://cmisexample.oasis-open.org/repl/type/customer-
11277 | folder/children/tree""/>
11278 |             <atom:published>2010-01-25T10:20:59.927-
11279 | 08:00</atom:published>
11280 |             <atom:summary type=""html"">HTML summary of Type
11281 | Definition customer-folder</atom:summary>
11282 |             <atom:title type=""text"">Type Definition - customer-
11283 | folder</atom:title>
11284 |             <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
11285 |             <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
11286 |             <cmisra:type xmlns:xsi=""http://www.w3.org/2001/XMLSchema-
11287 | instance"" xsi:type=""cmisa:cmisTypeFolderDefinitionType""
11288 | cmisra:id=""customer-folder"">
11289 |                 <cmis:id>dtcustomer-folder</cmis:id>
11290 |                 <cmis:localName>myrepname-customer-
11291 | folder</cmis:localName>
11292 |                 <cmis:localNamespace xsi:nil=""true""/>
11293 |                 <cmis:displayName>customer-folder</cmis:displayName>
11294 |                 <cmis:queryName>customer-folder</cmis:queryName>
11295 |                 <cmis:description>Description for type definition
11296 | customer-folder</cmis:description>
11297 |                 <cmis:baseId>cmisa:folder</cmis:baseId>
11298 |                 <cmis:parentId>parent</cmis:parentId>
11299 |                 <cmis:creatable>true</cmis:creatable>
11300 |                 <cmis:fileable>true</cmis:fileable>
11301 |                 <cmis:queryable>false</cmis:queryable>
11302 |                 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11303 |
11304 | <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11305 |
11306 | <cmis:controllablePolicy>true</cmis:controllablePolicy>
11307 | <cmis:controllableACL>true</cmis:controllableACL>

```

```

11308         </cmisra:type>
11309     </atom:entry>
11310 </atom:feed>
11311 </cmisra:children>
11312 </atom:entry>
11313 <atom:entry>
11314     <atom:author>
11315         <atom:name>Al Brown</atom:name>
11316         <atom:uri>http://www.ibm.com/</atom:uri>
11317         <atom:email>albertcbrown@us.ibm.com</atom:email>
11318     </atom:author>
11319     <atom:content>Type Definition for cmis:relationship</atom:content>
11320     <atom:id>http://cmisexample.oasis-
11321 open.org/repl/type/cmis:relationship</atom:id>
11322     <atom:link type=""application/atom+xml;type=entry"" rel=""self""
11323 href=""http://cmisexample.oasis-open.org/repl/type/cmis:relationship""/>
11324     <atom:link type=""application/atomsvc+xml"" rel=""service""
11325 href=""http://cmisexample.oasis-open.org/repl/type/cmis:relationship""/>
11326     <atom:link type=""application/atom+xml;type=entry""
11327 rel=""describedby"" href=""http://cmisexample.oasis-
11328 open.org/repl/type/cmis:relationship""/>
11329     <atom:link type=""application/atom+xml;type=entry"" rel=""up""
11330 href=""http://cmisexample.oasis-
11331 open.org/repl/type/cmis:relationship/parent""/>
11332     <atom:link type=""application/atom+xml;type=feed"" rel=""down""
11333 href=""http://cmisexample.oasis-
11334 open.org/repl/type/cmis:relationship/children/flat""/>
11335     <atom:link type=""application/cmistree+xml"" rel=""down""
11336 href=""http://cmisexample.oasis-
11337 open.org/repl/type/cmis:relationship/children/tree""/>
11338     <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
11339     <atom:summary type=""html"">HTML summary of Type Definition
11340 cmis:relationship</atom:summary>
11341     <atom:title type=""text"">Type Definition -
11342 cmis:relationship</atom:title>
11343     <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
11344     <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
11345     <cmisra:type xmlns:xsi=""http://www.w3.org/2001/XMLSchema-instance""
11346 xsi:type=""cmis:cmisTypeRelationshipDefinitionType""
11347 cmisra:id=""cmis:relationship"">
11348     <cmis:id>dtcmis:relationship</cmis:id>
11349     <cmis:localName>myrepname-cmis:relationship</cmis:localName>
11350     <cmis:localNamespace xsi:nil=""true""/>
11351     <cmis:displayName>cmis:relationship</cmis:displayName>
11352     <cmis:queryName>cmis:relationship</cmis:queryName>
11353     <cmis:description>Description for type definition
11354 cmis:relationship</cmis:description>
11355     <cmis:baseId>cmis:relationship</cmis:baseId>
11356     <cmis:parentId>parent</cmis:parentId>
11357     <cmis:creatable>true</cmis:creatable>
11358     <cmis:fileable>>false</cmis:fileable>
11359     <cmis:queryable>>false</cmis:queryable>
11360     <cmis:fulltextIndexed>>false</cmis:fulltextIndexed>
11361
11362     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11363     <cmis:controllablePolicy>true</cmis:controllablePolicy>
11364     <cmis:controllableACL>true</cmis:controllableACL>
11365     <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
11366     <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
11367     <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
11368 </cmisra:type>
11369 </cmisra:children>
11370 </atom:feed>

```

```

11371 |         <atom:title type=""text">Children for
11372 | Relationship</atom:title>
11373 |         <atom:author>
11374 |             <atom:name>Al Brown</atom:name>
11375 |             <atom:uri>http://www.ibm.com/</atom:uri>
11376 |             <atom:email>albertcbrown@us.ibm.com</atom:email>
11377 |         </atom:author>
11378 |         <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
11379 |         <atom:id>urn:uuid:9394ff3d-87c4-48c9-a951-
11380 | ba725560faac</atom:id>
11381 |         <atom:link type=""application/atom+xml;type=feed"
11382 | rel=""self"" href=""http://cmisexample.oasis-
11383 | open.org/repl/cmisis:relationship/3""/>
11384 |         <atom:link type=""application/atomsvc+xml"" rel=""service""
11385 | href=""http://cmisexample.oasis-open.org/repl//service""/>
11386 |         <atom:link type=""application/atom+xml;type=entry""
11387 | rel=""via"" href=""http://cmisexample.oasis-
11388 | open.org/repl/cmisis:relationship""/>
11389 |         <atom:link type=""application/atom+xml;type=feed"
11390 | rel=""down"" href=""http://cmisexample.oasis-
11391 | open.org/repl/cmisis:relationship/children""/>
11392 |         <atom:link type=""application/atom+xml;type=entry"" rel=""up""
11393 | href=""http://cmisexample.oasis-open.org/repl/cmisis:folder""/>
11394 |         <cmisra:numItems>1</cmisra:numItems>
11395 |         <atom:entry>
11396 |             <atom:author>
11397 |                 <atom:name>Al Brown</atom:name>
11398 |                 <atom:uri>http://www.ibm.com/</atom:uri>
11399 |                 <atom:email>albertcbrown@us.ibm.com</atom:email>
11400 |             </atom:author>
11401 |             <atom:content>Type Definition for customer-
11402 | relationship</atom:content>
11403 |             <atom:id>http://cmisexample.oasis-
11404 | open.org/repl/type/customer-relationship</atom:id>
11405 |             <atom:link type=""application/atom+xml;type=entry""
11406 | rel=""self"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11407 | relationship""/>
11408 |             <atom:link type=""application/atomsvc+xml""
11409 | rel=""service"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11410 | relationship""/>
11411 |             <atom:link type=""application/atom+xml;type=entry""
11412 | rel=""describedby"" href=""http://cmisexample.oasis-
11413 | open.org/repl/type/cmisis:relationship""/>
11414 |             <atom:link type=""application/atom+xml;type=entry""
11415 | rel=""up"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11416 | relationship/parent""/>
11417 |             <atom:link type=""application/atom+xml;type=feed""
11418 | rel=""down"" href=""http://cmisexample.oasis-open.org/repl/type/customer-
11419 | relationship/children/flat""/>
11420 |             <atom:link type=""application/cmistree+xml"" rel=""down""
11421 | href=""http://cmisexample.oasis-open.org/repl/type/customer-
11422 | relationship/children/tree""/>
11423 |             <atom:published>2010-01-25T10:20:59.943-
11424 | 08:00</atom:published>
11425 |             <atom:summary type=""html"">HTML summary of Type
11426 | Definition customer-relationship</atom:summary>
11427 |             <atom:title type=""text"">Type Definition - customer-
11428 | relationship</atom:title>
11429 |             <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
11430 |             <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
11431 |             <cmisra:type xmlns:xsi=""http://www.w3.org/2001/XMLSchema-
11432 | instance"" xsi:type=""cmis:cmisTypeRelationshipDefinitionType""
11433 | cmisra:id=""customer-relationship"">
11434 |                 <cmis:id>dtcustomer-relationship</cmis:id>

```



```

11435         <cmis:localName>myrepname-customer-
11436 relationship</cmis:localName>
11437         <cmis:localNamespace xsi:nil="true"/>
11438         <cmis:displayName>customer-
11439 relationship</cmis:displayName>
11440         <cmis:queryName>customer-relationship</cmis:queryName>
11441         <cmis:description>Description for type definition
11442 customer-relationship</cmis:description>
11443         <cmis:baseId>cmis:relationship</cmis:baseId>
11444         <cmis:parentId>parent</cmis:parentId>
11445         <cmis:creatable>true</cmis:creatable>
11446         <cmis:fileable>false</cmis:fileable>
11447         <cmis:queryable>false</cmis:queryable>
11448         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11449
11450 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11451
11452 <cmis:controllablePolicy>true</cmis:controllablePolicy>
11453         <cmis:controllableACL>true</cmis:controllableACL>
11454
11455 <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
11456
11457 <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
11458
11459 <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
11460         </cmisra:type>
11461     </atom:entry>
11462 </atom:feed>
11463 </cmisra:children>
11464 </atom:entry>
11465 <atom:entry>
11466     <atom:author>
11467         <atom:name>Al Brown</atom:name>
11468         <atom:uri>http://www.ibm.com/</atom:uri>
11469         <atom:email>albertcbrown@us.ibm.com</atom:email>
11470     </atom:author>
11471     <atom:content>Type Definition for cmis:policy</atom:content>
11472     <atom:id>http://cmisexample.oasis-
11473 open.org/rep1/type/cmis:policy</atom:id>
11474     <atom:link type="application/atom+xml;type=entry" rel="self"
11475 href="http://cmisexample.oasis-open.org/rep1/type/cmis:policy"/>
11476     <atom:link type="application/atomsvc+xml" rel="service"
11477 href="http://cmisexample.oasis-open.org/rep1/type/cmis:policy"/>
11478     <atom:link type="application/atom+xml;type=entry"
11479 rel="describedby" href="http://cmisexample.oasis-
11480 open.org/rep1/type/cmis:policy"/>
11481     <atom:link type="application/atom+xml;type=entry" rel="up"
11482 href="http://cmisexample.oasis-open.org/rep1/type/cmis:policy/parent"/>
11483     <atom:link type="application/atom+xml;type=feed" rel="down"
11484 href="http://cmisexample.oasis-
11485 open.org/rep1/type/cmis:policy/children/flat"/>
11486     <atom:link type="application/cmistree+xml" rel="down"
11487 href="http://cmisexample.oasis-
11488 open.org/rep1/type/cmis:policy/children/tree"/>
11489     <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
11490     <atom:summary type="html">HTML summary of Type Definition
11491 cmis:policy</atom:summary>
11492     <atom:title type="text">Type Definition - cmis:policy</atom:title>
11493     <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
11494     <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
11495     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
11496 xsi:type="cmis:cmisTypePolicyDefinitionType" cmisra:id="cmis:policy">
11497         <cmis:id>dtcmis:policy</cmis:id>
11498         <cmis:localName>myrepname-cmis:policy</cmis:localName>

```

```

11499 | <cmis:localNamespace xsi:nil="true"/>
11500 | <cmis:displayName>cmis:policy</cmis:displayName>
11501 | <cmis:queryName>cmis:policy</cmis:queryName>
11502 | <cmis:description>Description for type definition
11503 | cmis:policy</cmis:description>
11504 | <cmis:baseId>cmis:policy</cmis:baseId>
11505 | <cmis:parentId>parent</cmis:parentId>
11506 | <cmis:creatable>true</cmis:creatable>
11507 | <cmis:fileable>false</cmis:fileable>
11508 | <cmis:queryable>false</cmis:queryable>
11509 | <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11510 |
11511 | <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11512 | <cmis:controllablePolicy>true</cmis:controllablePolicy>
11513 | <cmis:controllableACL>true</cmis:controllableACL>
11514 | </cmisra:type>
11515 | </atom:entry>
11516 | </atom:feed>

```

11517
11518 Please also see the example documents included with the schema.

11519 **3.9.6.1 GET**

11520 The following arguments may be supplied. Please see the domain model for more information:

- 11521 • includePropertyDefinitions
- 11522 • depth

11523 **3.10 Resources**

11524 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP
11525 verb in a repository-specific manner.

11526 **3.10.1 Type Entry**

11527 This represents a type definition in the repository. This is enclosed as an atom entry

11528

11529 CMIS Services:

11530 GET: getTypeDefinition

11531 Media Type: application/atom+xml;type=entry

11532

11533 Link Relations:

- 11534 • service: Points to service document containing the CMIS repository. The service document
11535 MUST contain only one workspace element.
 - 11536 ○ Media Type: application/atomsvc+xml
- 11537 • up: Points to the parent type as atom entry if applicable
- 11538 • down: Points to the children of this type as atom feed if applicable
 - 11539 ○ (Children) Media Type: application/atom+xml;type=feed points to the atom feed
11540 document representing the children feed for this same type
 - 11541 ○ (Descendants) Media Type: application/cmistree+xml points to the atom feed document
11542 representing the descendants feed for this same type
- 11543 • describedby: Points to the type definition atom entry of the base type of this type definition.

11544

11545 The following CMIS Atom extension element MUST be included inside the atom entry:

11546 • cmisra:type

11547 3.10.1.1 GET

11548 There are no optional arguments for this resource.

11549

11550 Request:

```
11551 GET /obj/5070f89a-6f00-4acf-84e9-d8836a6c7d23 HTTP/1.1
11552 Host: example.org
11553
```

11554

11555 Response:

```
11556 HTTP/1.1 200 Ok
11557 Date: Mon, 25 Jan 2010 10:21:00 -0800
11558 Content-Length: 2995
11559 Content-Type: application/atom+xml;type=entry
11560 Location: http://cmisexample.oasis-open.org/repl/cmis:document
11561
11562
11563 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
11564 <atom:entry xmlns:app="http://www.w3.org/2007/app"
11565 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
11566 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
11567 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
11568 open.org/ns/cmisis/restatom/200908/">
11569   <atom:author>
11570     <atom:name>Al Brown</atom:name>
11571     <atom:uri>http://www.ibm.com/</atom:uri>
11572     <atom:email>albertcbrown@us.ibm.com</atom:email>
11573   </atom:author>
11574   <atom:content>Type Definition for cmis:document</atom:content>
11575   <atom:id>http://cmisexample.oasis-
11576 open.org/repl/type/cmis:document</atom:id>
11577   <atom:link type="application/atom+xml;type=entry" rel="self"
11578 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
11579   <atom:link type="application/atomsvc+xml" rel="service"
11580 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
11581   <atom:link type="application/atom+xml;type=entry" rel="describedby"
11582 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
11583   <atom:link type="application/atom+xml;type=entry" rel="up"
11584 href="http://cmisexample.oasis-open.org/repl/type/cmis:document/parent"/>
11585   <atom:link type="application/atom+xml;type=feed" rel="down"
11586 href="http://cmisexample.oasis-
11587 open.org/repl/type/cmis:document/children/flat"/>
11588   <atom:link type="application/cmistree+xml" rel="down"
11589 href="http://cmisexample.oasis-
11590 open.org/repl/type/cmis:document/children/tree"/>
11591   <atom:published>2010-01-25T10:21:00.380-08:00</atom:published>
11592   <atom:summary type="html">HTML summary of Type Definition
11593 cmis:document</atom:summary>
11594   <atom:title type="text">Type Definition - cmis:document</atom:title>
11595   <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
11596   <app:edited>2010-01-25T10:21:00.380-08:00</app:edited>
11597   <cmisra:type xsi:type="cmis:cmisTypeDocumentDefinitionType"
11598 cmisra:id="cmis:document" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
11599 instance">
11600     <cmis:id>dtcmis:document</cmis:id>
11601     <cmis:localName>myrepname-cmis:document</cmis:localName>
11602     <cmis:localNamespace xsi:nil="true"/>
11603     <cmis:displayName>cmis:document</cmis:displayName>
11604     <cmis:queryName>cmis:document</cmis:queryName>
```

```

11605     <cmis:description>Description for type definition
11606 cmis:document</cmis:description>
11607     <cmis:baseId>cmis:document</cmis:baseId>
11608     <cmis:parentId>parent</cmis:parentId>
11609     <cmis:creatable>true</cmis:creatable>
11610     <cmis:fileable>true</cmis:fileable>
11611     <cmis:queryable>false</cmis:queryable>
11612     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
11613     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
11614     <cmis:controllablePolicy>true</cmis:controllablePolicy>
11615     <cmis:controllableACL>true</cmis:controllableACL>
11616     <cmis:versionable>true</cmis:versionable>
11617     <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
11618   </cmisra:type>
11619 </atom:entry>
11620

```

11621

11622 Please also see the example documents included with the schema.

11623

11624 3.10.2 Document Entry

11625 This is a CMIS Document instance.

11626

11627 CMIS Services:

11628 GET: getObject, getObjectOfLatestVersion (getObject)

11629 PUT: updateProperties

11630 DELETE: deleteObject

11631 Media Type: application/atom+xml;type=entry

11632

11633 Link Relations:

- 11634 • self: Points to an URI that returns the atom entry for this document. Please see Atom for more
- 11635 information.
- 11636 • edit: Points to an URI that accepts PUT of atom entry. Please see AtomPub for more information.
- 11637 • service: Points to service document containing the CMIS repository. The service document
- 11638 MUST contain only one workspace element.
 - 11639 ○ Media Type: application/atomsvc+xml
- 11640 • up: Points to the atom feed containing the set of parents. If there is only one parent, the
- 11641 repository MAY point this link relation directly to the atom entry of the parent.
- 11642 • version-history: Points to atom feed containing the versions of this document
 - 11643 ○ If the document is not versionable, this link relation may not be on the resource
- 11644 • current-version: Points to the latest version of the document
 - 11645 ○ Uses query parameter "`returnVersion`" and `enumReturnVersion`
 - 11646 ○ If this version is the current-version, this link relation may not be on the resource
- 11647 • edit-media:
 - 11648 ○ Same as `setContentStream`. Allows updating the content stream on this document
 - 11649 ○ Please see AtomPub for more information
- 11650 • working-copy: Points to the private working copy if it exists.
- 11651 • describedby: Points to the type definition as an atom entry for the type of this document entry.

- 11652 • alternate: this is used to identify the renditions available for the specified object. Please see the
- 11653 Renditions section.
- 11654 • <http://docs.oasis-open.org/ns/cmismessaging/200908/allowableactions>: Points to the allowable actions
- 11655 document for this object.
- 11656 • <http://docs.oasis-open.org/ns/cmismessaging/200908/relationships>: Points to the relationships feed for
- 11657 this object
- 11658 • <http://docs.oasis-open.org/ns/cmismessaging/200908/policies>: Points to the policy feed for this object.
- 11659 • <http://docs.oasis-open.org/ns/cmismessaging/200908/acl>: Points to ACL document for this object

11660

11661 The following CMIS Atom extension element MUST be included inside the atom entry:

- 11662 • cmisra:object

11663

11664 3.10.2.1 GET

11665 The following arguments may be supplied. Please see the domain model for more information:

- 11666 • returnVersion
 - 11667 ○ Used to differentiate between getObject() and getObjectOfLatestVersion().
 - 11668 ○ valid values are described by the schema element cmisra:enumReturnVersion
 - 11669 ○ If not specified, return the version specified by the URI
- 11670 • includeAllowableActions
- 11671 • includeRelationships
- 11672 • includePolicyIds
- 11673 • includeACL
- 11674 • filter
- 11675 • renditionFilter
 - 11676 ○ If not specified, renditions will not be included.

11677

11678

11679 Request:

```
11680 GET /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId HTTP/1.1
11681 Host: example.org
11682
```

11683

11684 Response:

```
11685 HTTP/1.1 200 Ok
11686 Date: Mon, 25 Jan 2010 10:21:00 -0800
11687 Content-Length: 3403
11688 Content-Type: application/atom+xml;type=entry
11689 Location: /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId
11690
```

11691

```
11692 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
11693 <atom:entry xmlns:app="http://www.w3.org/2007/app"
11694 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
11695 open.org/ns/cmismessaging/200908/" xmlns:cmism="http://docs.oasis-
11696 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
11697 open.org/ns/cmismessaging/200908/">
11698 <atom:author>
11699 <atom:name>Al Brown</atom:name>
```

```

11700     <atom:uri>http://www.ibm.com/</atom:uri>
11701     <atom:email>albertcbrown@us.ibm.com</atom:email>
11702     </atom:author>
11703     <atom:content src="http://cmisexample.oasis-open.org/repl/7c088887-5991-
11704 4b3a-9ad3-16379127e647">
11705     <atom:id>urn:uuid:7c088887-5991-4b3a-9ad3-16379127e647</atom:id>
11706     <atom:title type="text">Invoice</atom:title>
11707     <atom:updated>2010-01-25T10:21:00.193-08:00</atom:updated>
11708     <atom:link rel="self" href="http://cmisexample.oasis-
11709 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647">
11710     <atom:link rel="edit" href="http://cmisexample.oasis-
11711 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647">
11712     <atom:link type="application/cmism+xml;type=allowableActions"
11713 rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
11714 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
11715 16379127e647/allowableactions">
11716     <atom:link type="application/atom+xml;type=entry" rel="describedby"
11717 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
11718 16379127e647/type">
11719     <atom:link type="application/atomsvc+xml" rel="service"
11720 href="http://cmisexample.oasis-open.org/repl//service">
11721     <atom:published>2010-01-25T10:21:00.193-08:00</atom:published>
11722     <atom:summary type="html">HTML summary of Entry 7c088887-5991-4b3a-
11723 9ad3-16379127e647</atom:summary>
11724     <atom:link rel="edit-media" href="http://cmisexample.oasis-
11725 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/edit-media">
11726     <atom:link rel="alternate" href="http://cmisexample.oasis-
11727 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/alternate">
11728     <atom:link type="application/atom+xml;type=feed" rel="up"
11729 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
11730 16379127e647/parents">
11731     <atom:link type="application/atom+xml;type=feed" rel="version-history"
11732 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
11733 16379127e647/allversions">
11734     <atom:link type="application/atom+xml;type=entry" rel="current-
11735 version" href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-
11736 9ad3-16379127e647/latest">
11737     <atom:link type="application/atom+xml;type=feed"
11738 rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
11739 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
11740 16379127e647/relationships">
11741     <atom:link type="application/atom+xml;type=feed"
11742 rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
11743 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
11744 16379127e647/policies">
11745     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
11746 open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
11747 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/acl">
11748     <cmisra:object>
11749     <cmis:properties>
11750     <cmis:propertyId localName="rep-cmis:objectId"
11751 propertyDefinitionId="cmis:objectId">
11752     <cmis:value>7c088887-5991-4b3a-9ad3-16379127e647</cmis:value>
11753     </cmis:propertyId>
11754     </cmis:properties>
11755     </cmisra:object>
11756 </atom:entry>
11757

```

11758

11759 Please also see the example documents included with the schema.

11760

11761 **3.10.2.2 PUT**

11762 This does a replacement of the atom entry with the atom entry document specified. If readwrite
11763 properties are not included, the repository SHOULD NOT modify them.

11764

11765 The server SHOULD respond with:

- 11766 • HTTP Status Code 200
- 11767 • Response Body containing the updated atom entry

11768

11769 **3.10.2.3 DELETE**

11770 This removes the document.

11771 Success HTTP code: 204

11772 **3.10.3 Document Private Working Copy (PWC) Entry**

11773 This is the private working copy of the document (checkedout version of document)

11774 CMIS Services:

11775 GET: getObject

11776 PUT: updateProperties or checkIn

11777 DELETE: cancelCheckOut

11778 Media Type: application/atom+xml;type=entry

11779

11780 Link relations:

- 11781 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 11782 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more
11783 information.
- 11784 • service: Points to service document containing the CMIS repository. The service document
11785 MUST contain only one workspace element.
 - 11786 ○ Media Type: application/atomsvc+xml
- 11787 • up: Points to the atom feed containing the set of parents. If there is only one parent, the
11788 repository MAY point this link relation directly to the atom entry of the parent.
- 11789 • version-history
 - 11790 ○ Points to an URI that returns the feed associated with the version history
- 11791 • edit-media
 - 11792 ○ Same as setContentStream. Allows updating the content stream on this document
 - 11793 ○ Please see AtomPub for more information
- 11794 • via: atom entry that created this private working copy
- 11795 • describedby: Points to the type definition as an atom entry for the type of this PWC entry.
- 11796 • alternate: this is used to identify the renditions available for the specified object. Please see the
11797 Renditions section.
- 11798 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
11799 document for this object.
- 11800 • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for
11801 this object
- 11802 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.

- 11803 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

11804

11805 The following element MUST be included inside the atom entry:

- 11806 • cmisra:object

11807

11808 3.10.3.1 GET

11809 The following arguments may be supplied. Please see the domain model for more information:

- 11810 • filter
- 11811 • includeAllowableActions
- 11812 • includeRelationships
- 11813 • renditionFilter
- 11814 ○ If not specified, renditions will not be included.

11815

11816 Request:

```
11817       GET /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId HTTP/1.1
11818       Host: example.org
11819
```

11820

11821 Response:

```
11822       HTTP/1.1 200 Ok
11823       Date: Mon, 25 Jan 2010 10:21:00 -0800
11824       Content-Length: 3564
11825       Content-Type: application/atom+xml;type=entry
11826       Location: /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId
11827
11828       <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
11829       <atom:entry xmlns:app="http://www.w3.org/2007/app"
11830       xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
11831       open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
11832       open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
11833       open.org/ns/cmisis/restatom/200908/">
11834        <atom:author>
11835         <atom:name>Al Brown</atom:name>
11836         <atom:uri>http://www.ibm.com/</atom:uri>
11837         <atom:email>albertcbrown@us.ibm.com</atom:email>
11838       </atom:author>
11839       <atom:content src="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-
11840       4ab2-978d-85ca2f4f3206"/>
11841       <atom:id>urn:uuid:3240a476-6de6-4ab2-978d-85ca2f4f3206</atom:id>
11842       <atom:title type="text">Invoice</atom:title>
11843       <atom:updated>2010-01-25T10:21:00.333-08:00</atom:updated>
11844       <atom:link rel="self" href="http://cmisexample.oasis-
11845       open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>
11846       <atom:link rel="edit" href="http://cmisexample.oasis-
11847       open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>
11848       <atom:link type="application/cmisis+xml;type=allowableActions"
11849       rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
11850       href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11851       85ca2f4f3206/allowableactions"/>
11852       <atom:link type="application/atom+xml;type=entry" rel="describedby"
11853       href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11854       85ca2f4f3206/type"/>
11855
```



```

11856 | <atom:link type="application/atomsvc+xml" rel="service"
11857 | href="http://cmisexample.oasis-open.org/rep1//service"/>
11858 | <atom:published>2010-01-25T10:21:00.333-08:00</atom:published>
11859 | <atom:summary type="html">HTML summary of Entry 3240a476-6de6-4ab2-
11860 | 978d-85ca2f4f3206</atom:summary>
11861 | <atom:link rel="edit-media" href="http://cmisexample.oasis-
11862 | open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/edit-media"/>
11863 | <atom:link rel="alternate" href="http://cmisexample.oasis-
11864 | open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/alternate"/>
11865 | <atom:link type="application/atom+xml;type=feed" rel="up"
11866 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11867 | 85ca2f4f3206/parents"/>
11868 | <atom:link type="application/atom+xml;type=feed" rel="version-history"
11869 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11870 | 85ca2f4f3206/allversions"/>
11871 | <atom:link type="application/atom+xml;type=entry" rel="current-
11872 | version" href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-
11873 | 978d-85ca2f4f3206/latest"/>
11874 | <atom:link type="application/atom+xml;type=feed"
11875 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
11876 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11877 | 85ca2f4f3206/relationships"/>
11878 | <atom:link type="application/atom+xml;type=feed"
11879 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
11880 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11881 | 85ca2f4f3206/policies"/>
11882 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
11883 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
11884 | open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/acl"/>
11885 | <atom:link type="application/atom+xml;type=feed" rel="working-copy"
11886 | href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
11887 | 85ca2f4f3206/pwc"/>
11888 | <cmisra:object>
11889 |   <cmis:properties>
11890 |     <cmis:propertyId localName="rep-cmis:objectId"
11891 |     propertyDefinitionId="cmis:objectId">
11892 |       <cmis:value>3240a476-6de6-4ab2-978d-85ca2f4f3206</cmis:value>
11893 |     </cmis:propertyId>
11894 |   </cmis:properties>
11895 | </cmisra:object>
11896 | </atom:entry>
11897 |

```

11898

11899 Please also see the example documents included with the schema.

11900

11901 3.10.3.2 PUT

11902 This does a replacement of the atom entry with the atom entry document specified. If modifiable
 11903 properties (whencheckedout or readwrite) are not included, the repository SHOULD NOT modify them.

11904

11905 The following arguments may be supplied. Please see the domain model for more information:

- 11906 • checkinComment
- 11907 • major
- 11908 • checkin
 - 11909 ○ Used to differentiate between updateProperties() or checkin() services. If TRUE, execute
 - 11910 checkin service.

- 11911 ○ If this argument is specified as TRUE, then the body to PUT MAY be omitted if there are
11912 no modifications to be made during checkin

11913

11914 The server SHOULD respond with:

- 11915 • HTTP Status Code 200
- 11916 • Location header of the resource (if changed via checkin)
- 11917 • Response Body containing the updated atom entry

11918 **3.10.3.3 DELETE**

11919 This removes the document entry, in this case, cancels the check out. The PWC will be removed.

11920

11921 Success HTTP code: 204

11922 **3.10.4 Folder Entry**

11923 This is a CMIS Folder instance. The properties of a folder map onto the feed tag.

11924 CMIS Services:

- 11925 GET: getObject
- 11926 PUT: updateProperties
- 11927 DELETE: deleteObject (this is deletion of the folder only and not any contained objects)

11928 Media Type: application/atom+xml;type=entry

11929

11930 Link Relations:

- 11931 • self: Points to the URI to retrieve this atom entry. Please see Atom for more informationedit:
11932 Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
- 11933 • service: Points to service document containing the CMIS repository. The service document
11934 MUST contain only one workspace element.
 - 11935 ○ Media Type: application/atomsvc+xml
- 11936 • describedby: Points to the type definition as an atom entry for the type of this folder entry.
- 11937 • down: Points to the children of this folder
 - 11938 ○ application/atom+xml : Points to the atom feed document representing the children feed
11939 for this same folder
 - 11940 ○ application/cmistree+xml: Points to the descendants feed of the same folder
- 11941 • up: Points to the atom entry for the parent
 - 11942 ○ If the root folder, this link will not be present
- 11943 • alternate: this is used to identify the renditions available for the specified object. Please see the
11944 Renditions section.
- 11945 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
11946 document for this object.
- 11947 • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for
11948 this object
- 11949 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 11950 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object
- 11951 • <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>: Points to the folder tree for this folder

11952

11953 The following CMIS Atom extension element MUST be included inside the atom entry:

- 11954 • cmisra:object

11955

11956 3.10.4.1 GET

11957 The following arguments may be supplied. Please see the domain model for more information:

- 11958 • filter
- 11959 • includeAllowableActions
- 11960 • includeRelationships
- 11961 • renditionFilter
 - 11962 ○ If not specified, renditions will not be included.

11963

11964 Request:

```
11965 GET /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId HTTP/1.1
11966 Host: example.org
11967
```

11968

11969 Response:

```
11970 HTTP/1.1 200 Ok
11971 Date: Mon, 25 Jan 2010 10:21:00 -0800
11972 Content-Length: 3332
11973 Content-Type: application/atom+xml;type=entry
11974 Location: /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId
11975
11976
11977 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
11978 <atom:entry xmlns:app="http://www.w3.org/2007/app"
11979 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
11980 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
11981 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
11982 open.org/ns/cmisis/restatom/200908/">
11983   <atom:author>
11984     <atom:name>Al Brown</atom:name>
11985     <atom:uri>http://www.ibm.com/</atom:uri>
11986     <atom:email>albertcbrown@us.ibm.com</atom:email>
11987   </atom:author>
11988   <atom:content src="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-
11989 471d-b4ba-6d8756cd5093"/>
11990   <atom:id>urn:uuid:cfc03a28-8240-471d-b4ba-6d8756cd5093</atom:id>
11991   <atom:title type="text">Customer Folder</atom:title>
11992   <atom:updated>2010-01-25T10:21:00.208-08:00</atom:updated>
11993   <atom:link rel="self" href="http://cmisexample.oasis-
11994 open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
11995   <atom:link rel="edit" href="http://cmisexample.oasis-
11996 open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
11997   <atom:link type="application/cmisis+xml;type=allowableActions"
11998 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
11999 href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12000 6d8756cd5093/allowableactions"/>
12001   <atom:link type="application/atom+xml;type=entry" rel="describedby"
12002 href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12003 6d8756cd5093/type"/>
12004   <atom:link type="application/atomsvc+xml" rel="service"
12005 href="http://cmisexample.oasis-open.org/rep1//service"/>
12006   <atom:published>2010-01-25T10:21:00.208-08:00</atom:published>
```

```

12007 | <atom:summary type="html">HTML summary of Entry cfc03a28-8240-471d-
12008 | b4ba-6d8756cd5093</atom:summary>
12009 | <atom:link type="application/atom+xml;type=entry" rel="up"
12010 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12011 | 6d8756cd5093/up"/>
12012 | <atom:link type="application/atom+xml;type=feed" rel="down"
12013 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12014 | 6d8756cd5093/children"/>
12015 | <atom:link type="application/cmistree+xml" rel="down"
12016 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12017 | 6d8756cd5093/tree"/>
12018 | <atom:link type="application/atom+xml;type=feed"
12019 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
12020 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12021 | 6d8756cd5093/foldertree"/>
12022 | <atom:link type="application/atom+xml;type=feed"
12023 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
12024 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12025 | 6d8756cd5093/relationships"/>
12026 | <atom:link type="application/atom+xml;type=feed"
12027 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
12028 | href="http://cmisexample.oasis-open.org/rep1/cfc03a28-8240-471d-b4ba-
12029 | 6d8756cd5093/policies"/>
12030 | <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
12031 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
12032 | open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093/acl"/>
12033 | <cmisra:object>
12034 |   <cmis:properties>
12035 |     <cmis:propertyId localName="rep-cmis:objectId"
12036 |     propertyDefinitionId="cmis:objectId">
12037 |       <cmis:value>cfc03a28-8240-471d-b4ba-6d8756cd5093</cmis:value>
12038 |     </cmis:propertyId>
12039 |   </cmis:properties>
12040 | </cmisra:object>
12041 | </atom:entry>
12042 |

```

12043

12044 Please also see the example documents included with the schema.

12045

12046 3.10.4.2 PUT

12047 This does a replacement of the atom entry with the atom entry document specified. If readwrite
12048 properties are not included, the repository SHOULD NOT modify them.

12049

12050 The server SHOULD respond with:

- 12051 • HTTP Status Code 200
- 12052 • Response Body containing the updated atom entry

12053

12054 3.10.4.3 DELETE

12055 This removes the object (folder) from the repository.

12056 Success HTTP code: 204

12057 3.10.5 Relationship Entry

12058 This is a CMIS relationship instance. These objects are exposed via "relationships" link type.

12059 CMIS Services:
12060 GET: getObject
12061 PUT: updateProperties
12062 DELETE: deleteObject
12063 Media Type: application/atom+xml;type=entry
12064

12065 Link Relations:

- 12066 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 12067 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more
12068 information.
- 12069 • service: Points to service document containing the CMIS repository. The service document
12070 MUST contain only one workspace element.
 - 12071 ○ Media Type: application/atomsvc+xml
- 12072 • describedby: Points to the type definition as an atom entry for the type of this relationship entry.
- 12073 • <http://docs.oasis-open.org/ns/cmisis/link/200908/target>
- 12074 • <http://docs.oasis-open.org/ns/cmisis/link/200908/source>
- 12075 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
12076 document for this object.
- 12077 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 12078 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

12079
12080 The following element MUST be included inside the atom entry:

- 12081 • cmisra:object
- 12082

12083 3.10.5.1 GET

12084 The following arguments may be supplied. Please see the domain model for more information:

- 12085 • filter
- 12086 • includeAllowableActions

12087
12088 Request:

```
12089 GET /obj/ad443afd-aa1a-4071-9735-1a49aac4e439?filter=cmis:objectId HTTP/1.1  
12090 Host: example.org  
12091
```

12092
12093 Response:

```
12094 HTTP/1.1 200 Ok  
12095 Date: Mon, 25 Jan 2010 10:21:00 -0800  
12096 Content-Length: 2861  
12097 Content-Type: application/atom+xml;type=entry  
12098 Location: /obj/ad443afd-aa1a-4071-9735-1a49aac4e439?filter=cmis:objectId  
12099  
12100  
12101 | <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

12102 | <atom:entry xmlns:app="http://www.w3.org/2007/app"
12103 | xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
12104 | open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
12105 | open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
12106 | open.org/ns/cmisis/restatom/200908/">
12107 |   <atom:author>
12108 |     <atom:name>Al Brown</atom:name>
12109 |     <atom:uri>http://www.ibm.com/</atom:uri>
12110 |     <atom:email>albertcbrown@us.ibm.com</atom:email>
12111 |   </atom:author>
12112 |   <atom:content src="http://cmisexample.oasis-open.org/repl/ad443afd-aala-
12113 | 4071-9735-1a49aac4e439"/>
12114 |   <atom:id>urn:uuid:ad443afd-aala-4071-9735-1a49aac4e439</atom:id>
12115 |   <atom:title type="text">Customer Relationship</atom:title>
12116 |   <atom:updated>2010-01-25T10:21:00.349-08:00</atom:updated>
12117 |   <atom:link rel="self" href="http://cmisexample.oasis-
12118 | open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439"/>
12119 |   <atom:link rel="edit" href="http://cmisexample.oasis-
12120 | open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439"/>
12121 |   <atom:link type="application/cmisis+xml;type=allowableActions"
12122 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
12123 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
12124 | 1a49aac4e439/allowableactions"/>
12125 |   <atom:link type="application/atom+xml;type=entry" rel="describedby"
12126 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
12127 | 1a49aac4e439/type"/>
12128 |   <atom:link type="application/atomsvc+xml" rel="service"
12129 | href="http://cmisexample.oasis-open.org/repl//service"/>
12130 |   <atom:published>2010-01-25T10:21:00.365-08:00</atom:published>
12131 |   <atom:summary type="html">HTML summary of Entry ad443afd-aala-4071-
12132 | 9735-1a49aac4e439</atom:summary>
12133 |   <atom:link type="application/atom+xml;type=entry"
12134 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/source"
12135 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
12136 | 1a49aac4e439/source"/>
12137 |   <atom:link type="application/atom+xml;type=entry"
12138 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/target"
12139 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
12140 | 1a49aac4e439/target"/>
12141 |   <atom:link type="application/atom+xml;type=feed"
12142 | rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
12143 | href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
12144 | 1a49aac4e439/policies"/>
12145 |   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
12146 | open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
12147 | open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/acl"/>
12148 |   <cmisra:object>
12149 |     <cmis:properties>
12150 |       <cmis:propertyId localName="rep-cmisis:objectId"
12151 |       propertyDefinitionId="cmisis:objectId">
12152 |         <cmis:value>ad443afd-aala-4071-9735-1a49aac4e439</cmis:value>
12153 |       </cmis:propertyId>
12154 |     </cmis:properties>
12155 |   </cmisra:object>
12156 | </atom:entry>
12157 |

```

12158

12159 Please also see the example documents included with the schema.

12160

12161 **3.10.5.2 PUT**

12162 This does a replacement of the atom entry with the atom entry document specified. If readwrite
12163 properties are not included, the repository SHOULD NOT modify them.

12164

12165 The server SHOULD respond with:

- 12166 • HTTP Status Code 200
- 12167 • Response Body containing the updated atom entry

12168

12169 **3.10.5.3 DELETE**

12170 This removes the relationship entry.

12171 Successful HTTP code: 204

12172 **3.10.6 Policy Entry**

12173 This is a CMIS policy instance.

12174 CMIS Services:

12175 GET: getObject

12176 PUT: updateProperties

12177 DELETE: deleteObject or removePolicy

12178 Media Type: application/atom+xml;type=entry

12179

12180 Link Relations:

- 12181 • self
- 12182 • edit
- 12183 • service: Points to service document containing the CMIS repository. The service document
12184 MUST contain only one workspace element.
 - 12185 ○ Media Type: application/atomsvc+xml
- 12186 • describedby: Points to the type definition as an atom entry for the type of this policy entry.
- 12187 • alternate: this is used to identify the renditions available for the specified object. Please see the
12188 Renditions section.
- 12189 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions
12190 document for this object.
- 12191 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 12192 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

12193

12194 The following element MUST be included inside the atom entry:

- 12195 • `cmisra:object`

12196

12197 **3.10.6.1 GET**

12198 The following arguments may be supplied. Please see the domain model for more information:

- 12199 • filter
- 12200 • includeAllowableActions

- 12201 • includeRelationships
- 12202 • renditionFilter
 - 12203 ○ If not specified, renditions will not be included.

12204

12205 Request:

```
12206 GET /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId HTTP/1.1
12207 Host: example.org
12208
```

12209

12210 Response:

```
12211 HTTP/1.1 200 Ok
12212 Date: Mon, 25 Jan 2010 10:21:00 -0800
12213 Content-Length: 2608
12214 Content-Type: application/atom+xml;type=entry
12215 Location: /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId
12216
12217
12218 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
12219 <atom:entry xmlns:app="http://www.w3.org/2007/app"
12220 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
12221 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
12222 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
12223 open.org/ns/cmisis/restatom/200908/">
12224   <atom:author>
12225     <atom:name>Al Brown</atom:name>
12226     <atom:uri>http://www.ibm.com/</atom:uri>
12227     <atom:email>albertcbrown@us.ibm.com</atom:email>
12228   </atom:author>
12229   <atom:content src="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-
12230 4940-b2f0-4e4cd4631bf0"/>
12231   <atom:id>urn:uuid:a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</atom:id>
12232   <atom:title type="text">Security Policy</atom:title>
12233   <atom:updated>2010-01-25T10:21:00.318-08:00</atom:updated>
12234   <atom:link rel="self" href="http://cmisexample.oasis-
12235 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
12236   <atom:link rel="edit" href="http://cmisexample.oasis-
12237 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
12238   <atom:link type="application/cmisis+xml;type=allowableActions"
12239 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
12240 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
12241 4e4cd4631bf0/allowableactions"/>
12242   <atom:link type="application/atom+xml;type=entry" rel="describedby"
12243 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
12244 4e4cd4631bf0/type"/>
12245   <atom:link type="application/atomsvc+xml" rel="service"
12246 href="http://cmisexample.oasis-open.org/rep1//service"/>
12247   <atom:published>2010-01-25T10:21:00.318-08:00</atom:published>
12248   <atom:summary type="html">HTML summary of Entry a09ed524-5f1b-4940-
12249 b2f0-4e4cd4631bf0</atom:summary>
12250   <atom:link type="application/atom+xml;type=feed" rel="up"
12251 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
12252 4e4cd4631bf0/parents"/>
12253   <atom:link type="application/atom+xml;type=feed"
12254 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
12255 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
12256 4e4cd4631bf0/relationships"/>
12257   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
12258 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
12259 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0/acl"/>
12260   <cmisra:object>
```



```
12261     <cmis:properties>
12262         <cmis:propertyId localName=""_rep-cmis:objectId""
12263         propertyDefinitionId=""_cmis:objectId"">
12264             <cmis:value>a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</cmis:value>
12265             </cmis:propertyId>
12266         </cmis:properties>
12267     </cmisra:object>
12268 </atom:entry>
12269
```

12270

12271 Please also see the example documents included with the schema.

12272

12273 **3.10.6.2 PUT**

12274 This does a replacement of the atom entry with the atom entry document specified. If read/write
12275 properties are not included, the repository SHOULD NOT modify them.

12276

12277 The server SHOULD respond with:

- 12278 • HTTP Status Code 200
- 12279 • Response Body containing the updated atom entry

12280

12281 **3.10.6.3 DELETE**

12282 This removes the policy entry. If this policy entry was discovered through a policy collection on an object,
12283 then removePolicy() is performed rather than deleteObject() on the policy itself.

12284

12285 Success HTTP code: 204

12286 **3.10.7 Content Stream**

12287 This is the content stream portion of the document object.

12288 CMIS Services:

12289 GET: getContentStream

12290 PUT: setContentStream

12291 DELETE: deleteContentStream

12292 Media Type: Mime/Type of resource (mime type of content stream on document)

12293 **3.10.7.1 GET**

12294 This returns the content stream.

12295

12296 It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED
12297 that HTTP compression is also supported.

12298

12299 Please see RFC2616 for more information on HTTP Range requests.

12300 **3.10.7.2 PUT**

12301 This does a replacement of the content stream.

12302

12303 The following optional arguments may be supplied. Please see the domain model for more information:

- 12304 • overwriteFlag.
 - 12305 ○ If not specified, this defaults to `true` in this binding and behaves consistent with
 - 12306 AtomPub.

12307

12308 Success HTTP code: 200 (with content), 204 (without content) or 201 if a new resource is created.
12309 Please see the HTTP specification for more information.

12310

12311 Returns headers:

- 12312 • Content-Location: URI for content stream
- 12313 • Location: URI for content stream

12314 3.10.7.3 DELETE

12315 This removes the content stream.

12316 3.10.8 ACL Resource

12317 CMIS Services:

- 12318 GET: getACL
- 12319 PUT: applyACL

12320

12321 Media Type: application/cmisacl+xml

12322

12323 3.10.8.1 GET

12324 This returns the CMIS ACL for a specified object. The client will follow the link on the atom entry to get
12325 the CMIS ACL for that object.

12326

12327 Request:

```
12328 GET /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef HTTP/1.1  
12329 Host: example.org  
12330
```

12331

12332 Response:

```
12333 HTTP/1.1 200 Ok  
12334 Date: Mon, 25 Jan 2010 10:21:00 -0800  
12335 Content-Length: 758  
12336 Content-Type: application/cmisacl+xml  
12337 Location: /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef  
12338  
12339  
12340 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
12341 <cmis:acl xmlns:app="http://www.w3.org/2007/app"  
12342 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
12343 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-  
12344 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-  
12345 open.org/ns/cmisis/restatom/200908/">  
12346 <cmis:permission>  
12347 <cmis:principal>  
12348 <cmis:principalId>Al Brown</cmis:principalId>  
12349 </cmis:principal>
```

```
12350 <cmis:permission>cmis:read</cmis:permission>
12351 <cmis:permission>cmis:write</cmis:permission>
12352 <cmis:permission>cmis:all</cmis:permission>
12353 <cmis:permission>publish</cmis:permission>
12354 <cmis:direct>true</cmis:direct>
12355 </cmis:permission>
12356 </cmis:acl>
12357
```

12358

12359 Please also see the example documents included with the schema.

12360 **4 Web Services Binding**

12361 **4.1 Overview**

12362 All services and operations defined in the Domain Model are presented in the Web Services binding.

12363 The WSDL for these services reference two XSD documents. One defines elements for the primary data
12364 types of documents, folders, relationships and policies as well as collections of these types of objects.

12365 The second XSD defines the message formats for each of the CMIS services; the messages often refer
12366 to the data types defined in the first XSD schema. The WSDL presents exactly the abstract services
12367 defined in the Services section.

12368 The normative CMIS Web Services binding is defined by the WSDL and XSD as well as the details given
12369 here in this part of the CMIS specification except the examples.

12370 **4.1.1 WS-I**

12371 A CMIS Web Services binding **MUST** comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

12372 **4.1.2 Authentication**

12373 A CMIS Web Services binding **SHOULD** support WS-Security 1.1 for Username Token Profile 1.1 and
12374 **MAY** also support other authentication mechanisms. A CMIS repository **MAY** grant access to all or a
12375 subset of the CMIS services to unauthenticated clients.

12376 **4.1.3 Content Transfer**

12377 All endpoints of the Web Services binding **MUST** be MTOM enabled.

12378 **4.1.4 Reporting Errors**

12379 Services **MUST** report errors via SOAP faults. The CMIS-Messaging.xsd defines a basic fault structure
12380 that includes an error code and an error message and the WSDL for each service defines specific
12381 messages that have the basic fault format.

12382 **4.2 Web Services Binding Mapping**

12383 The Domain Model defines all services, operations, parameters and objects of CMIS. The Web Services
12384 binding is an exact one-to-one mapping of this definition with small exceptions that are explained in the
12385 next section. Operations and parameters are named exactly after their counterparts in the Services
12386 section. All rules and exceptions defined there apply to the Web Services binding. Optional parameters
12387 and optional return values are not set if they are missing or their value is NULL.

12388 **4.3 Additions to the Services section**

12389 **4.3.1 updateProperties and checkIn Semantics**

12390 This binding supports partial properties updates. All properties passed to updateProperties or checkIn will
12391 be updated to their new values. Properties that are passed without a value will be set to their default
12392 value or un-set if no default value is defined. All others property values remain untouched.

12393 **4.3.2 Content Ranges**

12394 This binding supports the retrieval of content ranges. The operation getContentStream accepts two
12395 optional parameters:

- 12432 • **Integer offset:** The first byte of the content to retrieve. Default value is 0.
- 12433 • **Integer length:** The length of the range in bytes. Default value is the size of the content minus
- 12434 the offset.

12435

12436 If the **offset** value is greater than the size of the content the repository SHOULD throw a `constraint`

12437 exception.

12438 If **offset + length** is greater than the size of the content the repository should deliver the content from the

12439 offset to the end of the content.

12440

12441 4.3.3 Extensions

12442 On all input messages and some output messages exists an element called extension. This element is

12443 used to provide vendor or repository-specific information between client and server.

12444 All of the types referenced by the schema also support `xs:any` for vendor or repository-specific

12445 information.

12446 4.3.4 Web Services Specific Structures

12447 This binding requires specific structures that are not part of the general CMIS schema.

12448 Please also see the example request and response documents included with the schema.

12449 4.3.4.1 `cmisFaultType` and `cmisFault`

12450 `cmisFaultType` and `cmisFault` SHOULD be used to generate SOAP faults. See 0

12451 [Reporting Errors](#)

12452 [Reporting Errors](#).

12453 4.3.4.2 `cmisRepositoryEntryType`

12454 `cmisRepositoryEntryType` is the return structure of `getRepositoryEntries`. It contains the id and the name

12455 of a repository.

12456 4.3.4.3 `cmisTypeContainer`

12457 `cmisTypeContainer` is the return structure of `getTypeDescendants`. It holds a type hierarchy.

12458 4.3.4.4 `cmisTypeDefinitionListType`

12459 `cmisTypeDefinitionListType` is the return structure of `getTypeChildren`. It contains a list of types,

12460 the `hasMoreItems` flag and the `numItem` element.

12461 4.3.4.5 `cmisObjectInFolderType`, `cmisObjectParentsType` and

12462 `cmisObjectInFolderContainerType`

12463 `cmisObjectInFolderType` holds, in addition to a `cmisObjectType` object, a path segment string. It

12464 is used in all operations that support the `includePathSegments` parameter.

12465 `cmisObjectParentsType` is similar but has a relative path segment string instead of a path segment.

12466 For details about path segments and relative path segments see section 2.1.5.3 Paths.

12467 `cmisObjectInFolderContainerType` contains a folder hierarchy.

12468 **4.3.4.6 cmisObjectListType and cmisObjectInFolderListType**

12469 `cmisObjectListType` and `cmisObjectInFolderListType` hold lists of `cmisObjectType` and
12470 `cmisObjectInFolderType` structures. They also contain the `hasMoreItems` flag and the `numItems`
12471 element that are returned by operations that return these lists.

12472 **4.3.4.7 cmisContentStreamType**

12473 `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 <code>cmis:contentStreamMimeType</code> .	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 <code>cmis:contentStreamFileName</code> .	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

12474

12475 **4.3.4.8 cmisACLType**

12476 `cmisACLType` is the return structure of `getACL` and `applyACL`. It contains the current Access Control List
12477 (ACL) of the object and the `exact` flag that indicates if the ACL fully describes the permission of this
12478 object.

12479 **4.3.4.9 cmisExtensionType**

12480 `cmisExtensionType` is a placeholder for extensions. See 4.3.3 Extensions.

12481

12482 5 IANA Considerations

12483 5.1 Content-Type Registration

12484 5.1.1 CMIS Query

12485 A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

12486

12487 MIME media type name: application

12488 MIME subtype name: cmisquery+xml

12489 Mandatory parameters: None

12490 Optional parameters:

12491 | `charset`: This parameter has semantics identical to the `charset` parameter of the
12492 | `application/xml` media type as specified in [RFC3023].

12493 Encoding considerations:

12494 | Identical to those of `application/xml` as described in [RFC3023], Section 3.2.

12495 Security considerations: As defined in this specification.

12496 | In addition, as this media type uses the `+xml` convention, it shares the same security
12497 | considerations as described in [RFC3023], Section 10.

12498 Interoperability considerations:

12499 There are no known interoperability issues.

12500 Published specification: This specification.

12501 Applications that use this media type:

12502 No known applications currently use this media type.

12503 Additional information:

12504 Magic number(s):

12505 | As specified for `application/xml` in [RFC3023], Section 3.2.

12506 File extension: `.cmisquery`

12507 Fragment identifiers:

12508 | As specified for `application/xml` in [RFC3023], Section 5.

12509 Base URI:

12510 As specified in [RFC3023], Section 6.

12511 Macintosh File Type code: TEXT

12512 Person and email address to contact for further information:

12513 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)

12514 Intended usage: COMMON

12515 Author/Change controller: IESG

12516 5.1.2 CMIS AllowableActions

12517 A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following
12518 media type:

12519

12520 MIME media type name: application
12521 MIME subtype name: cmisallowableactions+xml
12522 Mandatory parameters: None.
12523 Optional parameters:
12524 | `charset`: This parameter has semantics identical to the charset parameter of the
12525 | `application/xml` media type as specified in [RFC3023].
12526 Encoding considerations:
12527 | Identical to those of `application/xml` as described in [RFC3023], Section 3.2.
12528 Security considerations: As defined in this specification.
12529 | In addition, as this media type uses the `+xml` convention, it shares the same security
12530 | considerations as described in [RFC3023], Section 10.
12531 Interoperability considerations:
12532 | There are no known interoperability issues.
12533 Published specification: This specification.
12534 Applications that use this media type:
12535 | No known applications currently use this media type.
12536 Additional information:
12537 Magic number(s):
12538 | As specified for `application/xml` in [RFC3023], Section 3.2.
12539 File extension: .cmisallowableactions
12540 Fragment identifiers:
12541 | As specified for `application/xml` in [RFC3023], Section 5.
12542 Base URI:
12543 | As specified in [RFC3023], Section 6.
12544 Macintosh File Type code: TEXT
12545 Person and email address to contact for further information:
12546 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
12547 Intended usage: COMMON
12548 Author/Change controller: IESG
12549

12550 **5.1.3 CMIS Tree**

12551 A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:

12552
12553 MIME media type name: application
12554 MIME subtype name: cmistree+xml
12555 Mandatory parameters: None.
12556 Optional parameters:
12557 | `charset`: This parameter has semantics identical to the charset parameter of the `application/xml`
12558 | media type as specified in [RFC3023].
12559 Encoding considerations:
12560 | Identical to those of `application/xml` as described in [RFC3023], Section 3.2.
12561 Security considerations: As defined in this specification.

12562 | In addition, as this media type uses the `application/xml+cmis` convention, it shares the same security considerations as
12563 | described in [RFC3023], Section 10.

12564 | Interoperability considerations:
12565 | There are no known interoperability issues.

12566 | Published specification: This specification.

12567 | Applications that use this media type:
12568 | No known applications currently use this media type.

12569 | Additional information:
12570 | Magic number(s):
12571 | As specified for `application/xml+cmis` in [RFC3023], Section 3.2.

12572 | File extension: `.cmistree`

12573 | Fragment identifiers:
12574 | As specified for `application/xml+cmis` in [RFC3023], Section 5.

12575 | Base URI:
12576 | As specified in [RFC3023], Section 6.

12577 | Macintosh File Type code: `TEXT`

12578 | Person and email address to contact for further information:
12579 | [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)

12580 | Intended usage: `COMMON`

12581 | Author/Change controller: `IESG`

12582

12583 **5.1.4 CMIS Atom**

12584 | A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:
12585 |
12586 | MIME media type name: `application`
12587 | MIME subtype name: `cmisatom+xml`
12588 | Mandatory parameters: `None`.
12589 | Optional parameters:
12590 | `charset`: This parameter has semantics identical to the `charset` parameter of the `application/xml+cmis`
12591 | media type as specified in [RFC3023].

12592 | `type`: This parameter has semantics identical to the `type` parameter of the `application/atom+xml` as
12593 | specified in [RFC4287]

12594 | Encoding considerations:
12595 | Identical to those of `application/xml+cmis` as described in [RFC3023], Section 3.2.

12596 | Security considerations: As defined in this specification.

12597 | In addition, as this media type uses the `application/xml+cmis` convention, it shares the same security considerations as
12598 | described in [RFC3023], Section 10.

12599 | Interoperability considerations:
12600 | There are no known interoperability issues.

12601 | Published specification: This specification.

12602 | Applications that use this media type:
12603 | No known applications currently use this media type.

12604 | Additional information:

12605 Magic number(s):
12606 | As specified for `application/xml` in [RFC3023], Section 3.2.
12607 File extension: `.cmisatom`
12608 Fragment identifiers:
12609 | As specified for `application/xml` in [RFC3023], Section 5.
12610 Base URI:
12611 As specified in [RFC3023], Section 6.
12612 Macintosh File Type code: `TEXT`
12613 Person and email address to contact for further information:
12614 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
12615 Intended usage: `COMMON`
12616 Author/Change controller: `IESG`
12617
12618 Please see section 3.1.1 on why this media type is needed above the Atom Media Type.

12619 **5.1.5 CMIS ACL**

12620 A CMIS ACL Document, when serialized as XML 1.0, can be identified with the following media type:

12621
12622 MIME media type name: `application`
12623 MIME subtype name: `cmisacl+xml`
12624 Mandatory parameters: `None`.
12625 Optional parameters:
12626 | `charset`: This parameter has semantics identical to the `charset` parameter of the `application/xml`
12627 media type as specified in [RFC3023].
12628 Encoding considerations:
12629 | Identical to those of `application/xml` as described in [RFC3023], Section 3.2.
12630 Security considerations: As defined in this specification.
12631 | In addition, as this media type uses the `+xml` convention, it shares the same security considerations as
12632 described in [RFC3023], Section 10.
12633 Interoperability considerations:
12634 There are no known interoperability issues.
12635 Published specification: This specification.
12636 Applications that use this media type:
12637 No known applications currently use this media type.
12638 Additional information:
12639 Magic number(s):
12640 | As specified for `application/xml` in [RFC3023], Section 3.2.
12641 File extension: `.cmisacl`
12642 Fragment identifiers:
12643 | As specified for `application/xml` in [RFC3023], Section 5.
12644 Base URI:
12645 As specified in [RFC3023], Section 6.
12646 Macintosh File Type code: `TEXT`

12647 Person and email address to contact for further information:
12648 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
12649 Intended usage: COMMON
12650 Author/Change controller: IESG
12651

6 Conformance

12695

12696 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level
12697 requirements defined within this specification.

12698 Specification:

12699 This specification references a number of other specifications (see the table above). In order to
12700 comply with this specification, an implementation MUST implement the portions of referenced
12701 specifications necessary to comply with the required provisions of this specification. Additionally,
12702 the implementation of the portions of the referenced specifications that are specifically cited in
12703 this specification MUST comply with the rules for those portions as established in the referenced
12704 specification.

12705

12706 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level
12707 requirements defined within this specification.

12708

12709

12710 Domain Model:

12711 Normative text within this specification takes precedence over the CMIS Core XML Schema.

12712 That is, the normative text in this specification further constrains the schemas and/or WSDL that
12713 are part of this specification; and this specification contains further constraints on the elements
12714 defined in referenced schemas.

12715

12716 Clients:

12717 Client implementations MAY implement either Restful AtomPub Binding or the Web
12718 Services Binding.

12719

12720 Repositories:

12721 Repositories MUST implement the following CMIS protocol bindings:

12722 [i.](#) Restful AtomPub Binding

12723 [ii.](#) Web Services Binding

12724

12725 Rest Binding:

12726 This specification references a number of other specifications. In order to comply with this
12727 specification, an implementation MUST implement the portions of referenced specifications
12728 necessary to comply with the required provisions of this specification. Additionally, the
12729 implementation of the portions of the referenced specifications that are specifically cited in this
12730 specification MUST comply with the rules for those portions as established in the referenced
12731 specification.

12732 Additionally normative text within this specification takes precedence over the CMIS RestAtom
12733 XML Schema. That is, the normative text in this specification further constrains the schemas
12734 and/or WSDL that are part of this specification; and this specification contains further constraints
12735 on the elements defined in referenced schemas.

12736 The CMIS RestAtom XML takes precedence over any examples or non-normative outlines
12737 included either in this document or as standalone examples.

12738

12739 Web Services Binding:

12740 Normative text within this specification takes precedence over the CMIS Messaging XML and
12741 CMIS WSDL. That is, the normative text in this specification further constrains the schemas and
12742 WSDL that are part of this specification; and this specification contains further constraints on the
12743 elements defined in referenced schemas.

12744 The CMIS Messaging XML and CMIS WSDL takes precedence over any examples or non-
12745 normative outlines included either in this document or as standalone examples.

12746

A. Acknowledgements

12747 The following individuals have participated in the creation of this specification and are gratefully
12748 acknowledged:

12749

12750 **Participants:**

12751 Philippe Allart, Adullact
12752 Florian Bartels, fme AG
12753 Fred Boiscuvier, Exalead, Inc.
12754 Al Brown, IBM
12755 Jay Brown, IBM
12756 Mark Carlson, Sun Microsystems
12757 Derek Carr, IBM
12758 David Caruana, Alfresco Software
12759 Eric Chan, Oracle Corporation
12760 Sameer Charles, Magnolia International AG
12761 Derek Chow, Genus Technologies, LLC
12762 David Choy, EMC Corporation
12763 Scott Conroy, Individual
12764 Cornelia Davis, EMC Corporation
12765 Doug Domeny, Ektron
12766 Kevin Dorr, Flatirons Solutions Corporation
12767 Jason Dubreuil, Fidelity Investments
12768 Michael Duerig, Day Software
12769 Randy Dufault, Genus Technologies, LLC
12770 Will Ezell, dotCMS
12771 Betsy Fanning, AIIM
12772 Steffen Frederiksen, Content Technologies ApS
12773 Stephan Friedl, Quark
12774 Dustin Friesenhahn, Microsoft Corporation
12775 Gary Gershon, Individual
12776 Paul Goetz, SAP AG
12777 Jens Goldhammer, fme AG
12778 Gregory Grefenstette, Exalead, Inc.
12779 Florent Guillaume, Nuxeo
12780 Ethan Gur-esh, Microsoft Corporation
12781 Alexander Haag, WeWebU Software AG
12782 Dennis Hamilton, Individual
12783 Martin Hermes, SAP AG
12784 Jens Huebel, Open Text Corporation
12785 David Izatt, Structured Software Systems Limited (3SL)
12786 Gershon Janssen, Individual
12787 Raphael Jean, Entropysoft
12788 Volker John, Saperion AG
12789 Shane Johnson, Citytech, Inc.
12790 Christophe Kijewska, Adullact
12791 Ijonas Kisselbach, Vamosa
12792 Mark Klamerus, Individual
12793 Stephan Klevenz, SAP AG
12794 Boris Kraft, Magnolia International AG
12795 Alison Macmillan, Oracle Corporation
12796 Michael Marth, Day Software
12797 Mary McRae, OASIS
12798 Ryan McVeigh, Oracle Corporation

12799 Juerg Meier, fme AG
12800 Gregory Melahn, IBM
12801 Pat Miller, Microsoft Corporation
12802 Florian Müller, Open Text Corporation
12803 Thomas Mueller, Day Software
12804 John Newton, Alfresco Software
12805 David Nuescheler, Day Software
12806 Conleth O'Connell, Vignette Corporation
12807 Marc Pallot, ESoCE-NET
12808 Rainer Pausch, WeWebU Software AG
12809 Dominique Pfister, Day Software
12810 Peeter Piegaze, Day Software
12811 David Pitfield, Oracle Corporation
12812 Thomas Pole, Harris Corp
12813 Norrie Quinn, EMC Corporation
12814 Craig Randall, Adobe Corporation
12815 Julian Reschke, Greenbytes GmbH
12816 Celso Rodriguez, ASG Software Solutions
12817 Steve Roth, Oracle Corporation
12818 Patrick Ryan, IBM
12819 Angela Schreiber, Day Software
12820 Spencer Shearer, Exalead, Inc.
12821 Madi Solomon, Pearson PLC
12822 Wojciech Specht, fme AG
12823 Dmitri Tcherevik, FatWire
12824 Jason Tesser, dotCMS
12825 David Torres, dotCMS
12826 Maik Uhlenberg, fme AG
12827 Oliver Walthard, Day Software
12828 Patrick Ward, Booz Allen Hamilton

12829
12830 Original Authors of the initial contribution:

12831 Al Brown, IBM
12832 David Choy, EMC
12833 Cornelia Davis, EMC
12834 Ethan Gur-Esh, Microsoft

12835
12836 Original Acknowledgements of the initial contribution:

12837 Al Brown, IBM
12838 David Caruana, Alfresco
12839 Derek Carr, IBM
12840 David Choy, EMC
12841 Cornelia Davis, EMC
12842 Paul Goetz, SAP
12843 Ethan Gur-Esh, Microsoft
12844 Martin Hermes, SAP
12845 Jens Hubel, OpenText
12846 Jay Brown, IBM
12847 Ryan McVeigh, Oracle
12848 Gregory Melahn, IBM
12849 Florian Müller, OpenText
12850 John Newton, Alfresco
12851 Norrie Quinn, EMC
12852 Steve Roth, Oracle
12853 Craig Randall, EMC

12854

B. Non-Normative Text

12855

12856

C. Revision History

12857

Revision	Date	Editor	Changes Made
1.0	01/11/2010	Al Brown	First specification

12858