



Content Management Interoperability Services (CMIS) Version 1.0

Committee Draft 04

23 September 2009

Specification URIs:**This Version:**

<http://docs.oasis-open.org/cmis/CMIS/v1.0/cd04/cmis-spec-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/cmis/CMIS/v1.0/cd04/cmis-spec-v1.0.html>
<http://docs.oasis-open.org/cmis/CMIS/v1.0/cd04/cmis-spec-v1.0.pdf>

Previous Version:

N/A

Latest Version:

<http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-spec-v1.0.html>
<http://docs.oasis-open.org/cmis/CMIS/v1.0/cmis-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, Oracle
Florian Muller, OpenText

Related work:

N/A

Declared XML Namespace(s):

<http://docs.oasis-open.org/ns/cmis/core/200908/>
<http://docs.oasis-open.org/ns/cmis/restatom/200908/>
<http://docs.oasis-open.org/ns/cmis/messaging/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. 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	16
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	23
2.1.4.2	Renditions	23
2.1.4.3	Document Object-Type Definition	25
2.1.5	Folder Object	32
2.1.5.1	File-able Objects	32
2.1.5.2	Folder Hierarchy	33
2.1.5.3	Paths	34
2.1.5.4	Folder Object-Type Definition	35
2.1.6	Relationship Object	39
2.1.6.1	Relationship Object-Type Definition	40
2.1.7	Policy Object	44
2.1.7.1	Policy Object-Type Definition	45
2.1.8	Access Control	49
2.1.8.1	ACL, ACE, Principal, and Permission	49
2.1.8.2	CMIS Permissions	49
2.1.8.3	ACL Capabilities	49
2.1.9	Versioning	58
2.1.9.1	Version Series	58
2.1.9.2	Latest Version	58
2.1.9.3	Major Versions	58
2.1.9.4	Services that modify Version Series	59
2.1.9.5	Versioning Properties on Document Objects	60
2.1.9.6	Document Creation and Initial Versioning State	61
2.1.9.7	Version Specific/Independent membership in Folders	61
2.1.9.8	Version Specific/Independent membership in Relationships	61
2.1.9.9	Versioning visibility in Query Services	62
2.1.10	Query	62
2.1.10.1	Relational View Projection of the CMIS Data Model	63
2.1.10.2	Query Language Definition	64

2.1.10.3 Escaping	73
2.1.11 Change Log	73
2.1.11.1 Completeness of the Change Log.....	74
2.1.11.2 Change Log Token	74
2.1.11.3 Change Event	74
2.2 Services	74
2.2.1 Common Service Elements.....	75
2.2.1.1 Paging.....	75
2.2.1.2 Retrieving additional information on objects in CMIS service calls	75
2.2.1.3 Change Tokens.....	77
2.2.1.4 Exceptions	78
2.2.1.5 ACLs	81
2.2.2 Repository Services	81
2.2.2.1 getRepositories	81
2.2.2.2 getRepositoryInfo.....	82
2.2.2.3 getTypeChildren.....	83
2.2.2.4 getTypeDescendants	84
2.2.2.5 getTypeDefinition	85
2.2.3 Navigation Services.....	85
2.2.3.1 getChildren	85
2.2.3.2 getDescendants	86
2.2.3.3 getFolderTree	87
2.2.3.4 getFolderParent	88
2.2.3.5 getObjectParents	89
2.2.3.6 getCheckedOutDocs.....	89
2.2.4 Object Services	90
2.2.4.1 createDocument.....	90
2.2.4.2 createDocumentFromSource.....	92
2.2.4.3 createFolder.....	93
2.2.4.4 createRelationship	94
2.2.4.5 createPolicy	95
2.2.4.6 getAllowableActions.....	96
2.2.4.7 getObject	96
2.2.4.8 getProperties.....	97
2.2.4.9 getObjectByPath.....	97
2.2.4.10 getContentStream.....	98
2.2.4.11 getRenditions	98
2.2.4.12 updateProperties.....	99
2.2.4.13 moveObject.....	100
2.2.4.14 deleteObject.....	100
2.2.4.15 deleteTree.....	101
2.2.4.16 setContentStream	101
2.2.4.17 deleteContentStream	102
2.2.5 Multi-filing Services	103
2.2.5.1 addObjectToFolder	103
2.2.5.2 removeObjectFromFolder	103
2.2.6 Discovery Services.....	104
2.2.6.1 query.....	104
2.2.6.2 getContentChanges.....	105

2.2.7 Versioning Services.....	106
2.2.7.1 checkOut.....	106
2.2.7.2 cancelCheckOut.....	107
2.2.7.3 checkIn	107
2.2.7.4 getObjectOfLatestVersion	108
2.2.7.5 getPropertiesOfLatestVersion.....	109
2.2.7.6 getAllVersions	109
2.2.8 Relationship Services.....	110
2.2.8.1 getObjectRelationships	110
2.2.9 Policy Services	111
2.2.9.1 applyPolicy	111
2.2.9.2 removePolicy	111
2.2.9.3 getAppliedPolicies.....	112
2.2.10 ACL Services.....	112
2.2.10.1 getACL.....	112
2.2.10.2 applyACL	113
3 Restful AtomPub Binding	115
3.1 Overview	115
3.1.1 Namespaces.....	115
3.1.2 Authentication.....	115
3.1.3 Response Formats	115
3.1.4 Optional Arguments.....	116
3.1.5 Errors and Exceptions	116
3.1.6 Renditions.....	116
3.1.7 Content Streams	116
3.1.8 Paging of Feeds	116
3.1.9 Services not Exposed.....	116
3.2 HTTP.....	117
3.2.1 Entity Tag	117
3.2.2 HTTP Range.....	117
3.2.3 HTTP OPTIONS Method.....	117
3.2.4 HTTP Status Codes	117
3.2.4.1 Common CMIS Exceptions.....	117
3.2.4.2 Other Exceptions	117
3.2.4.3 Notable HTTP Status Codes.....	117
3.3 Media Types	118
3.3.1 CMIS Atom	118
3.3.2 CMIS Query.....	119
3.3.3 CMIS Allowable Actions	119
3.3.4 CMIS Tree	120
3.3.5 CMIS ACL.....	124
3.4 Atom Extensions for CMIS.....	125
3.4.1 Atom Element Extensions	125
3.4.1.1 AtomPub Workspace	125
3.4.1.2 Atom Feed	125
3.4.1.3 Atom Entry	125
3.4.2 Attributes	126

3.4.2.1 cmisra:id	126
3.4.2.2 cmisra:renditionKind	127
3.4.3 CMIS Link Relations	127
3.4.3.1 Existing Link Relations	127
3.4.3.2 Hierarchy Navigation Internet Draft Link Relations	129
3.4.3.3 Versioning Internet Draft Link Relations	129
3.4.3.4 CMIS Specific Link Relations	130
3.5 Atom Resources	131
3.5.1 Feeds	131
3.5.2 Entries	132
3.5.2.1 Hierarchical Atom Entries	133
3.6 AtomPub Service Document (Repository)	134
3.6.1 URI Templates	136
3.6.1.1 Object By Id	137
3.6.1.2 Object By Path	138
3.6.1.3 Query	139
3.6.1.4 Type By Id	139
3.6.2 HTTP Methods	140
3.6.2.1 GET	140
3.7 Service Collections	140
3.7.1 Root Folder Collection	140
3.7.2 Query Collection	141
3.7.2.1 POST	141
3.7.3 Checked Out Collection	143
3.7.3.1 GET	144
3.7.3.2 POST	144
3.7.4 Unfiled Collection	147
3.7.4.1 GET	148
3.7.4.2 POST	148
3.7.5 Types Children Collection	151
3.7.5.1 GET	152
3.8 Collections	152
3.8.1 Relationships Collection	152
3.8.1.1 GET	153
3.8.1.2 POST	153
3.8.2 Folder Children Collection	156
3.8.2.1 GET	157
3.8.2.2 POST	157
3.8.3 Policies Collection	164
3.8.3.1 GET	165
3.8.3.2 POST	165
3.9 Feeds	167
3.9.1 Object Parents Feed	167
3.9.1.1 GET	170
3.9.2 Changes	170
3.9.2.1 GET	175
3.9.3 Folder Descendants	175

3.9.3.1 GET	181
3.9.3.2 DELETE	181
3.9.4 Folder Tree	182
3.9.4.1 GET	185
3.9.4.2 DELETE	185
3.9.5 AllVersions Feed	185
3.9.5.1 GET	187
3.9.5.2 DELETE	187
3.9.6 Type Descendants Feed	187
3.9.6.1 GET	195
3.10 Resources	195
3.10.1 Type Entry	195
3.10.1.1 GET	196
3.10.2 Document Entry	197
3.10.2.1 GET	198
3.10.2.2 PUT	200
3.10.2.3 DELETE	200
3.10.3 Document Private Working Copy (PWC) Entry	200
3.10.3.1 GET	201
3.10.3.2 PUT	203
3.10.3.3 DELETE	203
3.10.4 Folder Entry	203
3.10.4.1 GET	204
3.10.4.2 PUT	205
3.10.4.3 DELETE	206
3.10.5 Relationship Entry	206
3.10.5.1 GET	206
3.10.5.2 PUT	208
3.10.5.3 DELETE	208
3.10.6 Policy Entry	208
3.10.6.1 GET	208
3.10.6.2 PUT	210
3.10.6.3 DELETE	210
3.10.7 Content Stream	210
3.10.7.1 GET	210
3.10.7.2 PUT	210
3.10.7.3 DELETE	211
3.10.8 ACL Resource	211
3.10.8.1 GET	211
4 Web Services Binding	213
4.1 Overview	213
4.1.1 WS-I	213
4.1.2 Authentication	213
4.1.3 Content Transfer	213
4.1.4 Reporting Errors	213
4.2 Web Services Binding Mapping	213
4.3 Additions to Part I	213

4.3.1 updateProperties and checkIn Semantics.....	213
4.3.2 Content Ranges	214
4.3.3 Extensions.....	214
4.3.4 Web Services Specific Structures	214
4.3.4.1 cmisFaultType and cmisFault	214
4.3.4.2 cmisRepositoryEntryType	214
4.3.4.3 cmisTypeContainer	214
4.3.4.4 cmisTypeDefinitionListType	214
4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and cmisObjectInFolderContainerType	214
4.3.4.6 cmisObjectType and cmisObjectInFolderListType	215
4.3.4.7 cmisContentStreamType	215
4.3.4.8 cmisACLType.....	215
4.3.4.9 cmisExtensionType.....	215
5 IANA Considerations	216
5.1 Content-Type Registration.....	216
5.1.1 CMIS Query.....	216
5.1.2 CMIS AllowableActions	216
5.1.3 CMIS Tree	217
5.1.4 CMIS Atom	218
5.1.5 CMIS ACL.....	219
6 Conformance.....	221
A. Acknowledgements	223
B. Non-Normative Text	225
C. Revision History.....	226

1 **Introduction**

2 The Content Management Interoperability Services (CMIS) ReSTful AtomPub binding specification
3 defines a specification based on AtomPub that can be used by applications to work with one or more
4 Content Management Repositories.

5

6 **1.1 Terminology**

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

10 **1.2 Normative References**

- | | |
|--------------|--|
| 11 [RFC4287] | M. Nottingham, R. Sayre, <i>Atom Syndication Format</i> ,
http://www.ietf.org/rfc/rfc4287.txt , December 2005 |
| 12 [RFC5023] | J. Gregorio, B. de hOra, <i>Atom Publishing Protocol</i> ,
http://www.ietf.org/rfc/rfc5023.txt , October 2007 |
| 13 [RFC2616] | R. Fielding, J. Gettys, J. Mogul, H. Frystyk, L. Masinter, P. Leach, T. Berners-Lee, <i>Hypertext Transfer Protocol --HTTP/1.1</i> , http://www.ietf.org/rfc/rfc2616.txt , June 1999 |
| 14 [RFC2119] | S. Bradner, Key words for use in RFCs to Indicate Requirement Levels.
http://www.ietf.org/rfc/rfc2119.txt March 1997 |
| 15 [RFC4918] | L. Dusseault, <i>HTTP Extensions for Web Distributed Authoring and Versioning (WebDAV)</i> , June 2007 |
| 16 [RFC3986] | T. Berners-Lee, R. Fielding, L. Masinter, <i>Unified Resource Identifier</i> , January 2005 |
| 17 [CMISDM] | OASIS Committee Draft 0.63, “Content Management Interoperability Services (CMIS) Domain Model”, March 2009 |

26 **1.3 Non-Normative References**

27
28
29
30
31

32 2 Domain Model

33 2.1 Data Model

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

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

51 2.1.1 Repository

52 The repository itself is described by the CMIS “Get Repository Information” service. The service output is
53 fully described in section 2.2.2.2 *getRepositoryInfo*.

54 2.1.1.1 Optional Capabilities

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

62

63 Navigation Capabilities:

64 **capabilityGetDescendants**

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

67 See section: 2.2.3.2 *getDescendants*

68

69 **capabilityGetFolderTree**

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

71 See section: 2.2.3.1 *getFolderTree*

72

73 **Object Capabilities:**

74 **capabilityContentStreamUpdatability** (`enumCapabilityContentStreamUpdates`)

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

77 • **none**: The content stream may never be updated.

78 • **anytime**: The content stream may be updated any time.

79 • **pwconly**: The content stream may be updated only when checked out. The abbreviation
80 PWC is described in section 2.1.9 Versioning.

81 See Section: 2.1.4.1 Content Stream

82

83 **capabilityChanges** (`enumCapabilityChanges`)

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

86 • **none**: The repository does not support the change log feature.

87 • **objectidsonly**: The change log can return only the ObjectIDs for changed objects in
88 the repository and an indication of the type of change, not details of the actual change.

89 • **properties**: The change log can return properties and the ObjectID for the changed
90 objects

91 • **all**: The change log can return the ObjectIDs for changed objects in the repository and
92 more information about the actual change

93 See Section: 2.1.11 Change Log

94

95 **capabilityRenditions** (`enumCapabilityRendition`)

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

97 • **none**: The repository does not expose renditions at all.

98 • **read**: Renditions are provided by the repository and readable by the client.

99

100 **Filing Capabilities:**

101 **capabilityMultifiling**

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

103 See Section: 2.1.5 Folder Object

104

105 **capabilityUnfiling**

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

107 See Section: 2.1.5 Folder Object

108

109 **capabilityVersionSpecificFiling**

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

111 See Section: 2.1.9 Versioning

112

113 **Versioning Capabilities:**

114 **capabilityPWCUpdatable**

115 Ability for an application to update the "Private Working Copy" of a checked-out document
 116 See Section: 2.1.9 Versioning
 117
 118 **capabilityPWCSearchable**
 119 Ability of the Repository to include the "Private Working Copy" of checked-out documents in
 120 query search scope; otherwise PWC's are not searchable
 121 See Section: 2.1.9 Versioning
 122
 123 **capabilityAllVersionsSearchable**
 124 Ability of the Repository to include non-latest versions of document in query search scope;
 125 otherwise only the latest version of each document is searchable
 126 See Section: 2.1.9 Versioning
 127
 128 **Query Capabilities:**
 129 **capabilityQuery (enumCapabilityQuery)**
 130 Indicates the types of queries that the Repository has the ability to fulfill. Query support levels are:
 131

- **none**: No queries of any kind can be fulfilled.

 132

- **metadataonly**: Only queries that filter based on object properties can be fulfilled.
Specifically, the CONTAINS() predicate function is not supported.

 133

- **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.

 134

- **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.

 135

- **bothcombined**: The repository can fulfill queries that filter on both the full-text content of
documents and their properties in the same query.

 136 See Section: 2.1.10 Query
 137
 138
 139
 140
 141
 142
 143
 144 **capabilityJoin (enumCapabilityJoin)**
 145 Indicates the types of JOIN keywords that the Repository can fulfill in queries. Support levels are:
 146

- **none**: The repository cannot fulfill any queries that include any JOIN clauses.

 147

- **inneronly**: The repository can fulfill queries that include an INNER JOIN clause, but
cannot fulfill queries that include other types of JOIN clauses.

 148

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

 149 See Section: 2.1.10 Query
 150
 151
 152
 153 **ACL Capabilities:**
 154 **capabilityACL (enumCapabilityACL)**
 155 Indicates the level of support for ACLs by the repository
 156

- **none**: The repository does not support ACL services

 157

- **discover**: The repository supports discovery of ACLs (getACL and other services)

- 158 • **manage**: The repository supports discovery of ACLs AND applying ACLs (getACL and
159 applyACL services)

160 See Section: 2.8 Access Control

161 **2.1.1.2 Implementation Information**

162 The “Get Repository Information” service MUST also return implementation information including vendor
163 name, product name, product version, version of CMIS that it supports, the root folder ID (see section
164 2.1.5.2 Folder Hierarchy), and MAY include other implementation-specific information. The version of
165 CMIS that the repository supports MUST be expressed as a Decimal that matches the specification
166 version.

167 **2.1.2 Object**

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

- 170 • A *document object* represents a standalone information asset. Document objects are the
171 elementary entities managed by a CMIS repository.
- 172 • A *folder object* represents a logical container for a collection of “file-able” objects, which include
173 folder objects and document objects. Folder objects are used to organize file-able objects.
174 Whether or not an object is file-able is specified in its [object-type definition](#).
- 175 • A *relationship object* represents an instance of directional relationship between two objects. The
176 support for relationship objects is optional, and may be discovered via the “Get Type Children”
177 service.
- 178 • A *policy object* represents an administrative policy, which may be “applied” to one or more
179 “controllablePolicy” objects. Whether or not an object is controllable is specified in its object-type
180 definition. The support for policy objects is optional, and may be discovered via the “Get Type
181 Children” service.

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

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

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

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

199 Document or folder objects MAY have one **Access Control List** (ACL), which controls access to the
200 document or folder. A policy object may also control access to the document or folder. An ACL

201 represents a list of **Access Control Entries** (ACEs). An ACE in turn represents one or more permissions
202 being granted to a **principal** (a user, group, role, or something similar).

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

204 **2.1.2.1 Property**

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

209 If a value is not provided for a property, the property is in a “*value not set*” state. There is no “null” value
210 for a property. Through protocol binding, a property is either not set, or is set to a particular value or a list
211 of values.

212 A multi-valued property is either set or not set in its entirety. An individual value of a multi-valued property
213 MUST NOT be in an individual “*value not set*” state and hold a position in the list of values. An empty list
214 of values MUST NOT be allowed.

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

- 219 • string (xsd:string)
- 220 • boolean (xsd:boolean)
- 221 • decimal (see section 2.1.3.3.4 Attributes specific to Decimal Object-Type Property Definitions)
- 222 • integer (xsd:integer)
- 223 • datetime (xsd:dateTime and see section 2.1.3.3.4 Attributes specific to Decimal Object-Type
224 Property Definitions)
- 225 • uri (xsd:anyURI)

226

227

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

- 229 • id
 - 230 • html
- 231 Individual protocol bindings MAY override or re-specify these property types.

232

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

236

237 The string MUST NOT contain:

- 238 • whitespace “ ”,
- 239 • comma “,”
- 240 • double quotes “””
- 241 • single quotes “””
- 242 • backslash “\”
- 243 • the period “.” character or,

- 244 • the open "(" or close ")" parenthesis characters.

245

246 **2.1.2.1.1 ID Property**

247 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 collateable 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 "equal" or a "not equal" comparison with a string literal or with another ID.

253 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.

262 **2.1.2.1.2 HTML Property**

263 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.

266 **2.1.3 Object-Type**

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

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

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

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

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

281 **2.1.3.1 Object-Type Hierarchy and Inheritance**

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

- 283 • A CMIS repository MUST have these base types:
- 284 ○ *cmis:document* object-type

- 285 ○ *cmis:folder* object-type
- 286 • A CMIS repository MAY have these base types:
- 287 ○ *cmis:relationship* object-type
- 288 ○ *cmis:policy* object-type
- 289 • Additional base types MUST NOT exist. Additional object-types MAY be defined as sub-types or
- 290 descendant types of these four base types.
- 291 • A **Base Type** does not have a parent type.
- 292 • A non-base type has one and only one parent type. An object-type's **Parent Type** is a part of the
- 293 object-type definition.
- 294 • An object-type definition includes a set of **object-type attributes** (e.g. Fileable, Queryable, etc.)
- 295 and a property schema that will apply to Objects of that type.
- 296 ○ There is no inheritance of object-type attributes from a parent object-type to its sub-types.
- 297 • The properties of a CMIS base type MUST be inherited by its descendant types.
- 298 • A **Child Type** whose immediate parent is NOT its base type SHOULD inherit all the property
- 299 definitions that are specified for its parent type. In addition, it MAY have its own property
- 300 definitions.
- 301 ○ If a property is NOT inherited by a subtype, the exhibited behavior for query MUST be as if
- 302 the value of this property is "not set" for all objects of this sub-type.
- 303 • The scope of a query on a given object-type is automatically expanded to include all the
- 304 **Descendant Types** of the given object-type with the attribute `includedInSuperTypeQuery`
- 305 equals TRUE. This was added for synthetic types as well as to support different type hierarchies
- 306 that are not necessarily the same as CMIS. Only the properties of the given object-type,
- 307 including inherited ones, MUST be used in the query. Properties defined for its descendant types
- 308 MAY NOT be used in the query, and CAN NOT be returned by the query.
- 309 ○ If a property of its parent type is not inherited by this type, the property MUST still appear as
- 310 a column in the corresponding virtual table in the relational view, but this column MUST
- 311 contain a NULL value for all objects of this type. (See section 2.1.10 Query.)

312 2.1.3.2 Object-Type Attributes

313 2.1.3.2.1 Attributes common to ALL Object-Type Definitions

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

- | | |
|---|-------------------|
| 315 id | ID |
| 316 This opaque attribute uniquely identifies this object-type in the repository. | |
| 317 | |
| 318 localName | String (optional) |
| 319 This attribute represents the underlying repository's name for the object-type. This field is | |
| 320 opaque and has no uniqueness constraint imposed by this specification. | |
| 321 | |
| 322 localNamespace | String (optional) |
| 323 This attribute allows repositories to represent the internal namespace of the underlying | |
| 324 repository's name for the object-type. | |
| 325 | |
| 326 queryName | String |

327 Used for query and filter operations on object-types. This is an opaque String with limitations.
 328 This string SHOULD NOT contain any characters that negatively interact with the BNF grammar.
 329
 330 The string MUST NOT contain:
 331 • whitespace “ “,
 332 • comma “ ,”
 333 • double quotes “””
 334 • single quotes “””
 335 • backslash “\”
 336 • the period “.” character or,
 337 • the open “(“ or close “)” parenthesis characters.
 338
 339 **displayName** String (optional)
 340 Used for presentation by application.
 341
 342 **baseId** Enum
 343 A value that indicates whether the base type for this Object-Type is the Document, Folder,
 344 Relationship, or Policy base type.
 345
 346 **parentId** ID
 347 The ID of the Object-Type's immediate parent type.
 348 It MUST be “not set” for a base type.
 349
 350 **description** String (optional)
 351 Description of this object-type, such as the nature of content, or its intended use. Used for
 352 presentation by application.
 353
 354 **creatable** Boolean
 355 Indicates whether new objects of this type MAY be created. If the value of this attribute is FALSE,
 356 the repository MAY contain objects of this type already, but MUST NOT allow new objects of this
 357 type to be created.
 358
 359 **fileable** Boolean
 360 Indicates whether or not objects of this type are [fileable](#).
 361
 362 **queryable** Boolean
 363 Indicates whether or not this object-type can appear in the FROM clause of a query statement. A
 364 non-queryable object-type is not visible through the relational view that is used for query, and
 365 CAN NOT appear in the FROM clause of a query statement.
 366
 367 **controllablePolicy** Boolean
 368 Indicates whether or not objects of this type are controllable via policies. Policy objects can only
 369 be applied to controllablePolicy objects.
 370

371 **controllableACL** Boolean
372 This attribute indicates whether or not objects of this type are controllable by ACL's. Only objects
373 that are controllableACL can have an ACL.
374
375 **fulltextIndexed** Boolean
376 Indicates whether objects of this type are indexed for full-text search for querying via the
377 CONTAINS() query predicate.
378
379 **includedInSupertypeQuery** Boolean
380 Indicates whether this type and its subtypes appear in a query of this type's ancestor types.
381 For example: if Invoice is a sub-type of cmis:document, if this is TRUE on Invoice then for a query
382 on cmis:document, instances of Invoice will be returned if they match.
383 If this attribute is FALSE, no instances of Invoice will be returned even if they match the query.

384 **2.1.3.3 Object-Type Property Definitions**

385 Besides these object-type attributes, an object-type definition SHOULD contain inherited property
386 definitions and zero or more additional property definitions. All the properties of an object, including
387 inherited properties, MUST be retrievable through the "get" services, and MAY appear in the SELECT
388 clause of a query.

389 **2.1.3.3.1 Property Types**

390 Property types are defined in section 2.1.2.1 Property.

391 **2.1.3.3.2 Attributes common to ALL Object-Type Property Definitions**

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

393 **id** ID
394 This opaque attribute uniquely identifies the property in the repository. If two Object-Types each
395 contain property definitions with the same ID, those property definitions are the same.
396
397 **localName** String (optional)
398 This attribute represents the underlying repository's name for the property. This field is opaque
399 and has no uniqueness constraint imposed by this specification.
400
401 **localNamespace** String (optional)
402 This attribute allows repositories to represent the internal namespace of the underlying
403 repository's name for the property.
404
405 **queryName** String
406 Used for query operations on properties. This is an opaque String with limitations. Please see
407 queryName in Object-Type Attributes for the limitations on what characters are not allowed.
408
409 **displayName** String (optional)
410 Used for presentation by application.
411

412 **description** String (optional)
 413 This is an optional attribute containing a description of the property
 414
 415 **propertyType** Enum
 416 This attribute indicates the type of this property. It MUST be one of the allowed property types.
 417 (See section 2.1.2.1 Property.)
 418
 419 **cardinality** Enum
 420 Indicates whether the property can have “zero or one” or “zero or more” values.
 421 Values:
 422 • **single**: Property can have zero or one values (if property is not required), or exactly one value (if property is required)
 423
 424 • **multi**: Property can have zero or more values (if property is not required), or one or more values (if property is required).
 425
 426 Repositories SHOULD preserve the ordering of values in a multi-valued property. That is, the order in which the values of a multi-valued property are returned in get operations SHOULD be the same as the order in which they were supplied during previous create/update operation.
 427
 428
 429
 430 **updatability** Enum
 431 Indicates under what circumstances the value of this property MAY be updated.
 432 Values:
 433 • **readonly**: The value of this property MUST NOT ever be set directly by an application. It is a system property that is either maintained or computed by the repository.
 434
 435 ◦ The value of a readOnly property MAY be indirectly modified by other repository interactions (for example, calling “updateProperties” on an object will change the object’s last modified date, even though that property cannot be directly set via an updateProperties() service call.)
 436
 437
 438 • **readwrite**: The property value can be modified using the *updateProperties* service.
 439
 440 • **whencheckedout**: The property value MUST only be update-able using a “[private working copy](#)” Document.
 441
 442 ◦ I.e. the update is either made on a “private working copy” object or made using a “check in” service.
 443
 444 • **oncreate**: The property value MUST only be update-able during the Create operation on that Object.
 445
 446
 447 **inherited** Boolean
 448 Indicates whether the property definition is inherited from the parent-type when TRUE or it is explicitly defined for this object-type when FALSE.
 449
 450
 451 **required** Boolean
 452 If TRUE, then the value of this property MUST never be set to the “not set” state when an object of this type is created/updated. If not provided during a create or update operation, the repository MUST provide a value for this property.
 453
 454 If a value is not provided, then the default value defined for the property MUST be set. If no default value is provided and no default value is defined, the repository MUST throw an exception.
 455
 456
 457

458 A property definition SHOULD never state that a property has a “required” value of TRUE and an
 459 updatability value of “readonly”.

460

461 **queryable** Boolean
 462 Indicates whether or not the property MAY appear in the WHERE clause of a CMIS query
 463 statement.
 464 This attribute MUST have a value of FALSE if the Object-type’s attribute for “Queryable” is set to
 465 FALSE.

466

467 **orderable** Boolean
 468 Indicates whether the property can appear in the ORDER BY clause of a CMIS query statement
 469 or an ORDERBY parameter.
 470 This property MUST be FALSE for any property whose cardinality is “multi”.

471

472 **choices** <PropertyChoiceType list> (multi-valued)
 473 Indicates an explicit ordered set of values allowed for this property.
 474 If this attribute is “not set”, then any valid value for this property based on its type may be used.
 475 Each choice includes a displayName and a value. The displayName is used for presentation
 476 purpose. The value will be stored in the property when selected.
 477 Choices MAY be hierarchically presented.

478

479 **openChoice** Boolean
 480 This attribute is only applicable to properties that provide a value for the “Choices” attribute.
 481 If FALSE, then the data value for the property MUST only be one of the values specified in the
 482 “Choices” attribute. If TRUE, then values other than those included in the “Choices” attribute may
 483 be set for the property.

484

485 **defaultValue** <PropertyType>
 486 The value that the repository MUST set for the property if a value is not provided by an
 487 application when the object is created.
 488 If no default value is specified and an application creates an object of this type without setting a
 489 value for the property, the repository MUST attempt to store a “value not set” state for the
 490 property value. If this occurs for a property that is defined to be required, then the creation
 491 attempt MUST throw an exception.
 492 The attributes on the default value element are the same as the attributes on the property
 493 definition.

494 2.1.3.3 Attributes specific to Integer Object-Type Property Definitions

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

499 **minValue** Integer
 500 The minimum value allowed for this property.
 501 If an application tries to set the value of this property to a value lower than minValue, the
 502 repository MUST throw a **constraint** exception.

503
504 **maxValue** Integer
505 The maximum value allowed for this property.
506 If an application tries to set the value of this property to a value higher than maxValue, the
507 repository MUST throw a **constraint** exception.
508

509 **2.1.3.3.4 Attributes specific to DateTime Object-Type Property Definitions**

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

514 **resolution** String Enumeration
515 This is the precision in bits supported for values of this property. Valid values for this attribute are:
516 • Year: Year resolution is persisted
517 • Date: Date resolution is persisted
518 • Time: Time resolution is persisted
519

520 **2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions**

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

525 **precision** Integer Enumeration
526 This is the precision in bits supported for values of this property. Valid values for this attribute are:
527 • 32: 32-bit precision (“single” as specified in IEEE-754-1985).
528 • 64: 64-bit precision (“double” as specified in IEEE-754-1985.)
529

530 **minValue** Decimal
531 The minimum value allowed for this property.
532 If an application tries to set the value of this property to a value lower than minValue, the
533 repository MUST throw a **constraint** exception.
534

535 **maxValue** Decimal
536 The maximum value allowed for this property.
537 If an application tries to set the value of this property to a value higher than maxValue, the
538 repository MUST throw a **constraint** exception.

539 **2.1.3.3.6 Attributes specific to String Object-Type Property Definitions**

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

544 **maxLength** Integer

545 The maximum length (in characters) allowed for a value of this property.
546 If an application attempts to set the value of this property to a string larger than the specified
547 maximum length, the repository MUST throw a **constraint** exception.

548 **2.1.4 Document Object**

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

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

- 551 • **Versionable:** Can be acted upon via the Versioning Services (for example: *checkOut*, *checkIn*).
- 552 • **Fileable:** Can be filed in zero, one, or more than one folder via the Multi-filing services.
- 553 • **Queryable:** Can be located via the Discovery Services (*query*).
- 554 • **Controllable-Policy:** Can have Policies applied to it (see section 2.1.7 Policy Object.)
- 555 • **Controllable-ACL:** Can have an ACL applied to it (see section 2.8 Access Control)

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

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

560 **2.1.4.1 Content Stream**

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

565 CMIS provides basic CRUD services for content-stream, using the ID of a content-stream's containing
566 document object for identification. A content stream also has a streamId which is used for access to the
567 stream. The "Set Content-Stream" service (*setContentStream*) either creates a new content-stream for a
568 document object or replaces an existing content-stream. The "Get Content-Stream" service
569 (*getContentStream*) retrieves a content-stream. The "Delete Content-Stream" service
570 (*deleteContentStream*) deletes a content-stream from a document object. In addition, the
571 "CreateDocument" and "Check-in" services MAY also take a content-stream as an optional input. A
572 content stream MUST be specified if required by the type definition. These are the only services that
573 operate on content-stream. The "Get Properties" and "Query" services, for example, do not return a
574 content-stream.

575 "Set Content-Stream" and "Delete Content-Stream" services are considered modifications to a content-
576 stream's containing document object, and SHOULD therefore change the object's *LastModificationDate*
577 property upon successful completion.

578 The ability to set or delete a content stream is controlled by the
579 **capabilityContentStreamUpdatability** capability.

580 **2.1.4.2 Renditions**

581 Some ECM repositories provide a facility to retrieve alternative representations of a document. These
582 alternative representations are known as renditions. This could apply to a preview case which would

583 enable the client to preview the content of a document without needing to download the full content.
584 Previews are generally reduced fidelity representations such as thumbnails. Renditions can take on any
585 general form, such as a PDF version of a word document.
586 A CMIS repository MAY expose zero or more renditions for a document or folder in addition to a
587 document's content stream. CMIS provides no capability to create or update renditions accessed through
588 the rendition services. Renditions are specific to the version of the document and may differ between
589 document versions. Each rendition consists of a set of rendition attributes and a rendition stream.
590 Rendition attributes are not object properties, and are not queryable. They can be retrieved using the
591 getRenditions service. A rendition stream can be retrieved using the getContentStream service with the
592 rendition's streamId parameter.

593 **2.1.4.2.1 Rendition Attributes**

594 A rendition has the following attributes:

595 **streamId** ID
596 Identifies the rendition stream.
597
598 **contentType** String
599 The MIME type of the rendition stream.
600
601 **length** Integer (optional)
602 The length of the rendition stream in bytes.
603
604 **title** String (optional)
605 Human readable information about the rendition.
606
607 **kind** String
608 A categorization String associated with the rendition.
609
610 **height** Integer (optional)
611 Typically used for 'image' renditions (expressed as pixels). SHOULD be present if kind =
612 cmis:thumbnail.
613
614 **width** Integer (optional)
615 Typically used for 'image' renditions (expressed as pixels). SHOULD be present if kind =
616 cmis:thumbnail.
617
618 **renditionDocumentId** ID (optional)
619 If specified, then the rendition can also be accessed as a document object in the CMIS services.
620 If not set, then the rendition can only be accessed via the rendition services. Referential integrity
621 of this ID is repository-specific.

622 **2.1.4.2.2 Rendition Kind**

623 A Rendition may be categorized via its **kind**. The repository is responsible for assigning kinds to
624 Renditions, including custom kinds. A repository kind does not necessarily identify a single Rendition for
625 a given Object.
626 CMIS defines the following kind:

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

630

2.1.4.3 Document Object-Type Definition

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

634

2.1.4.3.1 Attributes specific to Document Object-Types

635 The following Object **attributes** MUST only apply to Object-Type definitions whose baseld is the
 636 cmis:document Object-Type, in addition to the common attributes specified above:

637 **versionable** Boolean

638 Indicates whether or not objects of this type are version-able. (See section 2.1.9 Versioning.)

639 **contentStreamAllowed** Enum

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

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

647

2.1.4.3.2 Attribute Values

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

649 Notes:

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

654 **id**

655 Value: cmis:document

656 **localName**

657 Value: <repository-specific>

658 **localNamespace**

659 Value: <repository-specific>

660 **queryName**

661 Value: cmis:document

662 **displayName**

```
668      Value: <repository-specific>
669
670      baseId
671          Value: cmis:document
672
673      parentId
674          Value: Not set
675
676      description
677          Value: <repository-specific>
678
679      creatable
680          Value: <repository-specific>
681
682      fileable
683          Value: TRUE
684
685      queryable
686          Value: SHOULD be TRUE
687
688      controllablePolicy
689          Value: <repository-specific>
690
691      includedInSupertypeQuery
692          Value: <repository-specific>
693
694      versionable
695          Value: <repository-specific>
696
697      contentStreamAllowed
698          Value: <repository-specific>
699
700      controllableACL
701          Value: <repository-specific>
702
703      fulltextIndexed
704          Value: <repository-specific>
```

705 **2.1.4.3.3 Property Definitions**

706 The Document base Object-Type MUST have the following property definitions, and MAY include
707 additional property definitions. Any attributes not specified for the property definition are repository
708 specific. For all property definitions on base types, the query name MUST be the same as the property
709 ID. The repository MUST have the following property definitions on the Document Type:
710

711	cmis:name	Name of the object
712	Inherited:	False
713	Property Type:	String
714	Cardinality:	Single
715		
716	cmis:objectId	Id of the object
717	Required:	False
718	Inherited:	False
719	Property Type:	ID
720	Cardinality:	Single
721	Updatability:	Read Only
722	Choices:	Not Applicable
723	Open Choice:	Not Applicable
724	MUST be set on the object	
725		
726	cmis:baseTypeId	Id of the base object-type for the object
727	Required:	False
728	Inherited:	False
729	Property Type:	ID
730	Cardinality:	Single
731	Updatability:	Read Only
732	Choices:	Not Applicable
733	Open Choice:	Not Applicable
734	MUST be set on the object	
735		
736	cmis:objectTypeId	Id of the object's type
737	Required:	True
738	Inherited:	False
739	Property Type:	ID
740	Cardinality:	Single
741	Updatability:	oncreate
742	Choices:	Not Applicable
743	Open Choice:	Not Applicable
744	MUST be set on the object	
745		
746	cmis:createdBy	User who created the object.
747	Required:	False
748	Inherited:	False
749	Property Type:	String
750	Cardinality:	Single
751	Updatability:	Read Only
752	Choices:	Not Applicable

753	Open Choice:	Not Applicable
754	Queryable:	True
755	Orderable:	True
756	MUST be set on the object	
757		
758	cmis:creationDate	DateTime when the object was created.
759	Required:	False
760	Inherited:	False
761	Property Type:	DateTime
762	Cardinality:	Single
763	Updatability:	Read Only
764	Choices:	Not Applicable
765	Open Choice:	Not Applicable
766	Queryable:	True
767	Orderable:	True
768	MUST be set on the object	
769		
770	cmis:lastModifiedBy	User who last modified the object.
771	Required:	False
772	Inherited:	False
773	Property Type:	String
774	Cardinality:	Single
775	Updatability:	Read Only
776	Choices:	Not Applicable
777	Open Choice:	Not Applicable
778	Queryable:	True
779	Orderable:	True
780	MUST be set on the object	
781		
782	cmis:lastModificationDate	DateTime when the object was last modified.
783	Required:	False
784	Inherited:	False
785	Property Type:	DateTime
786	Cardinality:	Single
787	Updatability:	Read Only
788	Choices:	Not Applicable
789	Open Choice:	Not Applicable
790	Queryable:	True
791	Orderable:	True
792	MUST be set on the object	
793		
794	cmis:changeToken	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
795		

796	Required:	False
797	Inherited:	False
798	Property Type:	String
799	Cardinality:	Single
800	Updatability:	Read Only
801	Choices:	Not Applicable
802	Open Choice:	Not Applicable
803	MUST be set on the object	
804		
805	cmis:isImmutable	TRUE if the repository MUST throw an error at any attempt to update or delete the object.
806		
807	Required:	False
808	Inherited:	False
809	Property Type:	Boolean
810	Cardinality:	Single
811	Updatability:	Read Only
812	Choices:	Not Applicable
813	Open Choice:	Not Applicable
814	MUST be set on the object	
815		
816	cmis:isLatestVersion	See section 2.1.9 Versioning.
817	Required:	False
818	Inherited:	False
819	Property Type:	Boolean
820	Cardinality:	Single
821	Updatability:	Read Only
822	Choices:	Not Applicable
823	Open Choice:	Not Applicable
824	MUST be set on the object	
825		
826	cmis:isMajorVersion	See section 2.1.9 Versioning.
827	Required:	False
828	Inherited:	False
829	Property Type:	Boolean
830	Cardinality:	Single
831	Updatability:	Read Only
832	Choices:	Not Applicable
833	Open Choice:	Not Applicable
834	MUST be set on the object	
835		
836	cmis:isLatestMajorVersion	See section 2.1.9 Versioning.
837	Required:	False

838	Inherited:	False
839	Property Type:	Boolean
840	Cardinality:	Single
841	Updatability:	Read Only
842	Choices:	Not Applicable
843	Open Choice:	Not Applicable
844		MUST be set on the object
845		
846	cmis:versionLabel	See section 2.1.9 Versioning.
847	Required:	False
848	Inherited:	False
849	Property Type:	String
850	Updatability:	Read Only
851	Choices:	Not Applicable
852	Open Choice:	Not Applicable
853		MUST be set on the object
854		
855	cmis:versionSeriesId	See section 2.1.9 Versioning.
856	Required:	False
857	Inherited:	False
858	Property Type:	ID
859	Cardinality:	Single
860	Updatability:	Read Only
861	Choices:	Not Applicable
862	Open Choice:	Not Applicable
863		MUST be set on the object
864		
865	cmis:isVersionSeriesCheckedOut	See section 2.1.9 Versioning.
866	Required:	False
867	Inherited:	False
868	Property Type:	Boolean
869	Cardinality:	Single
870	Updatability:	Read Only
871	Choices:	Not Applicable
872	Open Choice:	Not Applicable
873		MUST be set on the object
874		
875	cmis:versionSeriesCheckedOutBy	See section 2.1.9 Versioning.
876	Required:	False
877	Inherited:	False
878	Property Type:	String
879	Cardinality:	Single

880	Updatability:	Read Only
881	Choices:	Not Applicable
882	Open Choice:	Not Applicable
883		
884	cmis:versionSeriesCheckedOutId	See section 2.1.9 Versioning.
885	Required:	False
886	Inherited:	False
887	Property Type:	ID
888	Cardinality:	Single
889	Updatability:	Read Only
890	Choices:	Not Applicable
891	Open Choice:	Not Applicable
892		
893	cmis:checkinComment	See section 2.1.9 Versioning.
894	Required:	False
895	Inherited:	False
896	Property Type:	String
897	Cardinality:	Single
898	Updatability:	Read Only
899	Choices:	Not Applicable
900	Open Choice:	Not Applicable
901		
902	cmis:contentStreamLength	Length of the content stream (in bytes).
903	Required:	False
904	Inherited:	False
905	Property Type:	Integer
906	Cardinality:	Single
907	Updatability:	Read Only
908	Choices:	Not Applicable
909	Open Choice:	Not Applicable
910	MUST be set if the document has a content stream	
911		
912	cmis:contentStreamMimeType	MIME type of the Content Stream
913	Required:	False
914	Inherited:	False
915	Property Type:	String
916	Cardinality:	Single
917	Updatability:	Read Only
918	Choices:	Not Applicable
919	Open Choice:	Not Applicable
920	MUST be set if the document has a content stream	
921		

922	cmis:contentStreamFileName	File name of the Content Stream
923	Required:	False
924	Inherited:	False
925	Property Type:	String
926	Cardinality:	Single
927	MUST be set if the document has a content stream	
928		
929	cmis:contentStreamId	Id of the stream
930	Required:	False
931	Inherited:	False
932	Property Type:	ID
933	Cardinality:	Single
934	Updatability:	Read Only
935	Choices:	Not Applicable
936	Open Choice:	Not Applicable

937 **2.1.5 Folder Object**

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

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

948 **2.1.5.1 File-able Objects**

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

951	cmis:folder	
952	MUST be file-able	
953		
954	cmis:document	
955	MUST be file-able	
956		
957	cmis:relationship	
958	MUST NOT be file-able	
959		
960	cmis:policy	
961	MAY be file-able	

962 **2.1.5.1.1 Document Version Series and Filing**

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

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

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

974 **2.1.5.1.2 Filing Restrictions by Object-Type**

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

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

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

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

986 **2.1.5.2 Folder Hierarchy**

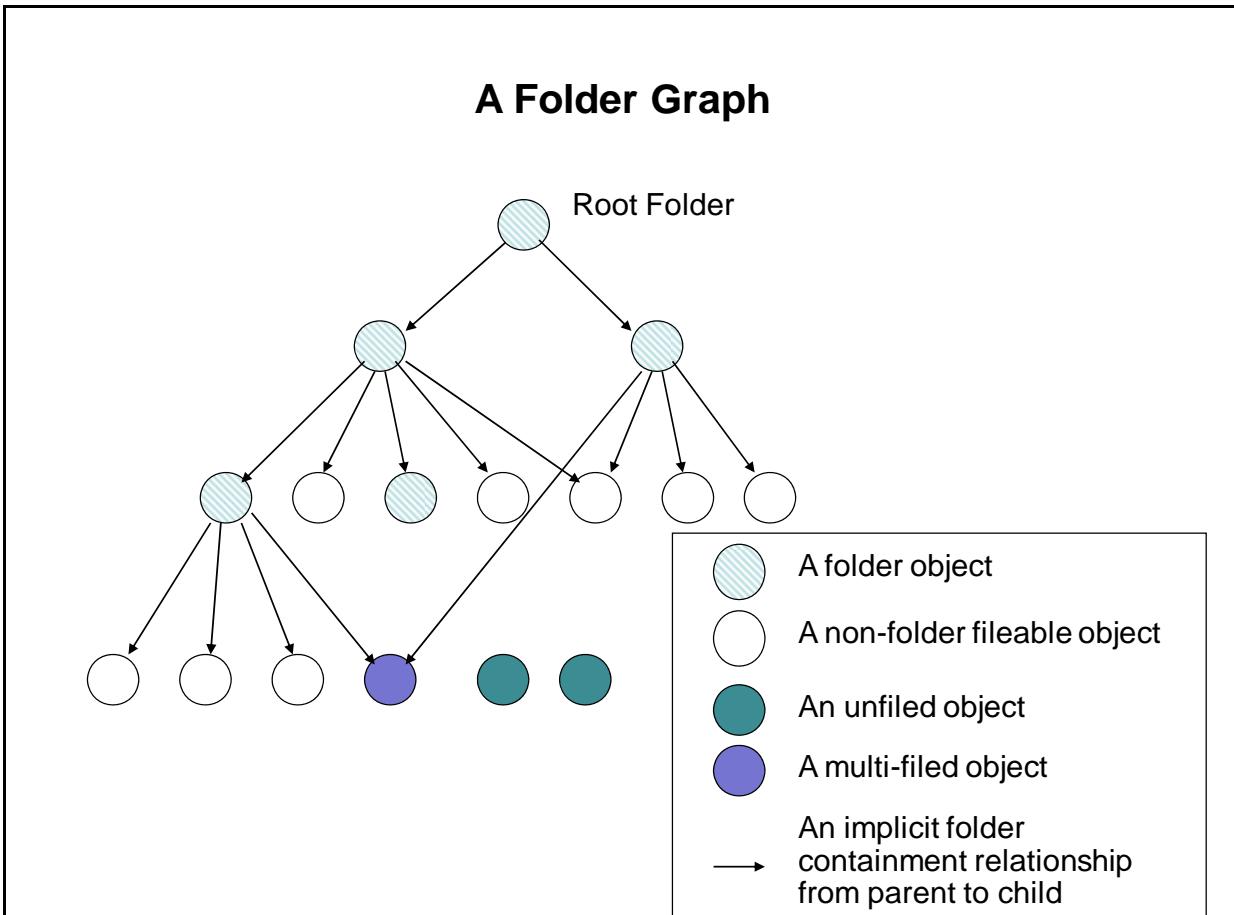
987 CMIS imposes the following constraints on folder objects:

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

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

995 The child objects of a given folder object, their child objects, and grandchild objects, etc., are called
996 **Descendant** objects of the given folder objectA folder object together with all its descendant objects are
997 collectively called a **Tree** rooted at that folder object.

998 A non-folder object does not have any descendant object. Thus, a **Folder Graph** that consists of all
999 fileable objects as nodes, and all the implicit folder containment relationships as directed edges from
1000 parent to child, is a directed acyclic graph, possibly with some disconnected (orphan) nodes. It follows
1001 that the tree rooted at any given folder object is also a directed acyclic graph, although a non-folder object
1002 in the tree MAY have ancestors that are not ancestors of the rooted folder.



1004 Folder objects are handled using [the basic CRUD services for objects](#), and the folder graph is traversed
 1005 using the [Navigation Services](#).

1006 The **Root Folder** is a special folder such that it cannot be created, deleted, or moved using CMIS
 1007 services. Otherwise, it behaves like any other folder object.

1008 **2.1.5.3 Paths**

1009 A folder hierarchy MAY be represented in a canonical notation such as path. For CMIS, a path is
 1010 represented by:

- 1011 • '/' for the root folder
- 1012 • All paths start with the root folder.
- 1013 • A set of the folder and object path segments separated by '/' in order of closest to the root.
- 1014 • Folder and object path segments are specified by pathSegment tokens which can be retrieved by
 1015 all services that take an includePathSegments parameter.
- 1016 • A pathSegment token MUST not include a '/' character.
 - 1017 ○ It is repository specific how a repository chooses the value for pathSegment.
 1018 Repositories might choose to use cmis:name or content stream filename for
 1019 pathSegment token.
- 1020 • The pathSegment token for each item MUST uniquely identify the item in the folder.

1021
 1022 E.g., if folder A is under the root, and folder B is under A, then the path would be /A/B.

1023 A path for an object may be calculated by taking the item's parent folder cmis:path property and
1024 appending the "/" character and the object's pathSegment. This constructed path may be given as input
1025 to the *getObjectByPath* service for object by path retrieval.

1026 The *getObjectParents* service returns relativePathSegment tokens. These tokens are the
1027 pathSegment of the input object relative to the parent folders.

1028 **2.1.5.4 Folder Object-Type Definition**

1029 This section describes the definition of the Folder Object-Type's attribute values and property definitions
1030 which must be present on Folder instance objects. All attributes and property definitions are listed by
1031 their ID.

1032 **2.1.5.4.1 Attribute Values**

1033 The Folder Object-Type MUST have the following attribute values.

1034 Notes:

- 1035 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid
1036 value for the attribute type.
- 1037 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1038 Object-Type definition.

1039

1040 **id**

1041 Value: cmis:folder

1042

1043 **localName**

1044 Value: <repository-specific>

1045

1046 **localNamespace**

1047 Value: <repository-specific>

1048

1049 **queryName**

1050 Value: cmis:folder

1051

1052 **displayName**

1053 Value: <repository-specific>

1054

1055 **baseId**

1056 Value: cmis:folder

1057

1058 **parentId**

1059 Value: Not set

1060

1061 **description**

1062 Value: <repository-specific>

1063

1064 **creatable**

1065 Value: <repository-specific>

```

1066
1067     fileable
1068         Value: TRUE
1069
1070     queryable
1071         Value: SHOULD be TRUE
1072
1073     controllablePolicy
1074         Value: <repository-specific>
1075
1076     includedInSupertypeQuery
1077         Value: <repository-specific>
1078
1079     controllableACL
1080         Value: <repository-specific>
1081
1082     fulltextIndexed
1083         Value: <repository-specific>
1084

```

1085 **2.1.5.4.2 Property Definitions**

1086 The Folder base Object-Type MUST have the following property definitions, and MAY include additional
 1087 property definitions. Any attributes not specified for the Property Definition are repository specific. For all
 1088 property definitions on base types, the query name MUST be the same as the property ID. The
 1089 repository MUST have the following property definitions on the Folder Type:

1090	cmis:name	Name of the object
1091	Inherited:	False
1092	Property Type:	String
1093	Cardinality:	Single
1094	Required:	True
1095		
1096	cmis:objectId	Id of the object
1097	Required:	False
1098	Inherited:	False
1099	Property Type:	ID
1100	Cardinality:	Single
1101	Updatability:	Read Only
1102	Choices:	Not Applicable
1103	Open Choice:	Not Applicable
1104	MUST be set on the object	
1105		
1106	cmis:baseTypeId	Id of the base object-type for the object
1107	Required:	False
1108	Inherited:	False

1109	Property Type:	ID
1110	Cardinality:	Single
1111	Updatability:	Read Only
1112	Choices:	Not Applicable
1113	Open Choice:	Not Applicable
1114	MUST be set on the object	
1115		
1116	cmis:objectTypeId	Id of the object's type
1117	Required:	False
1118	Inherited:	False
1119	Property Type:	ID
1120	Cardinality:	Single
1121	Updatability:	oncreate
1122	Choices:	Not Applicable
1123	Open Choice:	Not Applicable
1124	MUST be set on the object	
1125		
1126	cmis:createdBy	User who created the object.
1127	Required:	False
1128	Inherited:	False
1129	Property Type:	String
1130	Cardinality:	Single
1131	Updatability:	Read Only
1132	Choices:	Not Applicable
1133	Open Choice:	Not Applicable
1134	Queryable:	True
1135	Orderable:	True
1136	MUST be set on the object	
1137		
1138	cmis:creationDate	DateTime when the object was created.
1139	Required:	False
1140	Inherited:	False
1141	Property Type:	DateTime
1142	Cardinality:	Single
1143	Updatability:	Read Only
1144	Choices:	Not Applicable
1145	Open Choice:	Not Applicable
1146	Queryable:	True
1147	Orderable:	True
1148	MUST be set on the object	
1149		
1150	cmis:lastModifiedBy	User who last modified the object.

1151	Required:	False
1152	Inherited:	False
1153	Property Type:	String
1154	Cardinality:	Single
1155	Updatability:	Read Only
1156	Choices:	Not Applicable
1157	Open Choice:	Not Applicable
1158	Queryable:	True
1159	Orderable:	True
1160	MUST be set on the object	
1161		
1162	cmis:lastModificationDate	DateTime when the object was last modified.
1163	Required:	False
1164	Inherited:	False
1165	Property Type:	DateTime
1166	Cardinality:	Single
1167	Updatability:	Read Only
1168	Choices:	Not Applicable
1169	Open Choice:	Not Applicable
1170	Queryable:	True
1171	Orderable:	True
1172	MUST be set on the object	
1173		
1174	cmis:changeToken	Token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1175		
1176	Required:	False
1177	Inherited:	False
1178	Property Type:	String
1179	Cardinality:	Single
1180	Updatability:	Read Only
1181	Choices:	Not Applicable
1182	Open Choice:	Not Applicable
1183	MUST be set on the object	
1184		
1185	cmis:parentId	ID of the parent folder of the folder.
1186	Required:	False
1187	Inherited:	False
1188	Property Type:	ID
1189	Cardinality:	Single
1190	Updatability:	Read Only
1191	Choices:	Not Applicable
1192	Open Choice:	Not Applicable
1193	MUST be set on the object	

1194		
1195	cmis:path	The fully qualified path to this folder. See section 2.1.5.3 Paths.
1196		
1197	Required:	False
1198	Inherited:	False
1199	Property Type:	String
1200	Cardinality:	Single
1201	Updatability:	Read Only
1202	Choices:	Not Applicable
1203	Open Choice:	Not Applicable
1204	MUST be set on the object	
1205		
1206	cmis:allowedChildObjectTypeIds	Id's of the set of Object-types that can be created, moved or filed into this folder.
1207		
1208	Required:	False
1209	Inherited:	False
1210	Property Type:	ID
1211	Cardinality:	Multi
1212	Updatability:	Read Only
1213	Choices:	Not Applicable
1214	Open Choice:	Not Applicable

1215 **2.1.6 Relationship Object**

1216	A relationship object is semantically a <i>dependent</i> object. A relationship object MUST NOT have a content-stream, and MUST NOT be versionable, MUST NOT be queryable, and MUST NOT be fileable, although it MAY be controllable.
1219	If a repository does not support relationship objects, the relationship base object-type SHOULD NOT be returned by a "Get Types" service call.
1221	A Relationship Object instantiates an explicit, binary, directional, non-invasive, and typed relationship between a Source Object and a Target Object . The source object and the target object MUST both be independent objects, such as a document object, a folder object, or a policy object. Whether a policy object is allowed to be the source or target object of a relationship object is repository-specific.
1225	The relationship instantiated by a relationship object is <i>explicit</i> since it is explicitly represented by an object and is explicitly managed by application.
1227	This relationship is <i>non-invasive</i> in the sense that creating or removing this relationship SHOULD NOT modify either the source or the target object. That is, it SHOULD NOT require an update capability (or permission) on either object; SHOULD NOT affect the versioning state of either object; and SHOULD NOT change their "Last Modification Date".
1231	Explicit relationships can be used to create an arbitrary relationship graph among independent objects. Such a relationship graph is only structural in nature. No inheritance or transitive properties are attached to a relationship graph.
1234	The notion of a source object and a target object of a relationship is used solely to indicate the direction of the relationship. No semantics or implementation bias is implied by this terminology.

1236 The binding of a relationship object to a source document object or to a target document object MAY be
1237 either version-specific or version-independent. This version sensitivity is repository-specific, and is largely
1238 transparent to CMIS. An independent object MAY participate in any number of explicit relationships, as
1239 the source object for some and as the target object for others. Multiple relationships MAY exist between
1240 the same pair of source and target objects.

1241 Referential integrity, either between the source object and the target object, or between the relationship
1242 object and the source or target object, is repository-specific. Therefore, creating an explicit relationship
1243 between two objects MAY impose a constraint on any of the three objects, and removing a relationship or
1244 deleting either the source or the target object MAY be restricted by such a constraint. If the source or the
1245 target object of a relationship is deleted, the repository MAY automatically delete the relationship object.

1246 Like all CMIS objects, relationship objects are typed. Typing relationship allows them to be grouped,
1247 identified, and traversed by type id, and for properties to be defined for individual relationship types.

1248 Additionally, a relationship object-type MAY specify that only Objects of a specific Object-Type can
1249 participate as the source object or target object for relationship objects of that type. If no such constraints
1250 are specified, then an independent object of any type MAY be the source or the target of a relationship
1251 object of that type.

1252 When a relationship object is created, the source object ID and the target object ID MUST reference valid
1253 non-relationship CMIS objects.

1254 When a relationship object is retrieved, its source object or target object MAY no longer exist, since
1255 referential integrity MAY not be maintained by a repository.

1256 In addition to object CRUD services, a “Get Relationships” service (*getObjectRelationships*) may be used
1257 to return a set of relationship objects in which a given independent object is identified as the source or the
1258 target object, according to the binding semantics maintained by the repository (i.e., either a version-
1259 specific or a version-independent binding as described above).

1260 **2.1.6.1 Relationship Object-Type Definition**

1261 This section describes the definition of the Relationship Object-Type’s attribute values and property
1262 definitions which must be present on Relationship instance objects. All attributes and property definitions
1263 are listed by their ID.

1264 **2.1.6.1.1 Attributes specific to Relationship Object-Types**

1265 The following Object **attributes** MUST only apply to Object-Type definitions whose baseld is the
1266 cmis:relationship Object-Type, in addition to the common attributes specified above:

1267 **allowedSourceTypes** ID (multi-valued)

1268 A list of object-type IDs, indicating that the source object of a relationship object of this type
1269 MUST only be one of the types listed.

1270 If this attribute is “not set”, then the source object MAY be of any type.

1271

1272 **allowedTargetTypes** ID (multi-valued)

1273 A list of object-type IDs, indicating that the target object of a relationship object of this type MUST
1274 only be one of the types listed.

1275 If this attribute is “not set”, then the target object MAY be of any type.

1276 **2.1.6.1.2 Attribute Values**

1277 The Relationship Object-Type MUST have the following attribute values.

1278 Notes:

- 1279 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid
1280 value for the attribute type.
1281 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1282 Object-Type definition.
- 1283
- 1284 **id**
1285 Value: cmis:relationship
- 1286
- 1287 **localName**
1288 Value: <repository-specific>
- 1289
- 1290 **localNamespace**
1291 Value: <repository-specific>
- 1292
- 1293 **queryName**
1294 Value: cmis:relationship
- 1295
- 1296 **displayName**
1297 Value: <repository-specific>
- 1298
- 1299 **baseId**
1300 Value: cmis:relationship
- 1301
- 1302 **parentId**
1303 Value: Not set
- 1304
- 1305 **description**
1306 Value: <repository-specific>
- 1307
- 1308 **creatable**
1309 Value: <repository-specific>
- 1310
- 1311 **fileable**
1312 Value: FALSE
- 1313
- 1314 **queryable**
1315 Value: <repository-specific>
- 1316
- 1317 **includedInSupertypeQuery**
1318 Value: <repository-specific>
- 1319
- 1320 **controllablePolicy**
1321 Value: <repository-specific>

```

1322
1323     allowedSourceTypes
1324         Value: <repository-specific>
1325
1326     allowedTargetTypes
1327         Value: <repository-specific>
1328
1329     controllableACL
1330         Value: <repository-specific>
1331
1332     fulltextIndexed
1333         Value: <repository-specific>
1334

```

1335 **2.1.6.1.3 Property Definitions**

1336 The Relationship base Object-Type MUST have the following property definitions, and MAY include
 1337 additional property definitions. Any attributes not specified by the Property Definitions are repository
 1338 specific. For all property definitions on base types, the query name MUST be the same as the property
 1339 ID. The repository MUST have the following property definitions on the Relationship Type:

```

1340
1341     cmis:name                               Name of the object
1342         Inherited:                         False
1343         Property Type:                     String
1344         Cardinality:                      Single
1345
1346     cmis:objectId                           Id of the object
1347         Required:                          False
1348         Inherited:                         False
1349         Property Type:                     ID
1350         Cardinality:                      Single
1351         Updatability:                     Read Only
1352         Choices:                           Not Applicable
1353         Open Choice:                      Not Applicable
1354         MUST be set on the object
1355
1356     cmis:baseTypeId                          Id of the base object-type for the object
1357         Required:                          False
1358         Inherited:                         False
1359         Property Type:                     ID
1360         Cardinality:                      Single
1361         Updatability:                     Read Only
1362         Choices:                           Not Applicable
1363         Open Choice:                      Not Applicable
1364         MUST be set on the object

```

1365		
1366	cmis:objectTypeId	Id of the object's type
1367	Required:	False
1368	Inherited:	False
1369	Property Type:	ID
1370	Cardinality:	Single
1371	Updatability:	oncreate
1372	Choices:	Not Applicable
1373	Open Choice:	Not Applicable
1374	MUST be set on the object	
1375		
1376	cmis:createdBy	User who created the object.
1377	Required:	False
1378	Inherited:	False
1379	Property Type:	String
1380	Cardinality:	Single
1381	Updatability:	Read Only
1382	Choices:	Not Applicable
1383	Open Choice:	Not Applicable
1384	MUST be set on the object	
1385		
1386	cmis:creationDate	DateTime when the object was created.
1387	Required:	False
1388	Inherited:	False
1389	Property Type:	DateTime
1390	Cardinality:	Single
1391	Updatability:	Read Only
1392	Choices:	Not Applicable
1393	Open Choice:	Not Applicable
1394	MUST be set on the object	
1395		
1396	cmis:lastModifiedBy	User who last modified the object.
1397	Required:	False
1398	Inherited:	False
1399	Property Type:	String
1400	Cardinality:	Single
1401	Updatability:	Read Only
1402	Choices:	Not Applicable
1403	Open Choice:	Not Applicable
1404	MUST be set on the object	
1405		
1406	cmis:lastModificationDate	DateTime when the object was last modified.

1407	Required:	False
1408	Inherited:	False
1409	Property Type:	DateTime
1410	Cardinality:	Single
1411	Updatability:	Read Only
1412	Choices:	Not Applicable
1413	Open Choice:	Not Applicable
1414	MUST be set on the object	
1415		
1416	cmis:changeToken	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1417		
1418	Required:	False
1419	Inherited:	False
1420	Property Type:	String
1421	Cardinality:	Single
1422	Updatability:	Read Only
1423	Choices:	Not Applicable
1424	Open Choice:	Not Applicable
1425		
1426	cmis:sourcelId	ID of the source object of the relationship.
1427	Required:	True
1428	Inherited:	False
1429	Property Type:	ID
1430	Cardinality:	Single
1431	Choices:	Not Applicable
1432	Open Choice:	Not Applicable
1433		
1434	cmis:targetId	ID of the target object of the relationship.
1435	Required:	True
1436	Inherited:	False
1437	Property Type:	ID
1438	Cardinality:	Single
1439	Choices:	Not Applicable
1440	Open Choice:	Not Applicable

1441 **2.1.7 Policy Object**

1442 A policy object represents an administrative policy that can be enforced by a repository, such as a
 1443 retention management policy. CMIS 1.0 does not specify what kinds of administrative policies that are
 1444 specifically supported, nor attempts to model administrative policy of any particular kind. Only a base
 1445 object-type is specified for policy objects. Each policy object holds the text of an administrative policy as a
 1446 repository-specific string, which is opaque to CMIS and which may be used to support policies of various
 1447 kinds. A repository may create subtypes of this base type to support different kinds of administrative
 1448 policies more specifically. If a repository does not support policy objects, the policy base object-type

1449 SHOULD NOT be returned by a “Get Types” service call. This is an extension point for repositories that
1450 want to expose other capabilities via CMIS that are not supported directly in CMIS 1.0.

1451 Aside from allowing an application to create and maintain policy objects, CMIS allows an application to
1452 “apply” a policy to an object, and to remove an applied policy from an object. An object to which a policy
1453 may be applied is called a *controllable* object. A policy MAY be applied to multiple controllable objects.
1454 Conversely, a repository MAY allow multiple policies applied to a controllable object. (A repository may,
1455 for example, impose constraints such as only one policy of each kind can be applied to an object.)
1456 Whether or not an object is controllable is specified by the object’s type definition. Applying a policy to an
1457 object is to place the object under the control of that policy (while the object may also be under the control
1458 of other policies at the same time), and removing an applied policy from one of its controlled objects is to
1459 remove the corresponding control from that object. This control may change the state of the object, may
1460 impose certain constraints on service calls operating on this object, or may cause certain management
1461 actions to take place. The effect of this control, when this effect takes place, and how this control interacts
1462 with other controls, are repository-specific. Only directly/explicitly applied policies are covered by CMIS
1463 1.0. Indirectly applying policy to an object, e.g. through inheritance, is outside the scope of CMIS 1.0.

1464 A policy object does not have a content-stream and is not versionable. It may be fileable, queryable or
1465 controllable. Policy objects are handled using the basic CRUD services for objects. If a policy is updated,
1466 the change may alter the corresponding control on objects that the policy is currently applied to. If a
1467 controlled object is deleted, all the policies applied to that object, if there are any, are removed from that
1468 object. A policy object that is currently applied to one or more controllable objects CAN NOT be deleted.
1469 That is, there is an implicit referential constraint from a controlled object to its controlling policy object(s).
1470 Besides the basic CRUD services, the “Apply Policy” (*applyPolicy*) and the “Remove Policy”
1471 (*removePolicy*) services may be used to apply a policy object to a controllable object and respectively to
1472 remove an applied policy from one of its controlled objects. In addition, the “Get Applied Policies”
1473 (*getAppliedPolicies*) service may be used to obtain the policy objects that are currently applied to a
1474 controllable object.

1475 **2.1.7.1 Policy Object-Type Definition**

1476 This section describes the definition of the Policy Object-Type’s attribute values and property definitions
1477 which must be present on Policy instance objects. All attributes and property definitions are listed by their
1478 ID.

1479 **2.1.7.1.1 Attribute Values**

1480 The Policy Object-Type MUST have the following attribute values.

1481 Notes:

- 1482 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid
1483 value for the attribute type.
- 1484 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
1485 Object-Type definition.

1486

1487 **id**

1488 Value: cmis:policy

1489

1490 **localName**

1491 Value: <repository-specific>

1492

1493 **localNamespace**

1494 Value: <repository-specific>

1495

```
1496    queryName  
1497        Value: cmis:policy  
1498  
1499    displayName  
1500        Value: <repository-specific>  
1501  
1502    baseId  
1503        Value: cmis:policy  
1504  
1505    parentId  
1506        Value: Not set  
1507  
1508    description  
1509        Value: <repository-specific>  
1510  
1511    creatable  
1512        Value: <repository-specific>  
1513  
1514    fileable  
1515        Value: <repository-specific>  
1516  
1517    queryable  
1518        Value: <repository-specific>  
1519  
1520    includedInSupertypeQuery  
1521        Value: <repository-specific>  
1522  
1523    controllablePolicy  
1524        Value: <repository-specific>  
1525  
1526    controllableACL  
1527        Value: <repository-specific>  
1528  
1529    fulltextIndexed  
1530        Value: <repository-specific>  
1531
```

1532 **2.1.7.1.2 Property Definitions**

1533 The Policy base Object-Type MUST have the following property definitions, and MAY include additional
1534 property definitions. Any attributes not specified by the Property Definitions are repository specific. For
1535 all property definitions on base types, the query name MUST be the same as the property ID. The
1536 repository MUST have the following property definitions on the Policy Type:

```
1537  
1538    cmis:name          Name of the object
```

1539	Inherited:	False
1540	Property Type:	String
1541	Cardinality:	Single
1542		
1543	cmis:objectId	Id of the object
1544	Required:	False
1545	Inherited:	False
1546	Property Type:	ID
1547	Cardinality:	Single
1548	Updatability:	Read Only
1549	Choices:	Not Applicable
1550	Open Choice:	Not Applicable
1551		
1552	cmis:baseTypeId	Id of the base object-type for the object
1553	Required:	False
1554	Inherited:	False
1555	Property Type:	ID
1556	Cardinality:	Single
1557	Updatability:	Read Only
1558	Choices:	Not Applicable
1559	Open Choice:	Not Applicable
1560		
1561	cmis:objectTypeId	Id of the object's type
1562	Required:	False
1563	Inherited:	False
1564	Property Type:	ID
1565	Cardinality:	Single
1566	Updatability:	oncreate
1567	Choices:	Not Applicable
1568	Open Choice:	Not Applicable
1569		
1570	cmis:createdBy	User who created the object.
1571	Required:	False
1572	Inherited:	False
1573	Property Type:	String
1574	Cardinality:	Single
1575	Updatability:	Read Only
1576	Choices:	Not Applicable
1577	Open Choice:	Not Applicable
1578		
1579	cmis:creationDate	DateTime when the object was created.
1580	Required:	False

1581	Inherited:	False
1582	Property Type:	DateTime
1583	Cardinality:	Single
1584	Updatability:	Read Only
1585	Choices:	Not Applicable
1586	Open Choice:	Not Applicable
1587		
1588	cmis:lastModifiedBy	User who last modified the object.
1589	Required:	False
1590	Inherited:	False
1591	Property Type:	String
1592	Cardinality:	Single
1593	Updatability:	Read Only
1594	Choices:	Not Applicable
1595	Open Choice:	Not Applicable
1596		
1597	cmis:lastModificationDate	DateTime when the object was last modified.
1598	Required:	False
1599	Inherited:	False
1600	Property Type:	DateTime
1601	Cardinality:	Single
1602	Updatability:	Read Only
1603	Choices:	Not Applicable
1604	Open Choice:	Not Applicable
1605		
1606	cmis:changeToken	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1607		
1608	Required:	False
1609	Inherited:	False
1610	Property Type:	String
1611	Cardinality:	Single
1612	Updatability:	Read Only
1613	Choices:	Not Applicable
1614	Open Choice:	Not Applicable
1615		
1616	cmis:policyText	User-friendly description of the policy
1617	Required:	True
1618	Inherited:	False
1619	Property Type:	String
1620	Cardinality:	Single
1621	Choices:	Not Applicable
1622	Open Choice:	Not Applicable

1623 **2.1.8 Access Control**

1624 A repository can support either a base set of CMIS-defined permissions and/or its own set of repository
1625 specific permissions.

1626 The getACL service allows the requestor to specify that the result be expressed using only the CMIS
1627 defined permissions. Without this restriction, the response may include, or be solely expressed in
1628 repository specific permissions. The applyACL service permits either CMIS permissions or repository
1629 permissions, or a combination of both, to be used.

1630 **2.1.8.1 ACL, ACE, Principal, and Permission**

1631 An **ACL** is a list of **Access Control Entries** (ACEs) and MAY hold zero or more ACEs. If an ACL has no
1632 ACEs, the behavior is the same as if the ACL is not set.

1633 An **ACE** holds:

- 1634 • one **Principal**: A principal represents a user management object, e.g. a user, group, or role.
1635 It holds one **String** with the **principalid**.
- 1636 • One or more **Strings** with the names of the **permissions**.
- 1637 • a **Boolean** flag **direct**, which indicates if TRUE the ACE is directly assigned to the object. If
1638 FALSE, that the ACE is somehow derived.

1639 **2.1.8.2 CMIS Permissions**

1640 There are three basic permissions predefined by CMIS:

- 1641 • **cmis:read**: to be used to express “permission to read”. A Repository SHOULD express the
1642 permission for reading properties AND reading content with this permission.
- 1643 • **cmis:write**: to be used to express “permission to write”. SHOULD be used to express permission
1644 to write properties and content of an object. MAY include other basic CMIS permissions.
- 1645 • **cmis:all**: SHOULD be used to express all the permissions of a repository. SHOULD include all
1646 other basic CMIS permissions.

1647 How these basic permissions can be mapped to the allowable actions is repository specific. However, the
1648 actual repository semantics for the basic permissions with regard to allowable actions can be discovered
1649 by the *mappings* parameter returned by *getRepositoryInfo* (see below).

1650 Repositories MAY extend this set with repository-specific permissions.

1651 **2.1.8.3 ACL Capabilities**

1652 Whether a repository supports ACLs at all, may be discovered via *capabilityACL* returned by
1653 *getRepositoryInfo* (see section 2.1.1.1 Optional Capabilities). If *capabilityACL* is *none*, ACLs are not
1654 supported by the repository.

1655 If *capabilityACL* is *discover* or *manage*, additional information about the repositories permission model
1656 and how changes to ACL are handled, can be discovered via the *getRepositoryInfo* service:

- 1657 • **<Array> Enum propagation**: specifies, how non-direct ACEs can be handled by the repository
1658 using the following values (see section 2.2.10.2 *applyACL*):
 - 1659 ○ **objectonly** indicates, that the repository is able to apply ACEs to a document or folder,
1660 without changing the ACLs of other objects.
 - 1661 ○ **propagate**: indicates that the ACEs is to be applied to the given object and all inheriting
1662 objects.
 - 1663 ○ **repositorydetermined** indicates, that the repository has its own mechanism of
1664 computing how changing an ACL for an object influences the non-direct ACEs of other
1665 objects.

- 1666 • **<Array> PermissionDefinition repositoryPermissions:** is a list with names and descriptions of
1667 the supported permissions.
1668 • **<Array> PermissionMapping mappings:** contains a list with mappings for the basic CMIS
1669 permissions to allowed actions.

1670 **2.1.8.3.1 Supported Permissions**

1671 The list of permission definitions returned by *getRepositoryInfo* lists all the *permissions* a repository
1672 supports. This list also includes the CMIS permissions if supported by the repository.

1673 A *PermissionDefinition* holds:

- 1674 • **String permission:** the (technical) name of the permission (unique within the list of permission
1675 definitions).
1676 • **(Optional) String description:** an optional description of the permission that should be used as
1677 the permission's name to be presented to the user.

1678 **2.1.8.3.2 AllowableActions & Permission Mapping**

1679 CMIS provides a mechanism called "AllowableActions" which allows an application to discover the set of
1680 service operations that can currently be performed on a particular object, without having to actually invoke
1681 the service.

1682 The set of allowable actions on an object at a point in time are affected not only by CMIS ACLs, but also
1683 by other factors such as:

- 1684 • Constraints inherent in the CMIS Domain Model based on the object's base type or current
1685 versioning state.
1686 • Policies or other control mechanisms that are opaque to CMIS.

1688 CMIS defines several services that applications can use at run-time to discover the AllowableActions for
1689 an object.

1690 If a Repository supports ACLs, then the repository MUST provide a mapping table that defines how the
1691 permissions supported by the repository interact with the CMIS allowable actions, i.e. which permissions
1692 are necessary for a principal to have on one or more objects in order to potentially perform each action,
1693 subject to the other constraints on allowable actions above.

1694 This section defines both the allowable actions as well as how those actions are presented in the
1695 PermissionMapping table.

1696 The Permission Mapping table contains a set of (*key*, *permissions*) pairs:

- 1697 • **String Key:** Because several allowable actions may require permissions on more than one object
1698 – for example, moving a document from one folder to another may require permissions on the
1699 document and each of the folders – the mapping table is defined in terms of permission "keys",
1700 where each key combines the name of the allowable action as the object for which the principal
1701 needs the required permission.
1702 ○ For example – the *canMoveObject.Source* key indicates the permissions that the
1703 principal must have on the "source folder" to move an object from that folder into another
1704 folder.
1705 • **<Array> String permissions:** The names of one or more permissions that the principal MUST
1706 have. If more than one permission is specified, then the principal MUST be allowed to perform the
1707 operation if they have ANY of the listed permissions.

1708 The list below defines all mapping keys, as well as a permissions mapping that repositories SHOULD
1709 use. Repositories MAY require additional permissions.

1710 For convenience, the list below groups all mapping entries by the underlying Allowable Actions, and
1711 includes descriptive information. For each Allowable Action the following information is given:

1712	Description:	The description and name of the service the AllowableAction enables.
1713	Base Object:	The base object-types for which the allowable action MAY be TRUE.
1714	Operand:	The object the permission applies to.
1715	Key:	The permission mapping key.
1716	Permissions:	The permission values.
1717		
1718	Navigation Services:	
1719	canGetDescendants	
1720	Description:	Can get the descendants of the folder (getDescendants)
1721	Base Object:	cmis:folder
1722	Operand:	cmis:folder
1723	Key:	canGetDescendants.Folder
1724	Permission:	Read
1725		
1726	canGetFolderTree	
1727	Description:	Can get the sub-folder tree of the folder (getFolderTree)
1728	Base Object:	cmis:folder
1729	Operand:	cmis:folder
1730	Key:	canGetFolderTree.Folder
1731	Permission:	Read
1732		
1733	canGetChildren	
1734	Description:	Can get the children of the folder (getChildren)
1735	Base Object:	cmis:folder
1736	Operand:	cmis:folder
1737	Key:	canGetChildren.Folder
1738	Permission:	Read
1739		
1740	canGetFolderParent	
1741	Description:	Can get the parent/ancestor folder(s) of the folder (getFolderParent)
1742	Base Object:	cmis:folder
1743	Operand:	cmis:folder
1744	Key:	canGetFolderParent.Folder
1745	Permission:	Read
1746		
1747	canGetObjectParents	
1748	Description:	Can get the parent folders of the object. (getObjectParents)
1749	Base Object:	cmis:document, cmis:folder, cmis:policy
1750	Operand	Object
1751	Key:	canGetObjectParents.Object
1752	Permission:	Read
1753		

1754	Object Services:	
1755	canCreateDocument	
1756	Description:	Can create a cmis:document Object in the folder (createDocument)
1757	Base Object:	cmis:folder
1758	Operand:	Folder
1759	Key:	canCreateDocument.Folder
1760	Permission:	Read
1761		
1762	canCreateFolder	
1763	Description:	Can create a cmis:folder Object as a child of the specified folder
1764	(createFolder)	
1765	Base Object:	cmis:folder
1766	Operand:	Folder
1767	Key:	canCreateFolder.Folder
1768	Permission:	Read
1769		
1770	canCreateRelationship	
1771	Description:	Can create a Relationship in which this Object is a source
1772	(createRelationship)	
1773	Base Object:	cmis:document, cmis:folder
1774	Operand:	Object
1775	Key:	canCreateRelationship.Source
1776	Permission:	Read
1777		
1778	canCreateRelationship	
1779	Description:	Can create a Relationship in which this Object is a target
1780	(createRelationship)	
1781	Base Object:	cmis:document, cmis:folder
1782	Operand:	Object
1783	Key:	canCreateRelationship.Target
1784	Permission:	Read
1785		
1786	canGetProperties	
1787	Description:	Can read the properties of this object (getProperties)
1788	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1789	Operand:	Object
1790	Key:	canGetProperties.Object
1791	Permission:	Read
1792		
1793	canGetRenditions	
1794	Description:	Can retrieve the renditions of this object (getRenditions)
1795	Base Object:	cmis:document
1796	Operand:	Object

1797	Key:	canGetRenditions.Object
1798	Permission:	Read
1799		
1800	canGetContentStream	
1801	Description:	Can get the content stream for the Document object (getContentStream)
1802	Base Object:	cmis:document
1803	Operand:	Object
1804	Key:	canGetContentStream.Object
1805	Permission:	Read
1806		
1807		
1808	canUpdateProperties	
1809	Description:	Can update the properties of this object (updateProperties)
1810	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1811	Operand:	Object
1812	Key:	canUpdateProperties.Object
1813	Permission:	Write
1814		
1815	canMoveObject	
1816	Description:	Can move the object (moveObject)
1817	Base Object:	cmis:document, cmis:folder, cmis:policy
1818	Operand:	Object
1819	Key:	canMoveObject.Object
1820	Permission:	Write
1821		
1822	canMoveObject	
1823	Description:	Can move an object into this folder (moveObject)
1824	Base Object:	cmis:folder
1825	Operand:	Folder
1826	Key:	canMoveObject.Target
1827	Permission:	Read
1828		
1829	canMoveObject	
1830	Description:	Can move an object from this folder (moveObject)
1831	Base Object:	cmis:folder
1832	Operand:	Folder
1833	Key:	canMoveObject.Source
1834	Permission:	Read
1835		
1836	canDeleteObject	
1837	Description:	Can delete this object (deleteObject)
1838	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy

1839 Operand: Object
 1840 Key: canDelete.Object
 1841 Permission: Write
 1842
 1843 **canDeleteObject**
 1844 Description: Can delete an object that is a child of this folder (deleteObject)
 1845 Base Object: cmis:folder
 1846 Operand: Folder
 1847 Key: canDelete.Folder
 1848 Permission: Read
 1849
 1850 **canSetContentStream**
 1851 Description: Can set the content stream for the Document object
 1852 (setContentStream)
 1853 Base Object: cmis:document
 1854 Operand: Object
 1855 Key: canSetContentStream.Document
 1856 Permission: Write
 1857
 1858 **canDeleteContentStream**
 1859 Base Object: cmis:document
 1860 Action: Can delete the content stream for the Document object
 1861 (deleteContentStream)
 1862 Operand: Object
 1863 Key: canDeleteContentStream.Document
 1864 Permission: Write
 1865
 1866 **canDeleteTree**
 1867 Base Object: cmis:folder
 1868 Action: Can delete the folder and all contained objects (deleteTree)
 1869 Operand: Object
 1870 Key: canDeleteTree.Folder
 1871 Permission: Write
 1872
 1873 **Filing Services:**
 1874 **canAddObjectToFolder**
 1875 Description: Can file the document in a folder (addObjectToFolder)
 1876 Base Object: cmis:document, cmis:policy
 1877 Operand: Object
 1878 Key: canAddObjectToFolder.Object
 1879 Permission: Read
 1880
 1881 **canAddObjectToFolder**

1882 Description: Can file a document in the specified folder (`addObjectToFolder`)
1883 Base Object: cmis:document, cmis:policy
1884 Operand: Object
1885 Key: canAddObjectToFolder.Folder
1886 Permission: Read
1887
1888 **canRemoveObjectFromFolder**
1889 Description: Can unfile the specified document from a folder
1890 (`removeObjectFromFolder`)
1891 Base Object: cmis:document, cmis:policy
1892 Operand: Object
1893 Key: canRemoveObjectFromFolder.Object
1894 Permission: Read
1895
1896 **canRemoveObjectFromFolder**
1897 Description: Can unfile a document from the specified folder
1898 (`removeObjectFromFolder`)
1899 Base Object: cmis:document, cmis:policy
1900 Operand: Object
1901 Key: canRemoveObjectFromFolder.Folder
1902 Permission: Read
1903
1904 **Versioning Services:**
1905 **canCheckOut**
1906 Description: Can check out the Document object (`checkOut`)
1907 Base Object: cmis:document
1908 Operand: Object
1909 Key: canCheckOut.Document
1910 Permission: Write
1911
1912 **canCancelCheckOut**
1913 Description: Can cancel the check out the Document object (`cancelCheckOut`)
1914 Base Object: cmis:document
1915 Operand: Object
1916 Key: canCancelCheckOut.Document
1917 Permission: Write
1918
1919 **canCheckIn**
1920 Description: Can check in the Document object (`checkIn`)
1921 Base Object: cmis:document
1922 Operand: Object

1923	Key:	canCheckIn.Document
1924	Permission:	Write
1925		
1926	can GetAllVersions	
1927	Description:	Can get the version series for the Document object (getAllVersions)
1928	Base Object:	cmis:document
1929	Operand:	Object
1930	Key:	canGetAllVersions.Document
1931	Permission:	Read
1932		
1933	Relationship Services:	
1934	canGetObjectRelationships	
1935	Description:	Can get the relationship in which this object is a source/target (getObjectRelationships)
1936	Base Object:	cmis:document, cmis:folder, cmis:policy
1938	Operand:	Object
1939	Key:	canGetObjectRelationships.Object
1940	Permission:	Read
1941		
1942	Policy Services:	
1943	canApplyPolicy	
1944	Description:	Can apply a policy to the Object (applyPolicy)
1945	Base Object:	cmis:document, cmis:folder
1946	Operand:	Object
1947	Key:	canApplyPolicy.Object
1948	Permission:	Read
1949		
1950	canApplyPolicy	
1951	Description:	Can apply the specified policy to an Object (applyPolicy)
1952	Base Object:	cmis:policy
1953	Operand:	Object
1954	Key:	canApplyPolicy.Policy
1955	Permission:	Read
1956		
1957	canRemovePolicy	
1958	Description:	Can remove a policy from the specified Object (removePolicy)
1959	Base Object:	cmis:document, cmis:folder
1960	Operand:	Object
1961	Key:	canRemovePolicy.Object
1962	Permission:	Read
1963		

1964 **canRemovePolicy**

1965 Description: Can remove the specified policy from an Object (`removePolicy`)

1966 Base Object: `cmis:document, cmis:folder`

1967 Operand: `cmis:policy`

1968 Key: `canRemovePolicy.Policy`

1969 Permission: Read

1970

1971 **canGetAppliedPolicies**

1972 Description: Can get the list of Policies applied to the Object
 (`getAppliedPolicies`)

1973 Base Object: `cmis:document, cmis:folder`

1974 Operand: Object

1975 Key: `canGetAppliedPolicies.Object`

1976 Permission: Read

1977

1978

1979 **ACL Services:**

1980 **canGetACL**

1981 Description: Can get ACL for object (`getACL`)

1982 Base Object: `cmis:document, cmis:folder, cmis:relationship, cmis:policy`

1983 Operand: Object

1984 Key: `canGetACL.Object`

1985 Permission: Read

1986

1987 **canApplyACL**

1988 Description: Can apply ACL to this object (`applyACL`)

1989 Base Object: `cmis:document, cmis:folder, cmis:relationship, cmis:policy`

1990 Operand: Object

1991 Key: `canApplyACL.Object`

1992 Permission: Write

1993

1994

1995 2.1.9 Versioning

1996 CMIS supports versioning of Document objects. Folder objects, relationship objects, and policy objects
1997 cannot be versioned.

1998 Whether or not a Document object is versionable (i.e. whether or not operations performed on the object
1999 via the Versioning Services MUST be allowed) is specified by the “*versionable*” attribute on its Object-
2000 type.

2001 A **version** of a Document object is an explicit/“deep” copy of the object, preserving its state at a certain
2002 point in time. Each version of a Document object is itself a Document object, i.e. has its own *ObjectId*,
2003 property values, MAY be acted upon using all CMIS services that act upon Document objects, etc.

2004 2.1.9.1 Version Series

2005 A **version series** for a Document object is a transitively closed collection of all Document objects that
2006 have been created from an original Document in the Repository. Each version series has a unique,
2007 system-assigned, and immutable **version series ID**.

2008 The version series has transitive closure -- that is, if object B is a version of object A, and object C is a
2009 version of object B, then object C is also a version of object A. The objects in a version series can be
2010 conceptually sequenced by their respective *CreationDate* properties.

2011 Additionally, the repository MAY expose a textual **VersionLabel** that describes to a user the position of
2012 an individual object with respect to the version series. (For example, version 1.0).

2013 *Note:* A Document object that is NOT versionable will always have a single object in its Version Series. A
2014 versionable Document object MAY have one or more objects in its Version Series.

2015 2.1.9.2 Latest Version

2016 The version that has the most recent *LastModificationDate* is called the **Latest Version** of the series, or
2017 equivalently, the latest version of any Document object in the series.

2018 When the latest version of a version series is deleted, a previous version (if there is one) becomes the
2019 latest version.

2020 2.1.9.2.1 Behavioral constraints on non-Latest Versions

2021 Repositories NEED NOT allow the non-latest versions in a Version Series to be updated, queried, or
2022 searched.

2023 2.1.9.3 Major Versions

2024 A Document object in a Version Series MAY be designated as a **Major Version**.

2025 The CMIS specification does not define any semantic/behavioral differences between Major and non-
2026 Major versions in a Version Series. Repositories may enforce/apply additional constraints or semantics for
2027 Major versions, if the effect on CMIS services remains consistent with an allowable behavior of the CMIS
2028 model.

2029 If the Version Series contains one or more Major versions, the one that has the most recent
2030 *LastModificationDate* is the **Latest Major Version** of the version series.

2031 (*Note that while a Version Series MUST always have a Latest Version, it NEED NOT have a Latest Major
2032 Version.*)

2033 When the latest major version is deleted, a previous major version (if there is one) becomes the latest
2034 major version.

2035 **2.1.9.4 Services that modify Version Series**

2036 **2.1.9.4.1 Checkout**

2037 A new version of a versionable Document object is created when the *checkIn* service is invoked on the
2038 Private Working copy (PWC) of this object. A PWC is created by invoking *checkOut* on a versionable
2039 Document object. A repository MAY allow *any* Document object in a version series to be checked out, or
2040 MAY only allow the *Latest Version* to be checked out.

2041 The effects of invoking the *checkout* service MUST be as follows:

- 2042 • A new Document object, referred to herein as the **Private Working Copy (PWC)**, is created.
 - 2043 ○ The PWC NEED NOT be visible to users who have permissions to view other Document
2044 objects in the Version Series.
 - 2045 ○ Until it is checked in (using the *checkIn* service), the PWC MUST NOT be considered the
2046 *LatestMajorVersion* in the Version Series.
 - 2047 ○ The property values for the PWC SHOULD be identical to the properties of the Document
2048 object on which the *checkout* service was invoked. Certain properties such as cmis:objectId
2049 may be different. Properties such as cmis:creationDate most likely will be different. The
2050 content-stream of the PWC MAY be identical to the content-stream of the Document object
2051 on which the *checkout* service was invoked, or MAY be “not set”.

2052 After a successful *checkout* operation is completed, and until such time when the PWC is deleted (via the
2053 *cancelCheckOut* service) or checked-in (via the *checkIn*) service, the effects on other Documents in the
2054 Version Series MUST be as follows:

- 2055 • The repository MUST throw an exception if the *checkout* service is invoked on any Document in
2056 the Version Series. (I.e. there can only be one PWC for a version series at a time.)
- 2057 • The value of the cmis:isVersionSeriesCheckedOut property MUST be TRUE.
- 2058 • The value of the cmis:versionSeriesCheckedOutBy property MAY be set to a value indicating
2059 which user created the PWC. (The Repository MAY still show the “not set” value for this
2060 property.)
- 2061 • The value of the cmis:versionSeriesCheckedOutId property MAY be set to the ObjectId of the
2062 PWC. (The Repository MAY still show the “not set” value for this property).
- 2063 • The repository MAY prevent operations that modify or delete the other Documents in the Version
2064 Series.

2065 **2.1.9.4.2 Updates to the Private Working Copy**

2066 If the repository supports the optional “PWCUpdatable” capability, then the repository MUST allow
2067 authorized users to modify the PWC Object using the Object services (e.g. *UpdateProperties*).

2068 If the repository does NOT support the “PWCUpdatable” capability, then the PWC object can only be
2069 modified as part of the *checkIn* service call.

2070 **2.1.9.4.3 Discarding Check out**

2071 An authorized user MAY discard the check-out using the *cancelCheckOut* service on any Document in
2072 the Version Series or by using the *deleteObject* service on the PWC Object. The effects of discarding a
2073 check-out MUST be as follows:

- 2074 • The PWC Object MUST be deleted.
- 2075 • For all other Documents in the Version Series:
 - 2076 ○ The value of the cmis:isVersionSeriesCheckedOut property MUST be FALSE.
 - 2077 ○ The value of the cmis:versionSeriesCheckedOutBy property MUST be “not set”.
 - 2078 ○ The value of the cmis:versionSeriesCheckedOutId property MUST be “not set”.
 - 2079 ○ The repository MUST allow authorized users to invoke the *checkout* service.

2080 **2.1.9.4.4 Checkin**

2081 An authorized user/application MAY “check in” the Private Working Copy object via the *checkIn* service.
2082 The *checkIn* service allows users/applications to provide update property values and a content-stream for
2083 the PWC object.

2084 The effects of the *checkIn* service MUST be as follows for successful checkins:

- 2085 • The PWC object MUST be updated as specified by the inputs to the *checkIn* service. (Note that
2086 for repositories that do NOT support the “PWCUpdatable” property, this is the only way to update
2087 the PWC object.)
- 2088 • The Document object resulting from the *checkIn* operation MUST be considered the *Latest*
2089 *Version* in the Version Series.
- 2090 • If the inputs to the *checkIn* service specified that the PWC MUST be a “major version”, then the
2091 PWC MUST be considered the *Latest Major Version* in the Version Series.
- 2092 • If the checkin returns a new cmis:objected, then the PWC object MUST disappear if the *checkIn*
2093 call was successful and the new checked in version will use the new specified id.
- 2094 • For all Documents in the Version Series:
 - 2095 o The value of the cmis:isVersionSeriesCheckedOut property MUST be FALSE.
 - 2096 o The value of the cmis:versionSeriesCheckedOutBy property MUST be “not set”.
 - 2097 o The value of the cmis:versionSeriesCheckedOutId property MUST be “not set”.
 - 2098 o The repository MUST allow authorized users to invoke the *checkout* service.

2099 *Note:* The Repository MAY change the ID of the PWC upon completion of the *checkin* service invocation.

2100 *Note:* A repository MAY automatically create new versions of Document objects without an explicit
2101 invocation of the *checkout/checkin* services.

2102 **2.1.9.5 Versioning Properties on Document Objects**

2103 All Document objects will have the following read-only property values pertaining to versioning:

- 2104
- 2105 **cmis:isLatestVersion** Boolean
2106 TRUE if the Document object is the *Latest Version* in its *Version Series*. FALSE otherwise.
- 2107
- 2108 **cmis:isMajorVersion** Boolean
2109 TRUE if the Document object is a *Major Version* in its *Version Series*. FALSE otherwise.
- 2110
- 2111 **cmis:isLatestMajorVersion** Boolean
2112 TRUE if the Document object is the *Latest Major Version* in its *Version Series*. FALSE otherwise.
- 2113
- 2114 **cmis:versionLabel** String (optional)
2115 Optional textual description the position of an individual object with respect to the version series.
2116 (For example, version 1.0).
- 2117
- 2118 **cmis:versionSeriesID** ID
2119 ID of the Version Series for this Object.
- 2120
- 2121 **cmis:isVersionSeriesCheckedOut** Boolean
2122 TRUE if there currently exists a Private Working Copy for this Version Series. FALSE otherwise

2123
2124 **cmis:versionSeriesCheckedOutBy** String
2125 If IsVersionSeriesCheckedOut is TRUE: then an identifier for the user who created the Private
2126 Working Copy. "Not set" otherwise.
2127
2128 **cmis:versionSeriesCheckedOutId** ID
2129 If IsVersionSeriesCheckedOut is TRUE: The Identifier for the Private Working Copy. "Not set"
2130 otherwise.
2131
2132 **cmis:checkinComment** String
2133 Textual comment associated with the given version.
2134 Note: Changes made via the Versioning Services that affect the values of these properties MUST NOT
2135 constitute modifications to the Document objects in the Version Series (e.g. MUST NOT affect the
2136 cmis:lastModificationDate, etc.)

2137 **2.1.9.6 Document Creation and Initial Versioning State**

2138 A repository MAY create new Document objects in a "Private Working Copy" state when they are created
2139 via the *createDocument* or *createDocumentFromSource* services. This state is logically equivalent to
2140 having a Version Series that contains exactly one object (the PWC) and 0 other documents.
2141 The repository MAY also create new Document objects in a "Major Version" state. This state is logically
2142 equivalent to having a Version Series that contains exactly one Major Version and 0 other documents.
2143 The repository MAY also create new Document objects in a "Non-Major Version" state. This state is
2144 logically equivalent to having a Version Series that contains exactly one Non-Major Version and 0 other
2145 documents.
2146 If the repository does not support versioning the repository MAY create new Document objects in a "Major
2147 Version" state.

2148 **2.1.9.7 Version Specific/Independent membership in Folders**

2149 Repositories MAY treat membership of a Document object in a folder collection as "version-specific" or
2150 "version-independent".

2151 Repositories MUST indicate whether they support version-specific membership in a folder via the
2152 "VersionSpecificFiling" optional capability flag.

2153 If the repository is treating folder collection membership as "version-independent", then:

- 2154 • Moving or Filing a Document Object into a folder MUST result in ALL Documents in the Version
2155 Series being moved/filed into the folder.
- 2156 • The Repository MAY return only the latest-version OR latest major-version Document object in a
2157 version series in the response to Navigation service requests (getChildren, getDescendants), and
2158 NEED NOT return other Document Objects filed in the folder that are in the Version Series.

2159 If the repository is treating folder collection membership as "version-specific", then moving or Filing a
2160 Document Object into a folder MUST NOT result in other Documents in the Version Series being
2161 moved/filed.

2162 **2.1.9.8 Version Specific/Independent membership in Relationships**

2163 A relationship object MAY have either a version-specific or version-independent binding to its source
2164 and/or target objects. This behavior MAY vary between repositories and between individual relationship
2165 types defined for a Repository.

2166 If a relationship object has a version-independent binding to its source/target object, then:

- 2167 • The getObjectRelationships service invoked on a Document Object MUST return the relationship
2168 if Relationship was source/target is set to ANY Document Object in the Version Series.

2169 If a relationship object has a version-specific binding to its source/target object, then:

- 2170 • The getObjectRelationships service invoked on a Document Object MUST return the relationship
2171 if Relationship was source/target is set to the ID of the Document Object on which the service was
2172 invoked.

2173 **2.1.9.9 Versioning visibility in Query Services**

2174 Repositories MAY include non-latest-versions of Document Objects in results to the Discovery Services
2175 (*query*).

2176 Repositories MUST indicate whether they support querying for non-latest-versions via the
2177 “AllVersionsSearchable” optional capability flag.

2178 If “AllVersionsSearchable” is TRUE then the Repository MUST include in the query results ANY
2179 Document Object in the Version Series that matches the query criteria. (subject to other query constraints
2180 such as security.)

2181 Additionally, repositories MAY include Private Working Copy objects in results in results to the Discovery
2182 Services (*query*).

2183 Repositories MUST indicate whether they support querying for Private Working Copy objects via the
2184 “PWCSearchable” optional capability flag.

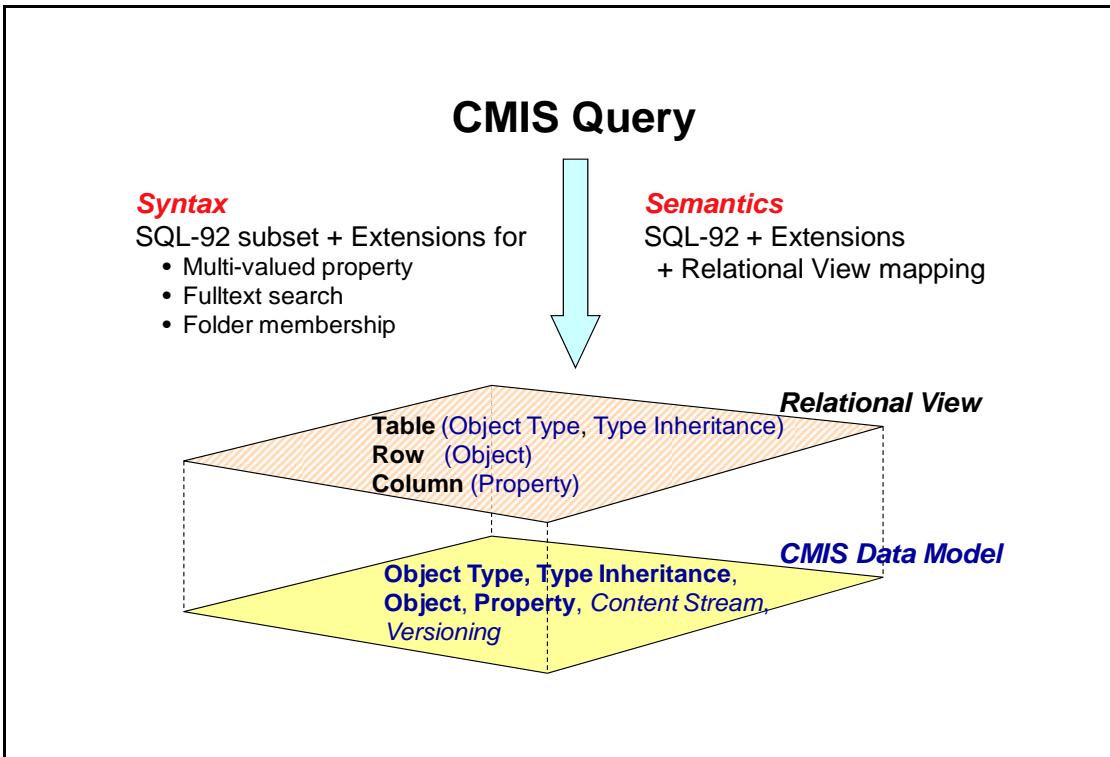
2185 If “PWCSearchable” is TRUE then the Repository MUST include in the query results ANY Private Working
2186 Copy Document Objects that matches the query criteria (subject to other query constraints such as
2187 security.)

2188 If “PWCSearchable” is FALSE then the Repository MUST NOT include in the query results ANY Private
2189 Working Copy Document Objects that match the query criteria (subject to other query constraints such as
2190 security.)

2191 **2.1.10 Query**

2192 CMIS provides a type-based query service for discovering objects that match specified criteria, by
2193 defining a read-only projection of the CMIS data model into a *Relational View*.

2194 Through this relational view, queries may be performed via a simplified SQL SELECT statement. This
2195 query language is based on a subset of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database
2196 Language SQL), with a few extensions to enhance its filtering capability for the CMIS data model, such as
2197 existential quantification for multi-valued property, full-text search, and folder membership. Other
2198 statements of the SQL language are not adopted by CMIS. The semantics of this query language is
2199 defined by the SQL-92 standard, plus the extensions, in conjunction with the model mapping defined by
2200 CMIS’s relational view.



2201

2.1.10.1 Relational View Projection of the CMIS Data Model

2202 The relational view of a CMIS repository consists of a collection of virtual tables that are defined on top of
 2203 the CMIS data model. This relational view is used for query purposes only.

2204 In this relational view a **Virtual Table** is implicitly defined for each *queryable* Object-Type defined in the
 2205 repository. (Non-queryable Object-Types are NOT exposed through this Relational View.)

2206 In each **Virtual Table**, a **Virtual Column** is implicitly defined for each property defined in the Object-Type
 2207 Definition AND for all properties defined on ANY ancestor-type of the Object-Type but NOT defined in the
 2208 Object-Type definition. Virtual Columns for properties defined on ancestor-types of the Object-type but
 2209 NOT defined in the Object-Type definition MUST contain the SQL NULL value. Virtual Columns for
 2210 properties whose value is "not set" MUST contain the SQL NULL value.
 2211

2212 An object-type's *queryName* attribute is used as the table name for the corresponding virtual table, and a
 2213 property's *queryName* attribute is used as the column name for the corresponding table column. Please
 2214 see the restrictions on *queryName* in the appropriate data model section.

2215 The Virtual Column for a multi-valued property MUST contain a single list value that includes all values of
 2216 the property.

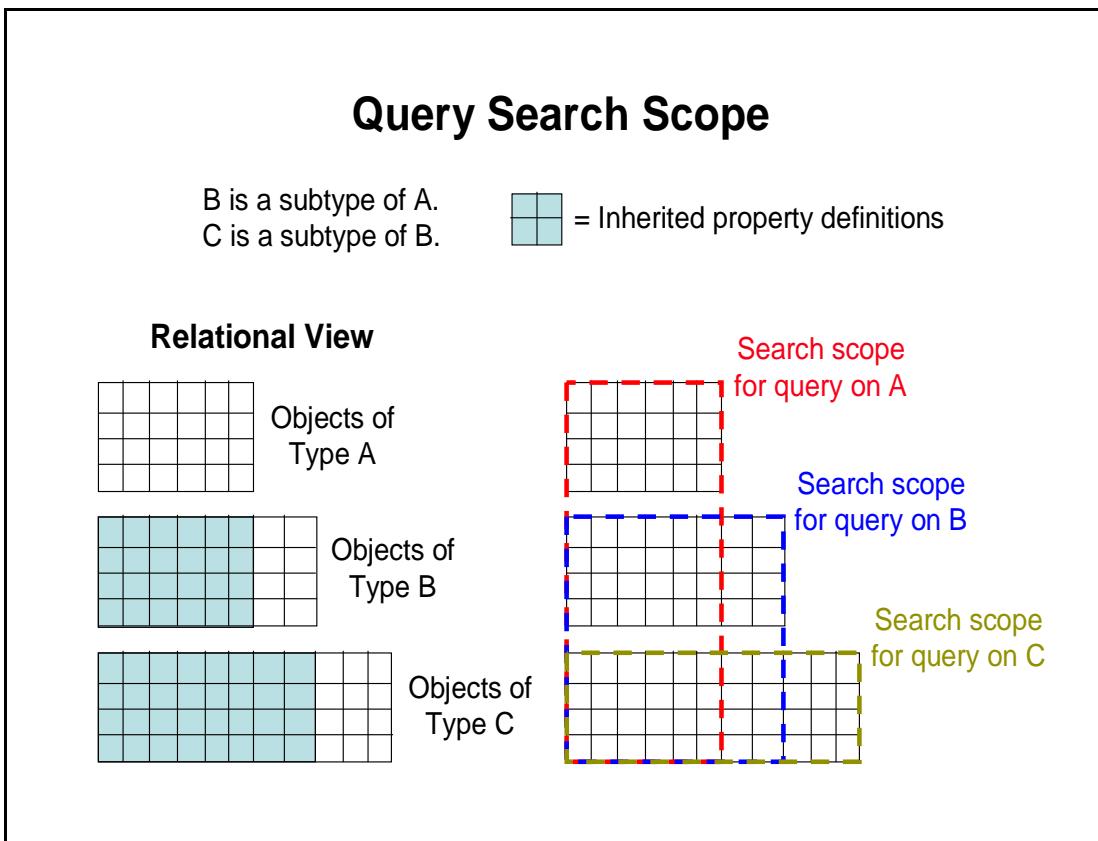
2.1.10.1.1 Object-Type Hierarchy in the Relational View Projection

2217 The Relational View projection of the CMIS Data Model ensures that the Virtual Table for a particular
 2218 Object-type is a complete super-set of the Virtual Table for any and all of its ancestor types.

2219 Additionally, an Object-Type definition's "*includedInSupertypeQuery*" specifies whether objects of that
 2220 Object-Type MUST be included in the Virtual Table for any of its ancestor types. If the
 2221 "*includedInSupertypeQuery*" attribute of the Object-Type is FALSE, then objects of that Object-Type
 2222 MUST NOT be included in the Virtual Table for any of its ancestor types.
 2223

2224 Thus the Virtual Table for an Object-type includes a row not only for each Object of that type, but all
2225 Objects of any of that Object-types' Descendant Types for which the "includedInSupertypeQuery"
2226 attribute is TRUE.

2227 But since the Virtual Table will include only columns for properties defined in the Object-Type underlying
2228 the Virtual Table, a row that is a query result representing an Object of a Descendant Type can only
2229 include those columns for properties defined on the Object-Type underlying the Virtual Table.



2230

2.1.10.1.2 Content Streams

2232 Content-streams are NOT exposed through this relational view.

2.1.10.1.3 Result Set

2234 When a query is submitted, a set of pseudo CMIS objects will be returned. These pseudo objects are
2235 comprised of the properties specified in the select clause of the query statement.

2236 For each property in each object in the result set, the Repository MUST include the property definition ID
2237 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
2238 used).

2239 If the select clause of the query statement contains properties from a single type reference then the
2240 repository MAY represent these pseudo-objects with additional object information.

2.1.10.2 Query Language Definition

2242 This query language is based on a subset of the SQL-92 grammar. CMIS-specific language extensions
2243 to SQL-92 are called out explicitly.

2244 The basic structure of a CMIS query is a SQL statement that MUST include the following clauses:

2245 • **SELECT [virtual columns]:** This clause identifies the set of virtual columns that will be included
2246 in the query results for each row.

2247 • **FROM [Virtual Table Names]:** This clause identifies which Virtual Table(s) the query will run
2248 against.

2249 Additionally, a CMIS query MAY include the following clauses:

2250 • **WHERE [conditions]:** This clause identifies the constraints that rows MUST satisfy to be
2251 considered a result for the query.

2252 • **ORDER BY [sort specification]:** This clause identifies the order in which the result rows MUST
2253 be sorted in the result row set.

2254 **2.1.10.2.1 BNF Grammar**

2255 This BNF grammar is a “subset” of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language
2256 SQL), except for some production alternatives. Specifically, except for these extensions, the following
2257 production rules are derived from the SQL-92 grammar. The non-terminals used in this grammar are also
2258 borrowed from the SQL-92 grammar without altering their semantics. Accordingly, the non-terminal
2259 <column name> is used for single-valued properties only so that the semantics of SQL can be preserved
2260 and borrowed. This approach not only facilitates comparison of the two query languages, and simplifies
2261 the translation of a CMIS query to a SQL query for a RDBMS-based implementation, but also allows
2262 future expansion of this query language to cover a larger subset of SQL with minimum conflict. The CMIS
2263 extensions are introduced primarily to support multi-valued properties and full-text search, and to test
2264 folder membership. Multi-valued properties are handled separately from single-valued properties, using
2265 separate non-terminals and separate production rules to prevent the extensions from corrupting SQL-92
2266 semantics.
2267

```

2268
2269 <CMIS 1.0 query statement> ::= <simple table> [ <order by clause> ]
2270 <simple table> ::= SELECT <select list> <from clause> [ <where clause> ]
2271 <select list> ::= "*"
2272     | <select sublist> [ { "," <select sublist> }... ]
2273 <select sublist> ::= <value expression> [ [ AS ] <column name> ]
2274     | <qualifier> "."
2275     | <multi-valued-column reference>
2276 <value expression> ::= <column reference> | <numeric value function>
2277 <column reference> ::= [ <qualifier> "."] <column name>
2278 <multi-valued-column reference> ::= [ <qualifier> "."] <multi-valued-column name>
2279 <numeric value function> ::= SCORE()
2280 <qualifier> ::= <table name> | <correlation name>
2281 <from clause> ::= FROM <table reference>
2282 <table reference> ::= <table name> [ [ AS ] <correlation name> ]
2283     | <joined table>
2284 <joined table> ::= "(" <joined table> ")"
2285     | <table reference> [ <join type> ] JOIN <table reference> <join specification>
2286 <join type> ::= INNER | LEFT [ OUTER ]
2287 <join specification> ::= ON <column reference> "=" <column reference>
2288 <where clause> ::= WHERE <search condition>
2289 <search condition> ::= <boolean term> | <search condition> OR <boolean term>
2290 <boolean term> ::= <boolean factor> | <boolean term> AND <boolean factor>
2291 <boolean factor> ::= [ NOT ] <boolean test>
2292 <boolean test> ::= <predicate> | "(" <search condition> ")"
2293 <predicate> ::= <comparison predicate> | <in predicate> | <like predicate> | <>null predicate>
2294     | <quantified comparison predicate> | <quantified in predicate>
2295     | <text search predicate> | <folder predicate>
2296 <comparison predicate> ::= <value expression> <comp op> <literal>
2297 <comp op> ::= "=" | "<" | ">" | "<=" | ">="
2298 <literal> ::= <signed numeric literal> | <character string literal> | <datetime literal> | <boolean literal>
2299 <in predicate> ::= <column reference> [ NOT ] IN "(" <in value list> ")"
2300 <in value list> ::= <literal> [ { "," <literal> }... ]
2301 <like predicate> ::= <column reference> [ NOT ] LIKE <character string literal>
2302 <null predicate> ::= { <column reference> | <multi-valued-column reference> } IS [ NOT ] NULL
2303 <quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>
2304 <quantified in predicate> ::= ANY <multi-valued-column reference> [ NOT ] IN "(" <in value list> ")"
2305 <text search predicate> ::= CONTAINS "("
2306     [ <qualifier> "," ] <quote> <text search expression> <quote> ")"
2307 <folder predicate> ::= { IN_FOLDER | IN_TREE } "(" [ <qualifier> "," ] <folder id> ")"
2308 <order by clause> ::= ORDER BY <sort specification> [ { "," <sort specification> }... ]
2309 <sort specification> ::= <column reference> [ ASC | DESC ]
2310 <correlation name> ::= <identifier>

```

```

2311 <table name> ::= <identifier> !! This MUST be the name of an object-type.
2312 <column name> ::= <identifier> !! This MUST be the name of a single-valued property,
2313                               or an alias for a scalar output value.
2314 <multi-valued-column name> ::= <identifier> !! This MUST be the name of a multi-valued property.
2315 <folder id> ::= <character string literal> !! This MUST be the object identity of a folder object.
2316 <identifier> ::= !! As defined by queryName attribute.
2317 <signed numeric literal> ::= !! As defined by SQL-92 grammar.
2318 <character string literal> ::= !! As defined by SQL-92 grammar.
2319
2320 !! This is full-text search criteria.
2321 <text search expression> ::= <disjunct> {<space> OR <space> <disjunct>}
2322 <disjunct> ::= <term> {<space> <term>}
2323 <term> ::= ['-'] <simple term>
2324 <simple term> ::= <word> | <phrase>
2325 <word> ::= <non space char> {<non space char>}
2326 <phrase> ::= <quote> <word> {<space> <word>} <quote>
2327 <space> ::= <space char> {<space char>}
2328 <non space char> ::= <char> - <space char>
2329 <space char> ::= ''
2330 <char> ::= !! Any character
2331
2332 <datetime literal> ::= TIMESTAMP <quote> <datetime string> <quote>
2333 <datetime string> ::= YYYY-MM-DDThh:mm:ss.sss[Z | +hh:mm | -hh:mm]
2334 <boolean literal> ::= TRUE | FALSE | true | false
2335 <quote> ::= "" | """

```

2.1.10.2.2 SELECT Clause

2336 The SELECT clause MUST contain exactly one of the following:

- 2337 • A comma separated list of one or more column names.
 - 2338 o If an explicit column list is provided: A repository MUST include in its result row set all of the columns specified in the SELECT clause.
- 2339 • * : If this token is specified, then the repository MUST return columns for ALL single-valued properties defined in the Object-Types whose Virtual Tables are listed in the FROM clause, and SHOULD also return all multi-valued properties.

2340 All column names MUST be valid “queryName” values for properties that are defined as “queryable” in the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

2.1.10.2.3 FROM Clause

2341 The FROM clause identifies which Virtual Table(s) the query will be run against, as described in the previous section.

2342 The FROM clause MUST contain only the queryNames of Object-Types whose *queryable* attribute value is TRUE.

- 2351 **2.1.10.2.3.1 Join Support**
- 2352 CMIS repositories MAY support the use of SQL JOIN queries, and MUST indicate their support level
 2353 using the [Optional Capability](#) attribute "capabilityJoin".
- 2354 • If the Repository's value for the capabilityJoin attribute is **none**, then no JOIN clauses can be
 2355 used in queries.
 - 2356 • If the Repository's value for the capabilityJoin attribute is **inneronly**, then only inner JOIN clauses
 2357 can be used in queries.
 - 2358 • If the Repository's value for the capabilityJoin attribute is **innerandouter**, then inner and/or outer
 2359 JOIN clauses can be used in queries.
- 2360 Only explicit joins using the "JOIN" keyword is supported. Queries MUST NOT include implicit joins as
 2361 part of the WHERE clause of a CMIS query.
- 2362 CMIS queries MUST only support join operations using the "equality" predicate on single-valued
 2363 properties.
- 2364 **2.1.10.2.4 WHERE Clause**
- 2365 This clause identifies the constraints that rows MUST satisfy to be considered a result for the query.
- 2366 All column names MUST be valid "queryName" or their aliased values for properties that are defined as
 2367 "queryable" in the Object-Type(s) whose Virtual Tables are listed in the FROM clause.
- 2368 Properties are defined to not support a "null" value, therefore the <null predicate> MUST be interpreted as
 2369 testing the not set or set state of the specified property.
- 2370 **2.1.10.2.4.1 Comparisons permitted in the WHERE clause.**
- 2371 SQL's simple comparison predicate, IN predicate, and LIKE predicate are supported, for single-valued
 2372 properties only (so that SQL's semantics is preserved). Boolean conjunction (AND), disjunction (OR), and
 2373 negation (NOT) of predicates are also supported.
- 2374 Repositories SHOULD support the comparisons for the property types as described in the list below.
 2375 Repositories MAY support additional comparisons and operators. Any additional operators not specified
 2376 are repository-specific:
- 2377
- 2378 <Property Type>
- 2379 Supported Operators: <List of Operators supported on Type>
- 2380 Supported Literal: <Supported type of Literal in comparison>
- 2381
- 2382 String (Single)
- 2383 Supported Operators: =, <>, [NOT] LIKE
- 2384 Supported Literal: String
- 2385
- 2386 String (IN)
- 2387 Supported Operators: [NOT] IN
- 2388 Supported Literal: List of Strings
- 2389
- 2390 Decimal
- 2391 Supported Operators: =, <>, <, <=, >, >=
- 2392 Supported Literal: Decimal
- 2393

2394 Decimal (IN)
2395 Supported Operators: [NOT] IN
2396 Supported Literal: List of Decimal
2397
2398 Integer
2399 Supported Operators: =, <>, <, <=, >, >=
2400 Supported Literal: Integer
2401
2402 Integer (IN)
2403 Supported Operators: [NOT] IN
2404 Supported Literal: List of Integer
2405
2406 Boolean
2407 Supported Operators: =
2408 Supported Literal: <boolean literal>
2409
2410 DateTime
2411 Supported Operators: =, <>, <*, <=*, >*, >=*

2412 Supported Literal: <datetime literal>
2413 * - comparison is based on chronological before or after date.
2414
2415 DateTime (IN)
2416 Supported Operators: [NOT] IN
2417 Supported Literal: List of <datetime literal>'s
2418
2419 ID
2420 Supported Operators: =, <>
2421 Supported Literal: String
2422
2423 ID (IN)
2424 Supported Operators: [NOT] IN
2425 Supported Literal: List of strings
2426
2427 URI
2428 Supported Operators: =, <>
2429 Supported Literal: String
2430
2431 URI (IN)
2432 Supported Operators: [NOT] IN
2433 Supported Literal: List of strings
2434
2435 URI

2436 Supported Operators: [NOT] LIKE
2437 Supported Literal: String
2438
2439 Operations on the SCORE() output MUST be treated the same as decimal operations.
2440
2441 When using properties in a join statement, comparison MUST be allowed on properties of the same types
2442 as defined by the table above. Repositories MAY extend this behavior.
2443
2444 The ANY operation argument MUST be one of the properties found in the table above which supports
2445 equality operations

2446 **2.1.10.2.4.2 Multi-valued property support (SQL-92 Extension)**

2447 The CMIS query language includes several new non-terminals to expose semantics for querying multi-
2448 valued properties, in a way that does not alter the semantics of existing SQL-92 production rules.

2449 **2.1.10.2.4.2.1 Multi-valued column references**

2450 **BNF grammar structure:** <Multi-valued-column reference>, <multi-valued-column name>

2451 • These are non-terminals defined for multi-valued properties whereas SQL-92's <column
2452 reference> and <column name> are retained for single-valued properties only. This is to preserve
2453 the single-value semantics of a regular "column" in the SQL-92 grammar.

2454 **2.1.10.2.4.2.2 <Quantified comparison predicate>**

2455 The SQL-92 production rule for <quantified comparison predicate> is extended to accept a multi-valued
2456 property in place of a <table subquery>. This operation is restricted to equality tests only.
2457
2458 <Table subquery> is not supported in CMIS-SQL.
2459
2460 The SQL-92 <quantifier> is restricted to ANY only.
2461
2462 The SQL-92 <row value constructor> is restricted to a literal only.

2463 **Example:**
2464 SELECT Y.CLAIM_NUM, X.PROPERTY_ADDRESS, Y.DAMAGE_ESTIMATES
2465 FROM POLICY AS X JOIN CLAIMS AS Y ON (X.POLICY_NUM = Y.POLICY_NUM)
2466 WHERE (100000 = ANY Y.DAMAGE_ESTIMATES)
2467
2468 *(Note: DAMAGE_ESTIMATES is a multi-valued Integer property.)*

2469 **2.1.10.2.4.2.3 IN/ANY Predicate**

2470 **BNF grammar structure:** <Quantified in predicate>

2471 CMIS-SQL exposes a new IN predicate defined for a multi-valued property. It is modeled after the SQL-
2472 92 IN predicate, but since the entire predicate is different semantically, it has its own production rule in
2473 the BNF grammar below.
2474
2475 The quantifier is restricted to ANY. The predicate MUST be evaluated to TRUE if at least one of the
2476 property's values is (or, is not, if NOT is specified) among the given list of literal values. Otherwise the
2477 predicate is evaluated to FALSE.
2478

2479 The ANY operation argument MUST be one of the properties found in the comparison list above which
2480 supports IN operations.

2481 **Example:**

```
2482     SELECT      *
2483     FROM  CAR REVIEW
2484     WHERE      (MAKE = "buick") OR
2485             ( ANY FEATURES IN ("NAVIGATION SYSTEM", "SATELLITE RADIO", "MP3") )
```

2486 *(Note: FEATURES is a multi-valued String property.)*

2487 **2.1.10.2.4.3 CONTAINS() predicate function (CMIS-SQL Extension)**

2488 **BNF grammar structure::** CONTAINS ([<qualifier> ,] <text search expression>)

2489 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
2490 by a Repository ([See previous section.](#))

2491 **Inputs:**

2492 **<Qualifier>**

2493 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2494 FROM clause for the query.

- 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.

2499 **<Text Search Expression>**

2500 The <text search expression> parameter MUST be a character string literal in quotes, specifying
2501 the full-text search criteria.

2503 The Text Search Expression may be a set of terms or phrases with an optional '-' to signal
2504 negation. A phrase is defined as a word or group of words. A group of words must be
2505 surrounded by quotes to be considered a single phrase.

2507 Terms separated by whitespace are AND'ed together.

2508 Terms separated by "OR" are OR'ed together

2509 Implicit "AND" has higher precedence than "OR"

2510 Within a word or phrase, each double quote must also be escaped by a preceding backslash "\\"

2511 **Return value:**

2512 The predicate returns a Boolean value.

2513 The predicate MUST return TRUE if the object is considered by the repository as "relevant" with
2514 respect to the given <text search expression> parameter.

2515 The predicate MUST return FALSE if the object is considered by the repository as not "relevant"
2516 with respect to the given <text search expression> parameter.

2517 **Constraints:**

2518 At most one CONTAINS() function MUST be included in a single query statement. The repository
2519 MUST throw an exception if more than one CONTAINS() function is found.

2520 The return value of the CONTAINS() function MAY only be included conjunctively (ANDed) with the
2521 aggregate of all other predicates, if there is any, in the WHERE clause.

2523 **2.1.10.2.4.4 SCORE() predicate function**

2524 **BNF grammar structure:** SCORE ()

2525 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
2526 by a Repository ([See previous section.](#))

2527 **Inputs:** No inputs MUST be provided for this predicate function.

2528 **Return value:**

2529 The SCORE() predicate function returns a decimal value in the interval [0,1] .

2530 A repository MUST return the value 0 if the object is considered by the repository as having
2531 absolutely no relevance with respect to the CONTAINS() function specified in the query.

2532 A repository MUST return the value 1 if the object is considered by the repository as having
2533 absolutely complete relevance with respect to the CONTAINS() function specified in the query.

2534 **Constraints:**

2535 The SCORE() function MUST only be used in queries that also include a CONTAINS() predicate
2536 function

2537 The SCORE() function MUST only be used in the SELECT clause of a query. It MUST NOT be
2538 used in the WHERE clause or in the ORDER BY clauses.

2539 An alias column name defined for the SCORE() function call in the SELECT clause (i.e.,
2540 "SELECT SCORE() AS column_name ...") may be used in the ORDER BY clause.

2541 If SCORE() is included in the SELECT clause and an alias column name is not provided, then a
2542 query name of SEARCH_SCORE is used for the query output, and the property definition ID is
2543 repository-specific.

2544 **2.1.10.2.4.5 IN_FOLDER() predicate function**

2545 **BNF grammar structure:** IN_FOLDER([<qualifier>,] <folder id>)

2546 **Usage:** This is a predicate function that tests whether or not a candidate object is a child-object of the
2547 folder object identified by the given <folder id>.

2548 **Inputs:**

2549 **<qualifier>**

2550 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2551 FROM clause for the query.

- 2552 • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
2553 Table, but a repository MAY ignore the value of the parameter.
- 2554 • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
2555 throw an exception if the qualifier is not specified.

2556 **<folder id>**

2557 The value of this parameter MUST be the ID of a folder object in the repository.

2558 **Return value:**

2559 The predicate function MUST return TRUE if the object is a child-object of the folder specified by
2560 <folder id>.

2561 The predicate function MUST return FALSE if the object is a NOT a child-object of the folder
2562 specified by <folder id>.

2563 **2.1.10.2.4.6 IN_TREE() predicate function**

2564 **BNF grammar structure:** IN_TREE([<qualifier>,] <folder id>)

2565 **Usage:** This is a predicate function that tests whether or not a candidate object is a descendant-object of
2566 the folder object identified by the given <folder id>.

2567 **Inputs:**

2568 **<qualifier>**

2569 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2570 FROM clause for the query.

- 2571 • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
2572 Table, but a repository MAY ignore the value of the parameter.
2573 • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
2574 throw an exception if the qualifier is not specified.

2575 **<folder id>**

2576 The value of this parameter MUST be the ID of a folder object in the repository.

2577 **Return value:**

2578 The predicate function MUST return TRUE if the object is a descendant-object of the folder
2579 specified by <folder id>.

2580 The predicate function MUST return FALSE if the object is a NOT a descendant -object of the
2581 folder specified by <folder id>.

2582 **2.1.10.2.5 ORDER BY Clause**

2583 This clause MUST contain a comma separated list of one or more column names.

2584 All column names referenced in this clause MUST be valid “queryName” or their aliased values for
2585 properties defined as *orderable* in the Object-type(s) whose Virtual Tables are listed in the FROM clause.

2586 Only columns in the SELECT clause MAY be in the ORDER BY clause.

2587 Collation rules for the ORDER BY clause are repository specific.

2588 **2.1.10.3 Escaping**

2589 Repositories MUST support the escaping of characters using a backslash (\) in the query statement. The
2590 backslash character (\) will be used to escape characters *within quoted strings* in the query as follows:

- 2591 1. \" will represent a double-quote ("") character
- 2592 2. \' will represent a single-quote(') character
- 2593 3. \\ will represent a backslash (\) character
- 2594 4. Within a LIKE string, \% and _ will represent the literal characters % and _, respectively.
- 2595 5. All other instances of a \ are errors.

2596 **2.1.11 Change Log**

2597 CMIS provides a “change log” mechanism to allow applications to easily discover the set of changes that
2598 have occurred to objects stored in the repository since a previous point in time. This change log can then
2599 be used by applications such as search services that maintain an external index of the repository to
2600 efficiently determine how to synchronize their index to the current state of the repository (rather than
2601 having to query for all objects currently in the repository).

2602 Entries recorded in the change log are referred to below as “change events”.

2603 Note that change events in the change log MUST be returned in ascending order from the time when the
2604 change event occurred.

2605 2.1.11.1 Completeness of the Change Log

2606 The Change Log mechanism exposed by a repository MAY be able to return an entry for every change
2607 ever made to content in the repository, or may only be able to return an entry for all changes made since
2608 a particular point in time. This “completeness” level of the change log is indicated via the [optional](#)
2609 [changesIncomplete](#) value found on the `getRepositoryInfo` service response

2610 However, repositories MUST ensure that if an application requests the entire contents of the repository’s
2611 change log, that the contents of the change log includes ALL changes made to any object in the
2612 repository *after* the first change listed in the change log. (I.e. repositories MAY truncate events from the
2613 change log on a “first-in first-out” basis, but not in any other order.)

2614 A Repository MAY record events such as filing/unfiling/moving of Documents as change events on the
2615 Documents, their parent Folder(s), or both the Documents and the parent Folders.

2616 2.1.11.2 Change Log Token

2617 The primary index into the change log of a repository is the “change log token”. The change log token is
2618 an opaque string that uniquely identifies a particular change in the change log.

2619 2.1.11.2.1 “Latest Change Token” repository information

2620 Repositories that support the `changeLogToken` event MUST expose the latest change log token (i.e. the
2621 change log token corresponding to the most recent change to any object in the repository) as a property
2622 returned by the `getRepositoryInfo` service.

2623 This will enable applications to begin “subscribing” to the change log for a repository by discovering what
2624 change log token they should use on a going-forward basis to discover change events to the repository.

2625 2.1.11.3 Change Event

2626 A change event represents a single action that occurred to an object in the repository that affected the
2627 persisted state of the object.

2628 A Repository that supports the change log capability MUST expose at least the following information for
2629 each change object:

- 2630 • **ID ObjectId:** The `ObjectId` of the object to which the change occurred
- 2631 • **Enum ChangeType:** An enumeration that indicates the type of the change. Valid values are:
 - 2632 o `created`: The object was created.
 - 2633 o `updated`: The object was updated.
 - 2634 o `deleted`: The object was deleted
 - 2635 o `security`: The access control or security policy for the object were changed.
- 2636 • **<Properties> properties:** Additionally, for events of `changeType` “`updated`”, the repository MAY
2637 optionally include the new values of properties on the object (if any).

2638 Repositories MUST indicate whether they include properties for “`updated`” change events via the [optional](#)
2639 [enumCapabilityChanges](#) capability.

2640

2641 2.2 Services

2642 Part I of the CMIS specification defines a set of services that are described in a protocol/binding-agnostic
2643 fashion.

2644 Every protocol binding of the CMIS specification MUST implement all of the methods described in this
2645 section or explain why the service is not implemented.

2646 However, the details of how each service & method is implemented will be described in those protocol
2647 binding specifications.

2648 2.2.1 Common Service Elements

2649 The following elements are common across many of the CMIS services.

2650 2.2.1.1 Paging

2651 All of the methods that allow for the retrieval of a collection of CMIS objects support paging of their result
2652 sets except where explicitly stated otherwise. The following pattern is used:

2653 Input Parameters:

- 2654 • **(optional) Integer maxItems:** This is the maximum number of items to return in a response. The
2655 repository MUST NOT exceed this maximum. Default is repository-specific.
- 2656 • **(optional) Integer skipCount:** This is the number of potential results that the repository MUST
2657 skip/page over before returning any results. Defaults to 0.

2658 Output Parameters:

- 2659 • **Boolean hasMoreItems:** TRUE if the Repository contains additional items after those contained
2660 in the response. FALSE otherwise. If TRUE, a request with a larger skipCount or larger
2661 maxItems is expected to return additional results (unless the contents of the repository has
2662 changed).
- 2663 • **Integer numItems:** If the repository knows the total number of items in a result set, the
2664 repository SHOULD include the number here. If the repository does not know the number of
2665 items in a result set, this parameter SHOULD not be set. The value in the parameter MAY NOT
2666 be accurate the next time the client retrieves the result set or the next page in the result set.

2667 If the caller of a method does not specify a value for maxItems, then the Repository MAY select an
2668 appropriate number of items to return, and MUST use the hasMoreItems output parameter to indicate if
2669 any additional results were not returned.

2670 Repositories MAY return a smaller number of items than the specified value for maxItems.

2671 Each binding will express the above in context and may have different mechanisms for communicating
2672 hasMoreItems and numItems.

2673 2.2.1.2 Retrieving additional information on objects in CMIS service calls

2674 Several CMIS services that return object information have the ability to return dependent object
2675 information as part of their response, such as the Allowable Actions for an object, rendition information,
2676 etc.

2677 The CMIS service methods that support returning a result set of objects will include the ability to return
2678 the following object information:

- 2679 • Properties (retrieves a subset instead of additional information)
- 2680 • Relationships
- 2681 • Renditions
- 2682 • ACLs
- 2683 • AllowableActions

2684

2685 This section describes the input parameter & output pattern for those services. All input parameters are
2686 optional.

2687 2.2.1.2.1 Properties

2688 **Description:** All of the methods that allow for the retrieval of properties for CMIS Objects have a
2689 "Property Filter" as an optional parameter, which allows the caller to specify a subset of properties for
2690 Objects that MUST be returned by the repository in the output of the method.

2691 Optional Input Parameter:

- 2692 • **String filter:** Value indicating which properties for Objects MUST be returned. Values are:
 2693 ○ **Not set:** The set of properties to be returned MUST be determined by the repository.
 2694 ○ **A comma-delimited list of property definition Query Names: The properties listed
 2695 MUST be returned.**
 2696 ○ **“*” :** All properties MUST be returned for all objects.

2697 Repositories SHOULD return only the properties specified in the property filter.

2698 **2.2.1.2.2 Relationships**

2699 **Description:** Used to retrieve the relationships in which the object(s) are participating.

2700 **Optional Input Parameter:**

- 2701 • **Enum includeRelationships:** Value indicating what relationships in which the objects returned
 2702 participate MUST be returned, if any. Values are:
 2703 **none** : No relationships MUST be returned. (Default).
 2704 **source** : Only relationships in which the objects returned are the source MUST be
 2705 returned.
 2706 **target** : Only relationships in which the objects returned are the target MUST be
 2707 returned.
 2708 **both** : Relationships in which the objects returned are the source or the target MUST be
 2709 returned.

2710 **Output Parameter for each object:**

- 2711 • **<Array> Relationships:** A collection of the relationship objects.

2712 **2.2.1.2.3 Policies**

2713 **Description:** Used to retrieve the policies currently applied to the object(s).

2714 **Optional Input Parameter:**

- 2715 • **Boolean includePolicyIds:** If TRUE, then the Repository MUST return the Ids of the policies
 2716 applied to the object. Defaults to FALSE.

2717 **Output Parameter or each object:**

- 2718 • **<Array> Policies:** A collection of the policy objects.

2719 **2.2.1.2.4 Renditions**

2720 **Description:** Used to retrieve the renditions of the object(s).

2721 **Optional Input Parameter:**

- 2722 • **String renditionFilter:** The Repository MUST return the set of renditions whose kind matches
 2723 this filter. See section below for the filter grammar.
 2724 ○ Defaults to “cmis:none”.

2725 **Output Parameter for each object:**

- 2726 • **<Array> Renditions:** The set of renditions.

2727 **2.2.1.2.4.1 Rendition Filter Grammar**

2728 The Rendition Filter grammar is defined as follows:

```

<renditionInclusion> ::= <none> | <wildcard> | <termlist>
<termlist> ::= <term> | <term> ',' <termlist>
<term> ::= <kind> | <mimetype>
<kind> ::= <text>
```

```
2733     <mimetype> ::= <type> '/' <subtype>
2734     <type> ::= <text>
2735     <subtype> ::= <text> | <wildcard>
2736     <text> ::= /* any char except whitespace */
2737     <wildcard> ::= '*'
2738     <none> ::= 'cmis:none'
```

2739 An inclusion pattern allows:

- 2740 • **Wildcard** : include all associated Renditions
- 2741 • **Comma-separated list of Rendition kinds or mimetypes** : include only those Renditions
2742 that match one of the specified kinds or mimetypes
- 2743 • **cmis:none**: (Default) exclude all associated Renditions

2744 Examples:

- 2745 • * (include all Renditions)
- 2746 • cmis:thumbnail (include only Thumbnails)
- 2747 • Image/* (include all image Renditions)
- 2748 • application/pdf, application/x-shockwave-flash (include web ready Renditions)
- 2749 • cmis:none (exclude all Renditions)

2.2.1.2.5 ACLs

2751 **Description:** Used to retrieve the ACLs for the object(s) described in the service response.

2752 **Optional Input Parameter:**

- 2753 • **Boolean includeACL**: If TRUE, then the Repository MUST return the ACLs for each object in
2754 the result set. Defaults to FALSE.

2755 **Output Parameter for each object:**

- 2756 • **<Array> ACLs**: The list of access control entries of the ACL for the object.

2.2.1.2.6 Allowable Actions

2758 **Description:** Used to retrieve the allowable actions for the object(s) described in the service response.

2759 **Optional Input Parameter:**

- 2760 • **Boolean includeAllowableActions**: If TRUE, then the Repository MUST return the
2761 available actions for each object in the result set. Defaults to FALSE.

2762 **Output Parameter for each object:**

- 2763 • **AllowableActions**: See `cmisAllowableActionsType` in the CMIS schema.

2.2.1.3 Change Tokens

2765 The CMIS base object-type definitions include an opaque string "ChangeToken" property that a
2766 Repository MAY use for optimistic locking and/or concurrency checking to ensure that user updates do
2767 not conflict.

2768 If a Repository provides values for the `ChangeLogToken` property for an Object, then all invocations of
2769 the "update" methods on that object (`updateProperties`, `setContentStream`, `deleteContentStream`) MUST
2770 provide the value of the `changeLogToken` property as an input parameter, and the Repository MUST
2771 throw an `updateConflictException` if the value specified for the `changeLogToken` does NOT match the
2772 `changeLogToken` value for the object being updated.

2773 **2.2.1.4 Exceptions**

2774 The following sections list the complete set of exceptions that MAY be returned by a repository in
2775 response to a CMIS service method call.

2776 **2.2.1.4.1 General Exceptions**

2777 The following exceptions MAY be returned by a repository in response to ANY CMIS service method call.

2778 The “Cause” field indicates the circumstances under which a repository SHOULD return a particular
2779 exception.

2780 **invalidArgument**

2781 Cause: One or more of the input parameters to the service method is missing or invalid.

2782

2783 **objectNotFound**

2784 Cause: The service call has specified an object that does not exist in the Repository.

2785

2786 **notSupported**

2787 Cause: The service method invoked requires [an optional capability](#) not supported by the
2788 repository.

2789

2790 **permissionDenied**

2791 Cause: The caller of the service method does not have sufficient permissions to perform the
2792 operation.

2793

2794 **runtime**

2795 Cause: Any other cause not expressible by another CMIS exception.

2796 **2.2.1.4.2 Specific Exceptions**

2797 The following exceptions MAY be returned by a repository in response to one or more CMIS service
2798 methods calls.

2799 For each exception, the general intent is listed as well as a list of the methods which MAY cause the
2800 exception to be thrown.

2801 **constraint**

2802 Intent: The operation violates a Repository- or Object-level constraint defined in the CMIS
2803 domain model.

2804 Methods:

2805 • **Navigation Services:**

2806 ○ `getObjectParents`

2807 • **Object Services:**

2808 ○ `createDocument`

2809 ○ `createDocumentFromSource`

2810 ○ `createFolder`

2811 ○ `createRelationship`

2812 ○ `createPolicy`

2813 ○ `updateProperties`

2814 ○ `moveObject`

```

2815          ○ deleteObject
2816          ○ setContentStream
2817          ○ deleteContentStream
2818      • Multi-filing Services:
2819          ○ addObjectToFolder
2820      • Versioning Services:
2821          ○ checkOut
2822          ○ cancelCheckOut
2823          ○ checkIn
2824      • Policy Services:
2825          ○ applyPolicy
2826          ○ removePolicy
2827      • Change Log Services:
2828          ○ getContentChanges
2829
2830  contentAlreadyExists
2831      Intent: The operation attempts to set the content stream for a Document that already has a
2832      content stream without explicitly specifying the "overwriteFlag" parameter.
2833      Methods:
2834          • Object Services:
2835              ○ setContentStream
2836
2837  filterNotValid
2838      Intent: The property filter or rendition filter input to the operation is not valid.
2839      Methods:
2840          • Navigation Services:
2841              ○ getDescendants
2842              ○ getChildren
2843              ○ getFolderParent
2844              ○ getObjectParents
2845              ○ getCheckedOutDocs
2846          • Object Services:
2847              ○ getProperties
2848              ○ getRenditions
2849              ○ getObject
2850              ○ getObjectByPath
2851          • Versioning Services:
2852              ○ getPropertiesOfLatestVersion
2853              ○ getAllVersions
2854          • Policy Services:
2855              ○ getAppliedPolicies
2856

```

2857 **nameConstraintViolation**
2858 Intent: The repository is not able to store the object that the user is creating/updating due to
2859 a name constraint violation.
2860 Methods:
2861 • **Object Services:**
2862 ○ createDocument
2863 ○ createDocumentFromSource
2864 ○ createFolder
2865 ○ updateProperties
2866 ○ moveObject
2867
2868 **storage**
2869 Intent: The repository is not able to store the object that the user is creating/updating due to
2870 an internal storage problem.
2871 Methods:
2872 • **Object Services:**
2873 ○ createDocument
2874 ○ createDocumentFromSource
2875 ○ createFolder
2876 ○ createRelationship
2877 ○ createPolicy
2878 ○ updateProperties
2879 ○ moveObject
2880 ○ setContentStream
2881 ○ deleteContentStream
2882 • **Versioning Services:**
2883 ○ checkOut
2884 ○ checkIn
2885
2886 **streamNotSupported**
2887 Intent: The operation is attempting to get or set a contentStream for a Document whose
2888 Object-type specifies that a content stream is not allowed for Document's of that type.
2889 Methods:
2890 • **Object Services:**
2891 ○ createDocument
2892 ○ createDocumentFromSource
2893 ○ getContentStream
2894 ○ setContentStream
2895 • **Versioning Services:**
2896 ○ checkIn
2897
2898 **updateConflict**

2899 Intent: The operation is attempting to update an object that is no longer current (as
2900 determined by the repository).

2901 Methods:

- **Object Services:**

- updateProperties
- moveObject
- deleteObject
- deleteTree
- setContentStream
- deleteContentStream

- **Versioning Services:**

- checkOut
- cancelCheckOut
- checkIn

versioning

2915 Intent: The operation is attempting to perform an action on a [non-current version](#) of a
2916 Document that cannot be performed on a non-current version.

2917 Methods:

- **Object Services:**

- updateProperties
- moveObject
- setContentStream
- deleteContentStream

- **Versioning Services:**

- checkOut
- cancelCheckOut
- checkIn

2.2.1.5 ACLs

2928 Those services which allow for the setting of ACLs may take the optional macro cmis:user which allows
2929 the caller to indicate the operation applies to the current authenticated user.

2.2.2 Repository Services

2931 The Repository Services (getRepositories, getRepositoryInfo, getTypeChildren, getTypeDescendants,
2932 getTypeDefinition) are used to discover information about the repository, including information about the
2933 repository and the object-types defined for the repository.

2.2.2.1 getRepositories

2935 **Description:** Returns a list of CMIS repositories available from this CMIS service endpoint.

2.2.2.1.1 Inputs

2937 None.

- 2938 **2.2.2.1.2 Outputs**
- 2939 A list of repository information, with (at least) the following information for each entry:
- 2940 • **ID repositoryId:** The identifier for the Repository.
- 2941 • **String repositoryName:** A display name for the Repository.
- 2942 **2.2.2.1.3 Exceptions Thrown & Conditions**
- 2943 See section 2.2.1.4.1 General Exceptions
- 2944 **2.2.2.2 getRepositoryInfo**
- 2945 **Description:** Returns information about the CMIS repository, the [optional capabilities](#) it supports and its
- 2946 Access Control information if applicable. .
- 2947 **2.2.2.2.1 Inputs**
- 2948 **Required:**
- 2949 • **ID repositoryId:** The identifier for the Repository.
- 2950 **2.2.2.2.2 Outputs**
- 2951 • **ID repositoryId:** The identifier for the Repository.
- 2952 ○ Note: This MUST be the same identifier as the input to the method.
- 2953 • **String repositoryName:** A display name for the Repository.
- 2954 • **String repositoryDescription:** A display description for the Repository.
- 2955 • **String vendorName:** A display name for the vendor of the Repository's underlying application.
- 2956 • **String productName:** A display name for the Repository's underlying application.
- 2957 • **String productVersion:** A display name for the version number of the Repository's
- 2958 underlying application.
- 2959 • **ID rootFolderId:** The ID of the Root Folder Object for the Repository.
- 2960 • **<List of capabilities>:** The set of values for the repository-optional capabilities specified in
- 2961 [section 2.1.1.1 Optional Capabilities](#)
- 2962 • **String latestChangeLogToken:** The change log token corresponding to the most recent
- 2963 change event for any object in the repository.
- 2964 • **Decimal cmisVersionSupported:** A decimal that indicates what version of the CMIS
- 2965 specification this repository supports as specified in 2.1.1.2 Implementation Information.
- 2966 • **URI thinClientURI:** A optional repository-specific URI pointing to the repository's web
- 2967 interface.
- 2968 • **Boolean changesIncomplete:** Indicates whether or not the repository's change log can return
- 2969 all changes ever made to any object in the repository or only changes made after a particular
- 2970 point in time. Applicable when the repository's optional capability `capabilityChanges` is not
- 2971 none.
- 2972 ○ If FALSE, then the change log can return all changes ever made to every object.
- 2973 ○ If TRUE, then the change log includes all changes made since a particular point in time,
- 2974 but not all changes ever made.
- 2975 • **<List of enum values> changesOnType:** Indicates whether changes are available for base
- 2976 types in the repository. Valid values are from `enumBaseObjectTypelds`. See [section 2.1.11 Change Log](#).

- 2978 ○ **cmis:document**
- 2979 ○ **cmis:folder**
- 2980 ○ **cmis:policy**
- 2981 ○ **cmis:relationship**
- 2982 • **Enum supportedPermissions**: specifies which types of permissions are supported.
 - 2983 ○ **basic**: indicates that the CMIS Basic permissions are supported.
 - 2984 ○ **repository**: Indicates that repository specific permissions are supported.
 - 2985 ○ **both**: indicates that both CMIS basic permissions and repository specific permissions are supported.
- 2987 • **Enum propagation**: The list of allowed values for applyACL, which control how non-direct ACEs are handled by the repository:
 - 2989 ○ **objectonly**: indicates that the repository is able to apply ACEs without changing the ACLs of other objects – i.e. ACEs are applied, potentially “breaking” the “sharing” dependency for non-direct ACEs.
 - 2992 ○ **propagate**: indicates that the repository is able to apply ACEs to a given object and propagate this change to all inheriting objects – i.e. ACEs are applied with the (intended) side effect to inheriting objects.
 - 2995 ○ **repositorydetermined**: indicates that the repository uses its own mechanisms to handle non-direct ACEs when applying ACLs.
- 2997 • **<Array> Permission permissions**: The list of repository-specific permissions the repository supports for managing ACEs (see section 2.8 Access Control).
- 2999 • **<Array> PermissionMapping mapping**: The list of mappings for the CMIS Basic permissions to allowable actions (see section 2.8 Access Control).
- 3001 • **String principalAnonymous**: If set, this field holds the principal who is used for anonymous access. This principal can then be passed to the ACL services to specify what permissions anonymous users should have.
- 3004 • **String principalAnyone**: If set, this field holds the principal who is used to indicate any authenticated user. This principal can then be passed to the ACL services to specify what permissions any authenticated user should have.

3007 The `cmisRepositoryInfoType` schema describes the markup that will be included in all CMIS protocol
3008 bindings to implement this service.

3009 **2.2.2.3 Exceptions Thrown & Conditions**

3010 See section 2.2.1.4.1 General Exceptions

3011 **2.2.2.3 getTypeChildren**

3012 **Description:** Returns the list of Object-Types defined for the Repository that are children of the specified
3013 Type.

3014 **2.2.2.3.1 Inputs**

3015 **Required:**

- 3016 • **String repositoryId**: The identifier for the Repository.

3017 **Optional:**

- 3018 • **String typeId**: The typeId of an Object-Type specified in the Repository.

- 3019 ○ If specified, then the Repository MUST return all of child types of the specified type.
- 3020 ○ If not specified, then the Repository MUST return all Base Object-Types.

- 3021 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
3022 definitions for each Object-Type returned.
3023 ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.
3024 • **Integer maxItems:** See section 2.2.1.1 Paging.
3025 • **Integer skipCount:** See section 2.2.1.1 Paging.

3026

2.2.2.3.2 Outputs

3027 **<Array> Object-Types:** The list of child [Object-Types](#) defined for the given typeId.

3028 **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3029 **Optional:**

3030 **Integer numItems:** See section 2.2.1.1 Paging.

3031

2.2.2.3.3 Exceptions Thrown & Conditions

3032 See section 2.2.1.4.1 General Exceptions

3033

2.2.2.4 getTypeDescendants

3034 **Description:** Returns the set of descendant [Object-Types](#) defined for the Repository under the specified
3035 Type.

3036 **Notes:**

- 3037 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
3038 • The order in which results are returned is repository-specific.

3039

2.2.2.4.1 Inputs

3040 **Required:**

- 3041 • **String repositoryId:** The identifier for the Repository.

3042 **Optional:**

- 3043 • **String typeId:** The typeId of an Object-Type specified in the Repository.
3044 ○ If specified, then the Repository MUST return all descendant types for the specified type.
3045 ○ If not specified, then the Repository MUST return all types and MUST ignore the value of
3046 the depth parameter
- 3047 • **Integer depth:** The number of levels of depth in the type hierarchy from which to return results.
3048 Valid values are:
3049 ○ **1:** Return only types that are children of the type.
3050 ○ **<Integer value greater than 1>:** Return only types that are children of the type and
3051 descendants up to **<value>** levels deep.
3052 ○ **-1:** Return ALL descendant types at all depth levels in the CMIS hierarchy.
3053 ○ The default value is repository specific and SHOULD be at least 2 or -1.
- 3054 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
3055 definitions for each Object-Type returned.
3056 ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

3057

2.2.2.4.2 Outputs

3058 **<Array> Object-Types:** The hierarchy of [Object-Types](#) defined for the Repository.

- 3059 **2.2.2.4.3 Exceptions Thrown & Conditions**
- 3060 See section 2.2.1.4.1 General Exceptions
- 3061 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
3062 an invalid depth.
- 3063 **2.2.2.5 getTypeDefinition**
- 3064 **Description:** Gets the definition of the specified Object-Type.Inputs
- 3065 **2.2.2.5.1 Inputs**
- 3066 **Required:**
- 3067 • **String repositoryId:** The identifier for the Repository.
- 3068 • **String typeId:** The typeId of an Object-Type specified in the Repository.
- 3069 **2.2.2.5.2 Outputs**
- 3070 • Object-type including all property definitions.
- 3071 **2.2.2.5.3 Exceptions Thrown & Conditions**
- 3072 See section 2.2.1.4.1 General Exceptions
- 3073 **2.2.3 Navigation Services**
- 3074 The Navigation Services (getDescendants, getChildren, getFolderParent, getObjectParents,
3075 getCheckedoutDocs), are used to traverse the folder hierarchy in a CMIS **Repository**, and to locate
3076 Documents that are checked out.
- 3077 **2.2.3.1 getChildren**
- 3078 **Description:** Gets the list of child objects contained in the specified folder.
- 3079 **Notes:**
- 3080 • If the Repository supports the optional “VersionSpecificFiling” capability, then the repository
3081 MUST return the document versions filed in the specified folder.
3082 ○ Otherwise, the latest version of the documents MUST be returned.
- 3083 **2.2.3.1.1 Inputs**
- 3084 **Required:**
- 3085 • **ID repositoryId:** The identifier for the Repository.
- 3086 • **ID folderId:** The identifier for the folder.
- 3087 **Optional:**
- 3088 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3089 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3090 • **String orderBy:** See ORDER BY Clause in the query
- 3091 • **String filter:** See **Error! Reference source not found**.section 2.2.1.2.1 Properties.
- 3092 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3093 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3094 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

- 3095 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
3096 child object for use in constructing that object's path.

3097 **2.2.3.1.2 Outputs**

- 3098 • **<Array> ObjectResults:** A list of the child objects for the specified folder. Each object result
3099 MUST include the following elements if they are requested:
3100 ○ **<Array> Properties:** The list of properties for the object.
3101 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
3102 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
3103 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
3104 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

- 3105 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3106 **Optional:**

- 3107 **Integer numItems:** See section 2.2.1.1 Paging.

3108 **2.2.3.1.3 Exceptions Thrown & Conditions**

- 3109 • See section 2.2.1.4.1 General Exceptions
3110 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
3111 parameter is not valid.
3112 • **invalidArgument:** if the specified folder is not a folder

3113 **2.2.3.2 getDescendants**

3114 **Description:** Gets the set of **descendant objects** contained in the specified folder or any of its child-
3115 folders.

3116 **Notes:**

- 3117 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
3118 • The order in which results are returned is repository-specific..
3119 • If the Repository supports the optional capability **capabilityVersionSpecificFiling**, then
3120 the repository MUST return the document versions filed in the specified folder or its descendant
3121 folders. Otherwise, the latest version of the documents MUST be returned.
3122 • If the Repository supports the optional capability **capabilityMutlifiling** and the same
3123 document is encountered multiple times in the hierarchy, then the repository MUST return that
3124 document each time it is encountered.

3125 **2.2.3.2.1 Inputs**

3126 **Required:**

- 3127 • **ID repositoryId:** The identifier for the Repository.
3128 • **ID folderId:** The identifier for the folder.

3129 **Optional:**

- 3130 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results.
3131 Valid values are:
3132 ○ **1:** Return only objects that are children of the folder.
3133 ○ **<Integer value greater than 1>:** Return only objects that are children of the folder and
3134 descendants up to <value> levels deep.

- 3135 ○ -1: Return ALL descendant objects at all depth levels in the CMIS hierarchy.
 3136 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 3137 • **String filter:** See section 2.2.1.2.1 Properties.
 - 3138 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
 - 3139 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
 - 3140 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3141 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each child object for use in constructing that object's path.

3143 **2.2.3.2.2 Outputs**

- 3144 • **<Array> ObjectResults:** A list of the descendant objects for the specified folder. Each object result MUST include the following elements if they are requested:
 - 3145 ○ **<Array> Properties:** The list of properties for the object.
 - 3146 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 3147 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 3148 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3149 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

3151 **2.2.3.2.3 Exceptions Thrown & Conditions**

- 3152 See section 2.2.1.4.1 General Exceptions
- 3153 • **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter is not valid.
 - 3155 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with "depth = 0".
 - 3157 • **invalidArgument:** if the specified folder is not a folder

3158 **2.2.3.3 getFolderTree**

3159 **Description:** Gets the set of descendant folder objects contained in the specified folder.

3160

3161 **Notes:**

- 3162 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3163 • The order in which results are returned is repository-specific..

3164 **2.2.3.3.1 Inputs**

3165 **Required:**

- 3166 • **ID repositoryId:** The identifier for the Repository.
- 3167 • **ID folderId:** The identifier for the folder.

3168 **Optional:**

- 3169 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results.
 3170 Valid values are:
 - 3171 ○ **1:** Return only folders that are children of the folder.
 - 3172 ○ **<Integer value greater than 1>:** Return only folders that are children of the folder and descendant folders up to <value> levels deep.

- 3174 ○ -1: Return ALL descendant folders at all depth levels in the CMIS hierarchy.
3175 ○ The default value is repository specific and SHOULD be at least 2 or -1
3176 • **String filter:** See section 2.2.1.2.1 Properties.
3177 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
3178 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
3179 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
3180 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
3181 child object for use in constructing that object's path.

3182 **2.2.3.3.2 Outputs**

- 3183 • **<Array> ObjectResults:** A list of the descendant folders for the specified folder. Each object
3184 result MUST include the following elements if they are requested:
3185 ○ **<Array> Properties:** The list of properties for the object.
3186 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
3187 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
3188 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
3189 ○ **String pathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

3190 **2.2.3.3.3 Exceptions Thrown & Conditions**

- 3191 • See section 2.2.1.4.1 General Exceptions
3192 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
3193 parameter is not valid.
3194 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
3195 an invalid depth
3196 • **invalidArgument:** if the specified folder is not a folder

3199 **2.2.3.4 getFolderParent**

3200 **Description:** Gets the parent folder object for the specified folder object.

3201 **2.2.3.4.1 Inputs**

3202 **Required:**

- 3203 • **ID repositoryId:** The identifier for the Repository.
- 3204 • **ID folderId:** The identifier for the folder.

3205 **Optional:**

- 3206 • **String filter:** See section 2.2.1.2.1 Properties.**Error! Reference source not found.**

3207 **2.2.3.4.2 Outputs**

- 3208 • **Object:** The parent folder object of the specified folder.

3209 **2.2.3.4.3 Exceptions Thrown & Conditions**

- 3210 • See section 2.2.1.4.1 General Exceptions

- 3211 • **filterNotValid**: The Repository MUST throw this exception if this property filter input
3212 parameter is not valid.
3213 • **invalidArgument**: The Repository MUST throw this exception if the folderId input is the root
3214 folder.

3215 **2.2.3.5 getObjectParents**

3216 **Description:** Gets the parent folder(s) for the specified non-folder, fileable object.

3217 **2.2.3.5.1 Inputs**

3218 **Required:**

- 3219 • **ID repositoryId**: The identifier for the Repository.
- 3220 • **ID objectId**: The identifier for the object.

3221 **Optional:**

- 3222 • **String filter**: See section 2.2.1.2.1 Properties
- 3223 • **Enum includeRelationships**: See section 2.2.1.2.2 Relationships.
- 3224 • **String renditionFilter**: See section 2.2.1.2.4 Renditions.
- 3225 • **Boolean includeAllowableActions**: See section 2.2.1.2.6 Allowable Actions.
- 3226 • **Boolean includeRelativePathSegment**: See section 2.1.5.3 Paths.

3227 **2.2.3.5.2 Outputs**

- 3228 • **<Array> ObjectResults**: A list of the parent folder(s) of the specified objects. Empty for unfiled
3229 objects or for the root folder. Each object result MUST include the following elements if they are
3230 requested:
 - 3231 ◦ **<Array> Properties**: The list of properties for the object.
 - 3232 ◦ **<Array> Relationships**: See section 2.2.1.2.2 Relationships.
 - 3233 ◦ **<Array> Renditions**: See section 2.2.1.2.4 Renditions.
 - 3234 ◦ **AllowableActions**: See section 2.2.1.2.6 Allowable Actions.
 - 3235 ◦ **String relativePathSegment**: If includeRelativePathSegment was TRUE. See section
3236 2.1.5.3 Paths.

3237 **2.2.3.5.3 Exceptions Thrown & Conditions**

- 3238 • See section 2.2.1.4.1 General Exceptions
- 3239 • **constraint**: The Repository MUST throw this exception if this method is invoked on an object
3240 who Object-Type Definition specifies that it is not fileable.
- 3241 • **filterNotValid**: The Repository MUST throw this exception if this property filter input
3242 parameter is not valid.
- 3243 • **invalidArgument**: if the specified folder is not a folder

3244 **2.2.3.6 getCheckedOutDocs**

3245 **Description:** Gets the list of documents that are checked out that the user has access to.

3246 **2.2.3.6.1 Inputs**

3247 **Required:**

- 3248 • **ID repositoryId:** The identifier for the Repository.
- 3249 **Optional:**
- 3250 • **ID folderId:** The identifier for a folder in the repository from which documents should be returned.
- 3251 ○ If specified, the Repository MUST only return checked out documents that are child-
3252 objects of the specified folder.
- 3253 ○ If not specified, the Repository MUST return checked out documents from anywhere in
3254 the repository hierarchy.
- 3255 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3256 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3257 • **String orderBy:** See ORDER BY Clause in the query
- 3258 • **String filter:** See section 2.2.1.2.1 Properties.
- 3259 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3260 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3261 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3262 2.2.3.6.2 Outputs

- 3263 • **<Array> ObjectResults:** A list of checked out documents. Each object result MUST include the
3264 following elements if they are requested:
- 3265 ○ **<Array> Properties:** The list of properties for the object.
- 3266 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3267 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3268 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3269 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3270 **Optional:**

3271 **Integer numItems:** See section 2.2.1.1 Paging.

3273 2.2.3.6.3 Exceptions Thrown & Conditions

- 3274 • See section 2.2.1.4.1 General Exceptions
- 3275 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
3276 parameter is not valid.

3277 2.2.4 Object Services

3278 CMIS provides ID-based CRUD (**Create**, **Retrieve**, **Update**, **Delete**), operations on objects in a Repository.

3279 2.2.4.1 createDocument

3280 **Description:** Creates a document object of the specified type (given by the cmis:objectTypeId property)
3281 in the (optionally) specified location.

3282 2.2.4.1.1 Inputs

3283 **Required:**

- 3284 • **ID repositoryId:** The identifier for the Repository.

- 3285 • **<Array> properties:** The property values that MUST be applied to the newly-created Document
3286 Object.

3287 **Optional:**

- 3288 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
3289 created Document Object.
3290 ◦ This parameter MUST be specified if the Repository does NOT support the optional
3291 “unfiling” capability.
- 3292 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the newly-
3293 created Document Object. The method of passing the contentStream to the server and the
3294 encoding mechanism will be specified by each specific binding. MUST be required if the type
3295 requires it.
- 3296 • **Enum versioningState:** An enumeration specifying what the versioning state of the newly-created
3297 object MUST be. Valid values are:
3298 ◦ none: The document MUST be created as a non-versionable document.
3299 ◦ checkedout: The document MUST be created in the checked-out state.
3300 ◦ major (**default**): The document MUST be created as a major version
3301 ◦ minor: The document MUST be created as a minor version.
- 3302 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
3303 object.
- 3304 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
3305 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3306 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3307 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
3308 specified.

3309

2.2.4.1.2 Outputs

3310 **ID objectId:** The ID of the newly-created document.

3311

2.2.4.1.3 Exceptions Thrown & Conditions

- 3312 • See section 2.2.1.4.1 General Exceptions
- 3313 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
3314 met:
3315 ◦ The cmis:objectTypeId property value is not an Object-Type whose baseType is
3316 “Document”.
3317 ◦ The cmis:objectTypeId property value is NOT in the list of AllowedChildObjectTypids of
3318 the parent-folder specified by folderId.
3319 ◦ The value of any of the properties violates the min/max/required/length constraints
3320 specified in the property definition in the Object-Type.
3321 ◦ The “contentStreamAllowed” attribute of the Object-Type definition specified by the
3322 cmis:objectTypeId property value is set to “required” and no contentStream input
3323 parameter is provided.
3324 ◦ The “versionable” attribute of the Object-Type definition specified by the
3325 cmis:objectTypeId property value is set to FALSE and a value for the versioningState
3326 input parameter is provided that is something other than “none”.
3327 ◦ The “versionable” attribute of the Object-Type definition specified by the
3328 cmis:objectTypeId property value is set to TRUE and the value for the versioningState
3329 input parameter is provided that is “none”.

- 3330 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the
3331 cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
- 3332 ○ The “controllableACL” attribute of the Object-Type definition specified by the
3333 cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
- 3334 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3335 repository.
- 3336 • nameConstraintViolation: See section 2.2.1.4.2 Specific Exceptions. If the repository
3337 detects a violation with the given cmis:name property value, the repository MAY throw this
3338 exception or chose a name which does not conflict.
- 3339 • storage: See section 2.2.1.4.2 Specific Exceptions.
- 3340 • streamNotSupported: The Repository MUST throw this exception if the
3341 “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId
3342 property value is set to “not allowed” and a contentStream input parameter is provided.

3343 **2.2.4.2 createDocumentFromSource**

3344 **Description:** Creates a document object as a copy of the given source document in the (optionally)
3345 specified location.

3346 **2.2.4.2.1 Inputs**

3347 **Required:**

- 3348 • **ID repositoryId:** The identifier for the Repository.
- 3349 • **ID sourceId:** The identifier for the source document.

3350 **Optional:**

- 3351 • **<Array> properties:** The property values that MUST be applied to the Object. This list of
3352 properties SHOULD only contain properties whose values differ from the source document.
- 3353 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
3354 created Document Object.
 - 3355 ○ This parameter MUST be specified if the Repository does NOT support the optional
3356 “unfiling” capability.
- 3357 • **Enum versioningState:** An enumeration specifying what the versioning state of the newly-created
3358 object MUST be. Valid values are:
 - 3359 ○ **none:** The document MUST be created as a non-versionable document.
 - 3360 ○ **checkedout:** The document MUST be created in the checked-out state.
 - 3361 ○ **major (default):** The document MUST be created as a major version
 - 3362 ○ **minor:** The document MUST be created as a minor version.
- 3363 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
3364 object.
- 3365 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
3366 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3367 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3368 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
3369 specified.

3370 **2.2.4.2.2 Outputs**

3371 **ID objectId:** The ID of the newly-created document.

3372 **2.2.4.2.3 Exceptions Thrown & Conditions**

- 3373 • See section 2.2.1.4.1 General Exceptions
- 3374 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are met:
- 3376 o The sourceId is not an Object whose baseType is “Document”.
- 3377 o The source document’s cmis:objectTypeId property value is NOT in the list of AllowedChildObjectTypids of the parent-folder specified by folderId.
- 3379 o The “versionable” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to FALSE and a value for the versioningState input parameter is provided that is something other than “none”.
- 3382 o The “versionable” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to TRUE and the value for the versioningState input parameter is provided that is “none”.
- 3385 o The “controllablePolicy” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
- 3387 o The “controllableACL” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
- 3389 o At least one of the permissions is used in an ACE provided which is not supported by the repository.
- 3391 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository detects a violation with the given cmis:name property value, the repository MAY throw this exception or chose a name which does not conflict.
- 3394 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3395 • **streamNotSupported:** The Repository MUST throw this exception if the “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to “not allowed” and a contentStream input parameter is provided.

3398 **2.2.4.3 createFolder**

3399 **Description:** Creates a folder object of the specified type in the specified location.

3400 **2.2.4.3.1 Inputs**

3401 **Required:**

- 3402 • **ID repositoryId:** The identifier for the Repository.
- 3403 • **<Array> properties:** The property values that MUST be applied to the newly-created Folder Object.
- 3405 • **ID folderId:** The identifier for the folder that MUST be the parent folder for the newly-created Folder Object.

3407 **Optional:**

- 3408 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Folder object.
- 3409 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Folder object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3411 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created Folder object, either using the ACL from folderId if specified, or being ignored if no folderId is specified.

- 3414 **2.2.4.3.2 Outputs**
- 3415 • **ID objectId:** The ID of the newly-created folder.
- 3416 **2.2.4.3.3 Exceptions Thrown & Conditions**
- 3417 • See section 2.2.1.4.1 General Exceptions
- 3418 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are met:
- 3419 ○ The cmis:objectTypeId property value is not an Object-Type whose baseType is “Folder”.
- 3420 ○ The value of any of the properties violates the min/max/required/length constraints specified in the property definition in the Object-Type.
- 3421 ○ The cmis:objectTypeId property value is NOT in the list of AllowedChildObjectTypids of the parent-folder specified by folderId.
- 3422 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
- 3423 ○ The “controllableACL” attribute of the Object-Type definition specified by the cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
- 3424 ○ At least one of the permissions is used in an ACE provided which is not supported by the repository.
- 3425 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository detects a violation with the given cmis:name property value, the repository MAY throw this exception or chose a name which does not conflict.
- 3426 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3427 **2.2.4.4 createRelationship**
- 3428 **Description:** Creates a relationship object of the specified type
- 3429 **2.2.4.4.1 Inputs**
- 3430 **Required:**
- 3431 • **ID repositoryId:** The identifier for the Repository.
- 3432 • **<Array> properties:** The property values that MUST be applied to the newly-created Relationship Object.
- 3433 **Optional:**
- 3434 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Relationship object.
- 3435 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Relationship object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3436 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created Relationship object, either using the ACL from folderId if specified, or being ignored if no folderId is specified.
- 3437 **2.2.4.4.2 Outputs**
- 3438 • **ID objectId:** The ID of the newly-created relationship.
- 3439 **2.2.4.4.3 Exceptions Thrown & Conditions**
- 3440 • See section 2.2.1.4.1 General Exceptions

- 3454 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
3455 met:
3456 ○ The cmis:objectTypeId property value is not an Object-Type whose baseType is
3457 “Relationship”.
3458 ○ The value of any of the properties violates the min/max/required/length constraints
3459 specified in the property definition in the Object-Type.
3460 ○ The sourceObjectId's ObjectType is not in the list of “allowedSourceTypes” specified by
3461 the Object-Type definition specified by cmis:objectTypeId property value.
3462 ○ The targetObjectId's ObjectType is not in the list of “allowedTargetTypes” specified by the
3463 Object-Type definition specified by cmis:objectTypeId property value.
3464 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the
3465 cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
3466 ○ The “controllableACL” attribute of the Object-Type definition specified by the
3467 cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
3468 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3469 repository.
3470 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

2.2.4.5 createPolicy

Description: Creates a policy object of the specified type

2.2.4.5.1 Inputs

Required:

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

Optional:

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

2.2.4.5.2 Outputs

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

2.2.4.5.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions
- **constraint:** The Repository MUST throw this exception if ANY of the following conditions are met:
 - The cmis:objectTypeId property value is not an Object-Type whose baseType is “Policy”.

- 3496 ○ The value of any of the properties violates the min/max/required/length constraints
3497 specified in the property definition in the Object-Type.
3498 ○ The cmis:objectTypeId property value is NOT in the list of AllowedChildObjectTypeIds of
3499 the parent-folder specified by folderId.
3500 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the
3501 cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
3502 ○ The “controllableACL” attribute of the Object-Type definition specified by the
3503 cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
3504 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3505 repository.
3506 • storage: See section 2.2.1.4.2 Specific Exceptions.

2.2.4.6 getAllowableActions

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

2.2.4.6.1 Inputs

Required:

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

2.2.4.6.2 Outputs

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

2.2.4.6.3 Exceptions Thrown & Conditions

See section 2.2.1.4.1 General Exceptions

2.2.4.7 getObject

Description: Gets the specified information for the Object.

2.2.4.7.1 Inputs

Required:

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

Optional:

- **String filter:** See **Error! Reference source not found.** section 2.2.1.2.1 Properties.
- **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

2.2.4.7.2 Outputs

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

- 3532 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
3533 **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
3534 **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
3535 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
3536 **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3537

2.2.4.7.3 Exceptions Thrown & Conditions

- 3538 See section 2.2.1.4.1 General Exceptions
3539 **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter
3540 is not valid.

3541

2.2.4.8 getProperties

3542 **Description:** Gets the list of properties for an Object.

3543

2.2.4.8.1 Inputs

3544 **Required:**

- 3545 • **ID repositoryId:** The identifier for the Repository.
3546 • **ID objectId:** The identifier for the object

3547 **Optional:**

- 3548 • **String filter:** See **Error! Reference source not found**.section 2.2.1.2.1 Properties.

3549

2.2.4.8.2 Outputs

3550 **<Array> Properties:** The list of properties for the object.

3551

2.2.4.8.3 Exceptions Thrown & Conditions

- 3552 See section 2.2.1.4.1 General Exceptions
3553 **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter
3554 is not valid.

3555

2.2.4.9 getObjectByPath

3556 **Description:** Gets the specified object.

3557

2.2.4.9.1 Inputs

3558 **Required:**

- 3559 • **ID repositoryId:** The identifier for the Repository.
3560 • **String path:** The path to the object. See section 2.1.5.3 Paths.

3561 **Optional:**

- 3562 • **String filter:** See section 2.2.1.2.1 Properties.
3563 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3564

2.2.4.9.2 Outputs

3565 **<Array> Properties:** The list of properties for the object.

3566 **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

- 3567 **2.2.4.9.3 Exceptions Thrown & Conditions**
- 3568
 - See section 2.2.1.4.1 General Exceptions
- 3569 **filterNotValid**: The Repository MUST throw this exception if this property filter input parameter
3570 is not valid.
- 3571 **2.2.4.10 getContentStream**
- 3572 **Description**: Gets the content stream for the specified Document object, or gets a rendition stream for a
3573 specified rendition of a document or folder object.
- 3574 **Notes**: Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,
3575 in a manner appropriate to that protocol.
- 3576 **2.2.4.10.1 Inputs**
- 3577 **Required**:
- 3578
 - **ID repositoryId**: The identifier for the Repository.
 - **ID objectId**: The identifier for the object
- 3580 **Optional**:
- 3581
 - **ID streamId**: The identifier for the rendition stream, when used to get a rendition stream. For
3582 Documents, if not provided then this method returns the content stream. For Folders, it MUST be
3583 provided.
- 3584 **2.2.4.10.2 Outputs**
- 3585
 - **<Stream> ContentStream**: The specified content stream or rendition stream for the object.
- 3586 **2.2.4.10.3 Exceptions Thrown & Conditions**
- 3587
 - See section 2.2.1.4.1 General Exceptions
- 3588 **constraint**: The Repository MUST throw this exception if the object specified by objectId does
3589 NOT have a content stream or rendition stream.
- 3590 **2.2.4.11 getRenditions**
- 3591 **Description**: Gets the list of associated Renditions for the specified object. Only rendition attributes are
3592 returned, not rendition stream.
- 3593 **Notes**: Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,
3594 in a manner appropriate to that protocol.
- 3595 **2.2.4.11.1 Inputs**
- 3596 **Required**:
- 3597
 - **ID repositoryId**: The identifier for the Repository.
 - **ID objectId**: The identifier for the object
- 3599 **Optional**:
- 3600
 - **String renditionFilter**: See Section 2.2.1.2.4
 - **Integer maxItems**: See section 2.2.1.1 Paging.
 - **Integer skipCount**: See section 2.2.1.1 Paging.
- 3603 **2.2.4.11.2 Outputs**
- 3604
 - **<Array> Renditions**: The set of renditions available on this object

3605 2.2.4.11.3 Exceptions Thrown & Conditions

- 3606 • See section 2.2.1.4.1 General Exceptions
- 3607 • notSupported: The service method requires functionality that is not supported by the repository
- 3609 • filterNotValid: The filter specified is not valid

3610 2.2.4.12 updateProperties

3611 **Description:** Updates properties of the specified object.

3612 **Notes:**

- 3613 • 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.
- 3615 • Each CMIS protocol bindings MUST specify whether the updateProperties service MUST always include all updatable properties, or only those properties whose values are different than the original value of the object.

3618 2.2.4.12.1 Inputs

3619 **Required:**

- 3620 • **ID repositoryId:** The identifier for the Repository.
- 3621 • **ID objectId:** The identifier of the object to be updated.
- 3622 • **<Array> properties:** The updated property values that MUST be applied to the Object.

3623 **Optional:**

- 3624 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3625 2.2.4.12.2 Outputs

- 3626 • **ID objectId:** The ID of the updated object.
- 3627 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3628 2.2.4.12.3 Exceptions Thrown & Conditions

- 3629 • See section 2.2.1.4.1 General Exceptions
- 3630 • **constraint:** The Repository MUST throw this exception if the value of any of the properties violates the min/max/required/length constraints specified in the property definition in the Object-Type.
- 3633 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY throw this exception or chose a name which does not conflict.
- 3635 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3636 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 3637 • **versioning:** The Repository MUST throw this exception if ANY of the following conditions are met:
 - 3639 o The object is not checked out and ANY of the properties being updated are defined in their Object-Type definition have an attribute value of *Updatability* when checked-out.
 - 3641 o Additionally, the repository MAY throw this exception if the object is a non-current Document Version.

3643 **2.2.4.13 moveObject**

3644 **Description:** Moves the specified file-able object from one folder to another.

3645 **2.2.4.13.1 Inputs**

3646 **Required:**

- 3647 • **ID repositoryId:** The identifier for the Repository.
- 3648 • **ID objectId:** The identifier of the object to be moved.
- 3649 • **ID targetFolderId:** The folder into which the object is to be moved.
- 3650 • **ID sourceFolderId:** The folder from which the object is to be moved.

3651 **2.2.4.13.2 Outputs**

- 3652 • **ID objectId:** The identifier of the object to be moved.

3653 **2.2.4.13.3 Exceptions Thrown & Conditions**

- 3654 • See section 2.2.1.4.1 General Exceptions
- 3655 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with a missing sourceFolderId or the sourceFolderId doesn't match the specified object's parent folder (or one of the parent folders if the repository supports multifiling.).
- 3658 • **constraint:** The Repository MUST throw this exception if the cmis:objectTypeId property value of the given object is NOT in the list of AllowedChildObjectTypids of the parent-folder specified by targetFolderId.
- 3661 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY throw this exception or chose a name which does not conflict.
- 3663 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3664 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 3665 • **versioning:** The repository MAY throw this exception if the object is a non-current Document Version.

3667 **2.2.4.14 deleteObject**

3668 **Description:** Deletes the specified object.

3669 **2.2.4.14.1 Inputs**

3670 **Required:**

- 3671 • **ID repositoryId:** The identifier for the Repository.
- 3672 • **ID objectId:** The identifier of the object to be deleted.

3673 **Optional:**

- 3674 • **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.

3678

3679 **2.2.4.14.2 Exceptions Thrown & Conditions**

- 3680 • See section 2.2.1.4.1 General Exceptions

- 3681 • **constraint**: The Repository MUST throw this exception if the method is invoked on a Folder
3682 object that contains one or more objects.
3683 • **updateConflict**: See section 2.2.1.4.2 Specific Exceptions.

3684 **2.2.4.15 deleteTree**

3685 **Description:** Deletes the specified folder object and all of its child- and descendant-objects.

3686 **Notes:**

- 3687 • A Repository MAY attempt to delete child- and descendant-objects of the specified folder in any
3688 order.
3689 • Any child- or descendant-object that the Repository cannot delete MUST persist in a valid state in
3690 the CMIS domain model.
3691 • This is not atomic.
3692 • However, if **deleteSinglefiled** is chosen and some objects fail to delete, then single-filed objects
3693 are either deleted or kept, never just unfiled. This is so that a user can call this command again to
3694 recover from the error by using the same tree.

3695 **2.2.4.15.1 Inputs**

3696 **Required:**

- 3697 • **ID repositoryId**: The identifier for the Repository.
3698 • **ID folderId**: The identifier of the folder to be deleted.

3699 **Optional:**

- 3700 • **Boolean allVersions**: If TRUE (default), then delete all versions of the document. If FALSE,
3701 delete only the document object specified. The Repository MUST ignore the value of this
3702 parameter when this service is invoke on a non-document object or non-versionable document
3703 object.
3704 • **Enum unfileObjects**: An enumeration specifying how the repository MUST process file-able
3705 child- or descendant-objects. Valid values are:
3706
 - **unfile**: Unfile all fileable objects.
 - **deleteSinglefiled**: Delete all fileable non-folder objects whose only parent-folders are in
3707 the current folder tree. Unfile all other fileable non-folder objects from the current folder tree.
 - **delete (default)**: Delete all fileable objects.
3708 • **boolean continueOnFailure**: If TRUE, then the repository SHOULD continue attempting to
3709 perform this operation even if deletion of a child- or descendant-object in the specified folder
3710 cannot be deleted.
3711
 - If FALSE (**default**), then the repository SHOULD abort this method when it fails to delete a
3712 single child- or descendant-object.

3715 **2.2.4.15.2 Outputs**

- 3716 • **<Array> ID failedToDelete**: A list of identifiers of objects in the folder tree that were not deleted.

3717 **2.2.4.15.3 Exceptions Thrown & Conditions**

- 3718 • See section 2.2.1.4.1 General Exceptions
3719 • **updateConflict**: See section 2.2.1.4.2 Specific Exceptions.

3720 **2.2.4.16 setContentStream**

3721 **Description:** Sets the content stream for the specified Document object.

3722 **Notes:** A Repository MAY automatically create new Document versions as part of this service method.
3723 Therefore, the obejctId output NEED NOT be identical to the objectId input.

3724 **2.2.4.16.1 Inputs**

3725 **Required:**

- 3726 • **ID repositoryId:** The identifier for the Repository.
3727 • **ID objectId:** The identifier for the Document object.
3728 • **<contentStream> contentStream:** The Content Stream

3729 **Optional:**

- 3730 • **Boolean overwriteFlag:** If TRUE (default), then the Repository MUST replace the existing
3731 content stream for the object (if any) with the input contentStream.
3732 o If FALSE, then the Repository MUST only set the input contentStream for the object if the
3733 object currently does not have a content-stream.
3734 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3735 **2.2.4.16.2 Outputs**

- 3736 • **ID objectId:** The ID of the document.
3737 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3738 **2.2.4.16.3 Exceptions Thrown & Conditions**

- 3739 • See section 2.2.1.4.1 General Exceptions
3740 • **contentAlreadyExists:** The Repository MUST throw this exception if the input parameter
3741 overwriteFlag is FALSE and the Object already has a content-stream.
3742 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
3743 • **streamNotSupported:** The Repository MUST throw this exception if the
3744 “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId
3745 property value of the given document is set to “notallowed”.
3746 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
3747 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
3748 Version.

3749 **2.2.4.17 deleteContentStream**

3750 **Description:** Deletes the content stream for the specified Document object.

3751 **Notes:** A Repository MAY automatically create new Document versions as part of this service method.
3752 Therefore, the objectId output NEED NOT be identical to the objectId input.

3753 **2.2.4.17.1 Inputs**

3754 **Required:**

- 3755 • **ID repositoryId:** The identifier for the Repository.
3756 • **ID objectId:** The identifier for the Document object.

3757 **Optional:**

- 3758 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3759 **2.2.4.17.2 Outputs**

- 3760 • **ID objectId:** The ID of the Document object.
3761 • **String changeToken:** See section 2.2.1.3 Change Tokens.

3762 **2.2.4.17.3 Exceptions Thrown & Conditions**

- 3763 • See section 2.2.1.4.1 General Exceptions
3764 • **constraint:** The Repository MUST throw this exception if the Object's Object-Type definition
3765 "contentStreamAllowed" attribute is set to "required".
3766 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
3767 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
3768 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
3769 Version.

3770 **2.2.5 Multi-filing Services**

3771 The Multi-filing services (*addObjectToFolder*, *removeObjectFromFolder*) are supported only if the
3772 repository supports the multifiling or unfileing [optional capabilities](#). The Multi-filing Services are used to
3773 file/un-file objects into/from folders.

3774 This service is NOT used to create or delete objects in the repository.

3775 **2.2.5.1 addObjectToFolder**

3776 **Description:** Adds an existing fileable non-folder object to a folder.

3777 **2.2.5.1.1 Inputs**

3778 **Required:**

- 3779 • **ID repositoryId:** The identifier for the Repository.
3780 • **ID objectId:** The identifier of the object.
3781 • **ID folderId:** The folder into which the object is to be filed.

3782 **Optional:**

- 3783 • **Boolean allVersions:** Add all versions of the object to the folder if the repository supports
3784 version-specific filing. Defaults to TRUE.

3785 **2.2.5.1.2 Exceptions Thrown & Conditions**

- 3786 • See section 2.2.1.4.1 General Exceptions.
3787 • **constraint:** The Repository MUST throw this exception if the cmis:objectTypeld property value
3788 of the given object is NOT in the list of AllowedChildObjectTypelds of the parent-folder specified
3789 by folderId.

3790 **2.2.5.2 removeObjectFromFolder**

3791 **Description:** Removes an existing fileable non-folder object from a folder.

3792 **2.2.5.2.1 Inputs**

3793 **Required:**

- 3794 • **ID repositoryId:** The identifier for the Repository.
3795 • **ID objectId:** The identifier of the object.

- 3796 **Optional:**
- 3797 • **ID folderId:** The folder from which the object is to be removed.
- 3798 ◦ If no value is specified, then the Repository MUST remove the object from all folders in which
- 3799 it is currently filed.

3800 **2.2.5.2.2 Exceptions Thrown & Conditions**

- 3801 • See section 2.2.1.4.1 General Exceptions

3802 **2.2.6 Discovery Services**

3803 The Discovery Services (*query*) are used to search for query-able objects within the Repository.

3804 **2.2.6.1 query**

3805 **Description:** Executes a CMIS query statement against the contents of the Repository.

3806 **2.2.6.1.1 Inputs**

3807 **Required:**

- 3808 • **ID repositoryId:** The identifier for the Repository.
- 3809 • **String statement:** CMIS query to be executed. (See section 2.1.10 Query.)

3810 **Optional:**

3811 • **Boolean searchAllVersions:**

- 3812 ◦ If TRUE, then the Repository MUST include latest and non-latest versions of document
3813 objects in the query search scope.
- 3814 ◦ If FALSE (**default**), then the Repository MUST only include latest versions of documents
3815 in the query search scope.
- 3816 ◦ If the Repository does not support the optional capabilityAllVersionsSearchable
3817 capability, then this parameter value MUST be set to FALSE.

3818 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.

- 3819 ◦ Note: For query statements where the SELECT clause contains properties from only one
3820 virtual table reference (i.e. referenced object-type), any value for this enum may be used.
3821 If the SELECT clause contains properties from more than one table, then the value of this
3822 parameter MUST be "none".

3823 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.

3824 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

- 3825 ◦ Note: For query statements where the SELECT clause contains properties from only one
3826 virtual table reference (i.e. referenced object-type), any value for this parameter may be
3827 used. If the SELECT clause contains properties from more than one table, then the value of this
3828 parameter MUST be "FALSE".

3829 • **Integer maxItems:** See section 2.2.1.1 Paging.

3830 • **Integer skipCount:** See section 2.2.1.1 Paging.

3831 **2.2.6.1.2 Outputs**

3832 • **<Array> Object QueryResults:** The set of results for the query. (See section 2.1.10 Query.).

3833 Each object result MUST include the following elements if they are requested:

- 3834 ◦ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.

- 3835 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
3836 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
3837 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3838 **Optional:**

- 3839 • **Integer numItems:** See section 2.2.1.1 Paging.

3840

3841 **2.2.6.1.3 Exceptions Thrown & Conditions**

- 3842 • See section 2.2.1.4.1 General Exceptions
3843 • If the select clause includes properties from more than a single type reference, then the repository SHOULD throw an exception if includeRelationships is something other than "none" or includeAllowableActions is specified as TRUE.

3846 **2.2.6.2 getContentChanges**

3847 **Description:** Gets a list of content changes. This service is intended to be used by search crawlers or other applications that need to efficiently understand what has changed in the repository.

3849 **Notes:**

- 3850 • The content stream is NOT returned for any change event.
3851 • The definition of the authority needed to call this service is repository specific.
3852 • The latest change log token for a repository can be acquired via the getRepositoryInfo service.

3853 **2.2.6.2.1 Inputs**

3854 **Required:**

- 3855 • **ID repositoryId:** The identifier for the Repository.

3856 **Optional:**

- 3857 • **String changeLogToken:**
3858 ○ If specified, then the Repository MUST return the change event corresponding to the value of the specified change log token as the first result in the output.
3859 ○ If not specified, then the Repository MUST return the first change event recorded in the change log.

- 3862 • **Boolean includeProperties:**

- 3863 ○ If TRUE, then the Repository MUST include the updated property values for "updated" change events.
3864 ○ If FALSE (default), then the Repository MUST NOT include the updated property values for "updated" change events. The single exception to this is that the objectId MUST always be included.

3868 **Boolean includePolicyIDs:**

- 3869 If TRUE, then the Repository MUST include the IDs of Policies applied to the object referenced in each change event, if the change event modified the set of policies applied to the object.
3870 If FALSE (default), then the Repository will not include policy information.

3872 **Boolean includeACL:** See section 2.2.1.2.5 ACLs.

3873 **Integer maxItems:** See section 2.2.1.1 Paging.

- 3874 **2.2.6.2.2 Outputs**
- 3875 • **<Array> changeEvents:** A collection of CMIS objects that MUST include the information [as specified](#) in 2.1.11.3. Each result MUST include the following elements if they are requested:
- 3876 ◦ **<Array> policyIDs:** The IDs of Policies applied to the object referenced in the change event.
- 3877 ◦ **<Array> ACLs:** The ACLs applied to the object reference in the change event.
- 3878 • **String lastChangeLogToken:** The change log token corresponding to the last change event in changeEvents.
- 3879 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.
- 3880 **Optional:**
- 3881 • **Integer numItems:** See section 2.2.1.1 Paging.
- 3882 **2.2.6.2.3 Exceptions Thrown & Conditions**
- 3883 • See section 2.2.1.4.1 General Exceptions
- 3884 • **constraint:** The Repository MUST throw this exception if the event corresponding to the change log token provided as an input parameter is no longer available in the change log. (E.g. because the change log was truncated).
- 3885 **2.2.7 Versioning Services**
- 3886 The Versioning services (checkOut, cancelCheckOut, getPropertiesOfLatestVersion, getAllVersions, deleteAllVersions) are used to navigate or update a Document Version Series.
- 3887 **2.2.7.1 checkOut**
- 3888 **Description:** Create a private working copy of the document.
- 3889 **2.2.7.1.1 Inputs**
- 3890 **Required:**
- 3891 • **ID repositoryId:** The identifier for the Repository.
- 3892 • **ID objectId:** The identifier of the object.
- 3893 **2.2.7.1.2 Outputs**
- 3894 • **ID objectId:** The identifier for the “Private Working Copy” document.
- 3895 • **Boolean contentCopied:** TRUE if the content-stream of the Private Working Copy is a copy of the contentStream of the Document that was checked out.
- 3896 ◦ FALSE if the content-stream of the Private Working Copy is “not set”.
- 3897 **2.2.7.1.3 Exceptions Thrown & Conditions**
- 3898 • See section 2.2.1.4.1 General Exceptions
- 3899 • **constraint:** The Repository MUST throw this exception if the Document’s Object-Type definition’s *versionable* attribute is FALSE.
- 3900 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3901 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 3902 • **versioning:** The repository MAY throw this exception if the object is a non-current Document Version.

3911 **2.2.7.2 cancelCheckOut**
3912 **Description:** Reverses the effect of a check-out. Removes the private working copy of the checked-out
3913 document, allowing other documents in the version series to be checked out again.

3914 **2.2.7.2.1 Inputs**

3915 **Required:**

- 3916 • **ID repositoryId:** The identifier for the Repository.
3917 • **ID objectId:** The identifier of the document.

3918 **2.2.7.2.2 Exceptions Thrown & Conditions**

- 3919 • See section 2.2.1.4.1 General Exceptions
3920 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type
3921 definition's *versionable* attribute is FALSE.
3922 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
3923 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
3924 Version.

3925 **2.2.7.3 checkIn**

3926 **Description:** Checks-in the Private Working Copy document.

3927 **Notes:**

- 3928 • For repositories that do NOT support the **optional "capabilityPWCUpdatable" capability**, the
3929 **properties** and **contentStream** input parameters MUST be provided on the checkIn method for
3930 updates to happen as part of checkIn.
3931 • Each CMIS protocol bindings MUST specify whether the checkin service MUST always include all
3932 updatable properties, or only those properties whose values are different than the original value
3933 of the object.

3934 **2.2.7.3.1 Inputs**

3935 **Required:**

- 3936 • **ID repositoryId:** The identifier for the Repository.
3937 • **ID objectId:** The identifier of the document.

3938 **Optional:**

- 3939 • **Boolean major:** TRUE (**default**) if the checked-in Document Object MUST be a major version.
3940 ○ FALSE if the checked-in Document Object MUST NOT be a major version.
3941 • **<Array> properties:** The property values that MUST be applied to the checked-in Document
3942 Object.
3943 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the checked-in
3944 Document Object. The method of passing the contentStream to the server and the encoding
3945 mechanism will be specified by each specific binding.
3946 • **String checkinComment:** See section 2.1.9.5 Versioning Properties on Document Objects.
3947 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
3948 object.
3949 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
3950 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.

- 3951 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3952 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
3953 specified.

2.2.7.3.2 Outputs

3955 **ID objectId:** The ID of the checked-in document.

2.2.7.3.3 Exceptions Thrown & Conditions

- 3957 • See section 2.2.1.4.1 General Exceptions
3958 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type
3959 definition's *versionable* attribute is FALSE.
3960 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
3961 • **streamNotSupported:** The Repository MUST throw this exception if the
3962 "contentStreamAllowed" attribute of the Object-Type definition specified by the cmis:objectTypeId
3963 property value is set to "not allowed" and a contentStream input parameter is provided.
3964 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

2.2.7.4 getObjectOfLatestVersion

3966 **Description:** Get a the latest Document object in the Version Series.

2.2.7.4.1 Inputs

3968 **Required:**

- 3969 • **ID repositoryId:** The identifier for the Repository.
3970 • **ID objectId:** The identifier for the Version Series.

3971 **Optional:**

- 3972 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major
3973 version object in the Version Series.
3974 o If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
3975 major) version object in the Version Series.
3976 • **String filter:** See **Error! Reference source not found.**section 2.2.1.2.1 Properties.
3977 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
3978 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
3979 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
3980 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
3981 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

2.2.7.4.2 Outputs

- 3983 • **<Array> Properties:** The list of properties for the object.
3984 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
3985 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
3986 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
3987 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
3988 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

- 3989 **2.2.7.4.3 Exceptions Thrown & Conditions**
- 3990 • See section 2.2.1.4.1 General Exceptions
- 3991 • **filterNotValid**: The Repository MUST throw this exception if this property filter input parameter is not valid.
- 3992 • **objectNotFound**: The Repository MUST throw this exception if the input parameter major is TRUE and the Version Series contains no major versions.
- 3995 **2.2.7.5 getPropertiesOfLatestVersion**
- 3996 **Description:** Get a subset of the properties for the latest Document Object in the Version Series.
- 3997 **2.2.7.5.1 Inputs**
- 3998 **Required:**
- 3999 • **ID repositoryId**: The identifier for the Repository.
- 4000 • **ID objectId**: The identifier for the Version Series.
- 4001 **Optional:**
- 4002 • **Boolean major**: If TRUE, then the Repository MUST return the properties for the latest major version object in the Version Series.
- 4003 ◦ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-major) version object in the Version Series.
- 4006 • **String filter**: See **Error! Reference source not found**.section 2.2.1.2.1 Properties.
- 4007 **2.2.7.5.2 Outputs**
- 4008 **<Array> Properties**: The list of properties for the object.
- 4009 **2.2.7.5.3 Exceptions Thrown & Conditions**
- 4010 • See section 2.2.1.4.1 General Exceptions
- 4011 • **filterNotValid**: The Repository MUST throw this exception if this property filter input parameter is not valid.
- 4013 • **objectNotFound**: The Repository MUST throw this exception if the input parameter major is TRUE and the Version Series contains no major versions.
- 4015 **2.2.7.6 getAllVersions**
- 4016 **Description:** Returns the list of all Document Objects in the specified Version Series, sorted by cmis:creationDate descending.
- 4018 **Notes:**
- 4019 • The result set for this operation MUST include the Private Working Copy, subject to caller's access privileges.
- 4021 **2.2.7.6.1 Inputs**
- 4022 **Required:**
- 4023 • **ID repositoryId**: The identifier for the Repository.
- 4024 • **ID objectId**: The identifier for the Version Series.
- 4025 **Optional:**
- 4026 • **String filter**: See **Error! Reference source not found**.section 2.2.1.2.1 Properties.

- 4027 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4028 **2.2.7.6.2 Outputs**

- 4029 • **<Array> ObjectResults:** A list of Document Objects in the specified Version Series. Each object
4030 result MUST include the following elements if they are requested:
4031 ○ **<Array> Properties:** The list of properties for the object.
4032 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4033

4034 **2.2.7.6.3 Exceptions Thrown & Conditions**

- 4035 • See section 2.2.1.4.1 General Exceptions
4036 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
4037 parameter is not valid.

4038 **2.2.8 Relationship Services**

4039 The Relationship Services (*getObjectRelationships*) are used to retrieve the dependent Relationship
4040 objects associated with an independent object.

4041 **2.2.8.1 getObjectRelationships**

4042 **Description:** Gets all or a subset of relationships associated with an independent object.

4043 **2.2.8.1.1 Inputs**

4044 **Required:**

- 4045 **ID repositoryId:** The identifier for the Repository.
4046 • **ID objectId:** The identifier of the object.

4047

4048 **Optional:**

- 4049 • **Boolean includeSubRelationshipTypes:** If TRUE, then the Repository MUST return all
4050 relationships whose Object-Types are descendant-types of the given object's cmis:objectTypeId
4051 property value.
4052 ○ Default is FALSE
4053 ○ If FALSE, then the Repository MUST only return relationships whose Object-Type is
4054 equivalent to the given object's cmis:objectTypeId property value.
4055 • **Enum relationshipDirection:** An enumeration specifying whether the Repository MUST
4056 return relationships where the specified Object is the source of the relationship, the target of the
4057 relationship, or both. Valid values are:
4058 ○ **source: (default)** The Repository MUST return only relationship objects where the specified
4059 object is the source object.
4060 ○ **target:** The Repository MUST return only relationship objects where the specified object is
4061 the target object.
4062 ○ **either:** The Repository MUST return relationship objects where the specified object is
4063 either the source or the target object.
4064 • **ID typeId:** If specified, then the Repository MUST return only relationships whose Object-Type is
4065 of the type specified (and possibly its descendant-types – see next parameter.)
4066 ○ If not specified, then the repository MUST return Relationship objects of all types.

- 4067 • **Integer maxItems:** See section 2.2.1.1 Paging.
4068 • **Integer skipCount:** See section 2.2.1.1 Paging.
4069 • **String filter:** See section 2.2.1.2.1 Properties.
4070 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4071 **2.2.8.1.2 Outputs**

- 4072 • **<Array> Objects:** A list of the relationship objects. Each object result MUST include the following
4073 elements if they are requested:
4074 ○ **<Array> Properties:** The list of properties for the object.
4075 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4076 **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4077 **Optional:**

- 4078 • **Integer numItems:** See section 2.2.1.1 Paging.

4079

4080 **2.2.8.1.3 Exceptions Thrown & Conditions**

- 4081 • See section 2.2.1.4.1 General Exceptions
4082 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
4083 parameter is not valid.

4084 **2.2.9 Policy Services**

4085 The Policy Services (*applyPolicy*, *removePolicy*, *getAppliedPolicies*) are used to apply or remove a policy
4086 object to a controllablePolicy object.

4087 **2.2.9.1 applyPolicy**

4088 **Description:** Applies a specified policy to an object.

4089 **2.2.9.1.1 Inputs**

4090 **Required:**

- 4091 • **ID repositoryId:** The identifier for the Repository.
4092 • **ID policyId:** The identifier for the Policy to be applied.
4093 • **ID objectId:** The identifier of the object.

4094 **2.2.9.1.2 Exceptions Thrown & Conditions**

4095 See section 2.2.1.4.1 General Exceptions

4096 **constraint :** The Repository MUST throw this exception if the specified object's Object-Type
4097 definition's attribute for *controllablePolicy* is FALSE.

4098 **2.2.9.2 removePolicy**

4099 **Description:** Removes a specified policy from an object.

4100 **2.2.9.2.1 Inputs**

4101 **Required:**

- 4102 • **ID repositoryId:** The identifier for the Repository.
4103 • **ID policyId:** The identifier for the Policy to be removed.
4104 • **ID objectId:** The identifier of the object.

4105 **2.2.9.2.2 Exceptions Thrown & Conditions**

- 4106 • See section 2.2.1.4.1 General Exceptions
4107 • **constraint:** The Repository MUST throw this exception if the specified object's Object-Type
4108 definition's attribute for *controllablePolicy* is FALSE.

4109 **2.2.9.3 getAppliedPolicies**

4110 **Description:** Gets the list of policies currently applied to the specified object.

4111 **2.2.9.3.1 Inputs**

4112 **Required:**

- 4113 • **ID repositoryId:** The identifier for the Repository.
4114 • **ID objectId:** The identifier of the object.

4115 **Optional:**

4116 **String filter:** See Error! Reference source not found.section 2.2.1.2.1 Properties.

4117 **2.2.9.3.2 Outputs**

4118 **<Array> Objects:** A list of Policy Objects.

4119 **2.2.9.3.3 Exceptions Thrown & Conditions**

- 4120 • See section 2.2.1.4.1 General Exceptions
4121 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
4122 parameter is not valid.

4123 **2.2.10 ACL Services**

4124 **2.2.10.1 getACL**

4125 **Description:** Get the ACL currently applied to the specified document or folder object.

4126 **2.2.10.1.1 Inputs**

4127 **Required:**

- 4128 • **ID repositoryId:** The identifier for the repository.
4129 • **ID objectId:** The identifier for the object

4130 **Optional:**

- 4131 • **Boolean onlyBasicPermissions:** See section 2.8 Access Control. The repository SHOULD
4132 make a best effort to fully express the native security applied to the object
4133 ○ **TRUE:** (default value if not provided) indicates that the client requests that the returned
4134 ACL be expressed using only the CMIS Basic permissions.
4135 ○ **FALSE:** indicates that the server may respond using either solely CMIS Basic
4136 permissions, or repository specific permissions or some combination of both.

- 4137 **2.2.10.1.2 Outputs**
- 4138 • **<Array> AccessControlEntryType:** The list of access control entries of the ACL for the object.
- 4139 **Optional:**
- 4140 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
4141 i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.
- 4142 **2.2.10.1.3 Exceptions Thrown & Conditions**
- 4143 • See section 2.2.1.4.1 General Exceptions
- 4144 **2.2.10.1.4 Notes**
- 4145 This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=discover* or
4146 *=manage*.
- 4147 How an ACL for the object is computed is up to the repository. A client MUST NOT assume that the ACEs
4148 from the ACL as returned by this service can be applied via *applyACL*.
- 4149 **2.2.10.2 applyACL**
- 4150 **Description:** Adds or removes the given ACEs to or from the ACL of document or folder object.
- 4151 **2.2.10.2.1 Inputs**
- 4152 **Required:**
- 4153 • **ID repositoryId:** The identifier for the repository.
- 4154 • **ID objectId:** The identifier for the object
- 4155 **Optional:**
- 4156 • **<Array> AccessControlEntryType addACEs:** The ACEs to be added.
- 4157 • **<Array> AccessControlEntryType removeACEs:** The ACEs to be removed.
- 4158 • **Enum ACLPropagation:** Specifies how ACEs should be handled:
- 4159 ◦ **objectonly:** ACEs must be applied without changing the ACLs of other objects.
- 4160 ◦ **propagate:** ACEs must be applied by propagate the changes to all “inheriting” objects.
- 4161 ◦ **repositorydetermined:** **Default value.** Indicates that the client leaves the behavior to
4162 the repository.
- 4163 **2.2.10.2.2 Outputs**
- 4164 • **<Array> AccessControlEntryType:** The list of access control entries of the resulting ACL for the
4165 object
- 4166 **Optional:**
- 4167 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
4168 i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.
- 4169 • **String changeToken:** See section 2.2.1.3 Change Tokens.
- 4170 **2.2.10.2.3 Exceptions Thrown & Conditions**
- 4171 • See section 2.2.1.4.1 General Exceptions
- 4172 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
4173 met:
- 4174 ◦ The specified object’s Object-Type definition’s attribute for *controllableACL* is FALSE.

- 4175 ○ The value for *ACLPropagation* does not match the values as returned via
4176 *getACLCapabilities*.
4177 ○ At least one of the specified values for *permission* in ANY of the *ACEs* does not match
4178 ANY of the *permissionNames* as returned by *getACLCapability* and is not a CMIS Basic
4179 *permission*

4180 **2.2.10.2.4 Notes**

4181 This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=manage*.
4182 How ACEs are added or removed to or from the object is up to the repository – with respect to the
4183 *ACLPropagation* provided by the client. For “shared” ACEs (e.g. via inheritance), the repository MAY
4184 merge the ACEs provided with the ACEs of the ACL already applied to the object (i.e. the ACEs provided
4185 MAY not be completely added or removed from the effective ACL for the object).

4186
4187

4188 3 Restful AtomPub Binding

4189 3.1 Overview

4190 This binding is based upon the Atom (RFC4287) and Atom Publishing Protocol (RFC5023).
4191 Implementations of CMIS MUST be compliant with RFC4287 and RFC5023.

4192

4193 In this binding, the client interacts with the repository by acquiring the service document. The client will
4194 request the service document by the URI provided by the vendor. The client will then choose a CMIS
4195 collection, and then start accessing the repository by following the references in the returned documents.

4196

4197 This binding consists of a service document specifying at least CMIS service collections, atom collections,
4198 feeds and entry documents. CMIS extends the Atom and AtomPub documents utilizing the Atom and
4199 AtomPub extension mechanism. CMIS also leverages link tags to specify additional resources related to
4200 the requested resource.

4201

4202 When requesting a resource, optional parameters may be specified to change default behavior via query
4203 parameters.

4204 3.1.1 Namespaces

4205 This specification uses the following namespaces and prefixes when referring to xml or xml schema
4206 elements in the text or examples:

- 4207 • CMIS-Core: <http://docs.oasis-open.org/ns/cmis/core/200908/>
 - 4208 ○ Prefix: cmis
- 4209 • CMIS-RestAtom: <http://docs.oasis-open.org/ns/cmis/restatom/200908/>
 - 4210 ○ Prefix: cmisra
- 4211 • Atom : <http://www.w3.org/2005/Atom>
 - 4212 ○ Prefix: atom
- 4213 • AtomPub: <http://www.w3.org/2007/app>
 - 4214 ○ Prefix: app

4215

4216 3.1.2 Authentication

4217 Authentication SHOULD be handled by the transport protocol. Please see AtomPub (RFC5023) section
4218 14.

4219

4220 3.1.3 Response Formats

4221 The client can specify, in HTTP the Accept header, which formats are acceptable to the client. With this
4222 mechanism the client can chose which response format the CMIS implementation should respond with.
4223 The CMIS compliant implementation MUST support the appropriate Media Types specified in this
4224 document.

4225 **3.1.4 Optional Arguments**

4226 The binding supports adding optional parameters to CMIS resources to modify the default behavior.

4227 CMIS implementations MUST support arguments being specified as HTTP query string parameters.

4228 Names and valid values for HTTP query string parameters are as described in the appropriate CMIS

4229 Service descriptions [see CMIS Domain Model]. Valid values of enumeration types are also represented

4230 in the CMIS Core XML Schema

4231 **3.1.5 Errors and Exceptions**

4232 Exceptions MUST be mapped to the appropriate HTTP status code.

4233 Repositories SHOULD provide sufficient information in the body of the HTTP response for a user to

4234 determine corrective action.

4235 See Section 3.2.4 HTTP Status Codes for more information.

4236 **3.1.6 Renditions**

4237 Each Rendition included in a CMIS AtomPub response is represented as an Atom link with relationship

4238 alternate.

4239

4240 The following attributes SHOULD be included on the link element:

- href: URI to the rendition content stream
- type: The Media Type of the Rendition
- cmisra:renditionKind: The Rendition Kind for the Rendition

4244

4245 The following attributes MAY be included

4246 • title: The Filename (or name property if object) of Rendition

4247 • length: The length of the rendition

4248 **3.1.7 Content Streams**

4249 The content stream for a document SHOULD be referenced by the content src attribute as well as the

4250 edit-media link relation.

4251 A CMIS Repository MAY use different URIs for both content src attribute and the edit-media link relation

4252 for the same content stream.

4253 The following attributes SHOULD be included on the link element:

4254 • href: URI to the content stream

4255 • type: The Media Type of the content stream

4256 **3.1.8 Paging of Feeds**

4257 For paging, please see the AtomPub RFC. CMIS leverages first, next, previous, and last link relations to

4258 express paging.

4259 If the repository can include the number of items (numItems in CMIS Domain Model) in a feed, then the

4260 repository SHOULD include the cmisra:numItems extension element in the feed.

4261 **3.1.9 Services not Exposed**

4262 The following services are not exposed in this binding:

4263 • getRenditions: This is exposed as part of getObject

4264 • getProperties: This is exposed as part of getObject

- 4265 • createDocumentFromSource: This is not exposed in this binding except as the client saving the
4266 resource and resubmitting it without the cmis:objectId.
4267
4268

4269 **3.2 HTTP**

4270 **3.2.1 Entity Tag**

4271 CMIS changeTokens are represented as Entity Tags and follow HTTP's use of Entity Tags. CMIS server
4272 implementations SHOULD support Entity Tags. ChangeTokens are also provided as properties and
4273 SHOULD be provided when the object is included inside an atom entry or feed.

4274 **3.2.2 HTTP Range**

4275 Repositories MAY support HTTP Range requests on Content Streams.

4276 **3.2.3 HTTP OPTIONS Method**

4277 The repository MAY support the HTTP OPTIONS method on all the resources defined in this
4278 specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP
4279 methods specified for that resource in the Allow header.

4280 **3.2.4 HTTP Status Codes**

4281 **3.2.4.1 Common CMIS Exceptions**

4282 The following listing defines the HTTP status codes that repositories MUST return for the various common
4283 exceptions defined in CMIS Domain Model.

CMIS Services Exception	HTTP Status Code
invalidArgument	400
objectNotFound	404
permissionDenied	403
operationNotSupported	405
updateConflict	409
runtime	500

4291 **3.2.4.2 Other Exceptions**

CMIS Services Exception	HTTP Status Code
constraint	409
filterNotValid	400
streamNotSupported	403
storage	500
contentAlreadyExists	409
versioning	409

4299 **3.2.4.3 Notable HTTP Status Codes**

- 4300 • 415 Unsupported Media Type
4301 o When a document is POST'ed to a collection that does not support the media type of the
4302 document, this status code MUST be returned

- 4303 • 422 Unprocessable Entity (Defined in RFC4918 Section 11.2)
4304 ○ When a request has been POST'ed but cannot be processed, this status code MUST be
4305 returned

4306
4307 Please see RFC2616 Section 10 for more information.
4308

4309 **3.3 Media Types**

4310 CMIS introduces new media types for:

- 4311 • a CMIS Query document (application/cmisquery+xml)
4312 • a CMIS AllowableActions document (application/cmisallowableactions+xml)
4313 • an Atom Document (Entry or Feed) with any CMIS Markup (application/cmisatom+xml)
4314 • an Atom Feed Document with CMIS Hierarchy extensions (application/cmistree+xml)

4315
4316 In addition to those media types specified by CMIS, CMIS also leverages these media types:
4317 • AtomPub Service (application/atomsvc+xml)
4318 • Atom Entry (application/atom+xml;type=entry)
4319 • Atom Feed (application/atom+xml;type=feed)

4320 **3.3.1 CMIS Atom**

4321 Media Type: application/cmisatom+xml

4322 Starting tag: atom:feed or atom:entry

4323 Type Parameters:

- 4324 • type – the semantics of the type parameter MUST be the same as the media type parameter for
4325 atom documents.

4326
4327 This allows clients to differentiate between repositories that require atom media type with CMIS
4328 extensions (application/cmisatom+xml) for creation and repositories that allow generic atom media type
4329 without CMIS extensions (application/atom+xml).

4330
4331 This is only used for CMIS repositories to advertise what media types are accepted for adding to a
4332 collection (e.g., creating resources in a collection).

4333
4334 All feeds and entries from a CMIS repository MUST utilize the atom media type for exposing Atom
4335 resources. Please see the individual resources for more information on the media type. This provides
4336 the interoperability with Atom clients.

4337
4338 Example:

4339
4340 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4341 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4342 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4343 xmlns:atom="http://www.w3.org/2005/Atom"
4344 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4345 open.org/ns/cmis/restatom/200908/">
4346 <atom:author>
4347 <atom:name>Al Brown</atom:name>

```

4348     </atom:author>
4349     <atom:id>urn:uuid:5f716462-7b6b-4a0d-bd1d-34677d5514d7</atom:id>
4350     <atom:title type="text">CMIS Example Document</atom:title>
4351     <atom:updated>2009-10-19T10:09:59.296-07:00</atom:updated>
4352     <atom:content type="text">some text</atom:content>
4353     <cmisra:object>
4354         <cmis:properties>
4355             <cmis:propertyId localName="rep-cmis:objectTypeId"
4356             propertyDefinitionId="cmis:objectTypeId">
4357                 <cmis:value>invoice</cmis:value>
4358                 </cmis:propertyId>
4359                 <cmis:propertyString localName="rep-cmis:name"
4360             propertyDefinitionId="cmis:name">
4361                 <cmis:value>CMIS Example Document</cmis:value>
4362                 </cmis:propertyString>
4363             </cmis:properties>
4364         </cmisra:object>
4365     </atom:entry>
4366

```

4367 **3.3.2 CMIS Query**

4368 Media Type: application/cmisquery+xml

4369 Starting tag: cmis:query

4370

4371 This document contains the representation of a query to be executed in a CMIS repository.

4372

4373 Example:

```

4374 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4375 <cmis:query xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4376   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4377   xmlns:atom="http://www.w3.org/2005/Atom"
4378   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4379   open.org/ns/cmis/restatom/200908/">
4380     <cmis:statement>SELECT * FROM cmis:document</cmis:statement>
4381     <cmis:searchAllVersions>true</cmis:searchAllVersions>
4382     <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
4383     <cmis:includeRelationships>none</cmis:includeRelationships>
4384     <cmis:renditionFilter>*</cmis:renditionFilter>
4385     <cmis:maxItems>50</cmis:maxItems>
4386     <cmis:skipCount>0</cmis:skipCount>
4387 </cmis:query>

```

4388

4389 Please also see the example documents included with the schema.

4390

4391 **3.3.3 CMIS Allowable Actions**

4392 Media Type: application/cmisallowableactions+xml

4393 Starting tag: cmis:allowableactions

4394

4395 This document contains the representation of the allowable actions the user may perform on the
4396 referenced object.

4397

4398 Example:

```

4399 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

4400 <cmis:allowableActions xmlns:cmis="http://docs.oasis-
4401 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
4402 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
4403 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4404 open.org/ns/cmis/restatom/200908/">
4405     <cmis:canDeleteObject>true</cmis:canDeleteObject>
4406     <cmis:canUpdateProperties>true</cmis:canUpdateProperties>
4407     <cmis:canGetProperties>true</cmis:canGetProperties>
4408     <cmis:canGetObjectRelationships>true</cmis:canGetObjectRelationships>
4409     <cmis:canGetObjectParents>true</cmis:canGetObjectParents>
4410     <cmis:canMoveObject>true</cmis:canMoveObject>
4411     <cmis:canDeleteContentStream>true</cmis:canDeleteContentStream>
4412     <cmis:canCheckOut>true</cmis:canCheckOut>
4413     <cmis:canCancelCheckOut>true</cmis:canCancelCheckOut>
4414     <cmis:canCheckIn>true</cmis:canCheckIn>
4415     <cmis:canSetContentStream>true</cmis:canSetContentStream>
4416     <cmis:canGetAllVersions>true</cmis:canGetAllVersions>
4417     <cmis:canAddObjectToFolder>true</cmis:canAddObjectToFolder>
4418     <cmis:canRemoveObjectFromFolder>true</cmis:canRemoveObjectFromFolder>
4419     <cmis:canGetContentStream>true</cmis:canGetContentStream>
4420     <cmis:canApplyPolicy>true</cmis:canApplyPolicy>
4421     <cmis:canGetAppliedPolicies>true</cmis:canGetAppliedPolicies>
4422     <cmis:canRemovePolicy>true</cmis:canRemovePolicy>
4423     <cmis:canCreateDocument>true</cmis:canCreateDocument>
4424 </cmis:allowableActions>

```

4425

4426 Please also see the example documents included with the schema.

4427

4428 3.3.4 CMIS Tree

4429 Media Type: application/cmistree+xml

4430 Starting tag: atom:feed

4431

4432 This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.

4433

4434 Please see Section 3.3.2.1 for more information.

4435

4436 Example:

```

4437 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4438 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4439   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4440   xmlns:atom="http://www.w3.org/2005/Atom"
4441   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4442     open.org/ns/cmis/restatom/200908/">
4443     <atom:title type="text">Feed for folder1</atom:title>
4444     <atom:author>
4445       <atom:name>Al Brown</atom:name>
4446       <atom:uri>http://www.ibm.com/</atom:uri>
4447       <atom:email>albertcbrown@us.ibm.com</atom:email>
4448     </atom:author>
4449     <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4450     <atom:id>urn:uuid:08d13325-cba1-4290-95e3-f485068b08e8</atom:id>
4451     <atom:link type="application/atom+xml;type=feed" rel="self"
4452       href="http://cmisexample.oasis-open.org/repl/4f942e30-895c-4c82-9b31-
4453         58f639b6c2d7/3"/>
4454     <atom:link type="application/atomsvc+xml" rel="service"
4455       href="http://cmisexample.oasis-open.org/repl/service"/>

```

```

4456      <atom:link type="application/atom+xml;type=entry" rel="via"
4457      href="http://cmisexample.oasis-open.org/rep1/4f942e30-895c-4c82-9b31-
4458      58f639b6c2d7"/>
4459          <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
4460          open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
4461          open.org/rep1/4f942e30-895c-4c82-9b31-58f639b6c2d7/foldertree"/>
4462              <atom:link type="application/atom+xml;type=feed" rel="down"
4463              href="http://cmisexample.oasis-open.org/rep1/4f942e30-895c-4c82-9b31-
4464              58f639b6c2d7/children"/>
4465                  <atom:link type="application/atom+xml;type=entry" rel="up"
4466                  href="http://cmisexample.oasis-open.org/rep1/b8803070-0253-4160-9ad6-
4467                  26ad7adb561"/>
4468                      <atom:entry>
4469                          <atom:author>
4470                              <atom:name>Al Brown</atom:name>
4471                              <atom:uri>http://www.ibm.com/</atom:uri>
4472                              <atom:email>albertcbrown@us.ibm.com</atom:email>
4473                          </atom:author>
4474                          <atom:content src="http://cmisexample.oasis-open.org/rep1/962b4fb8-
4475                          aefa-4d0c-8562-a84e297e8107"/>
4476                              <atom:id>urn:uuid:962b4fb8-aefa-4d0c-8562-a84e297e8107</atom:id>
4477                              <atom:title type="text">CMIS Example Folder as Customer
4478                          type</atom:title>
4479                              <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4480                              <atom:link rel="self" href="http://cmisexample.oasis-
4481          open.org/rep1/962b4fb8-aefa-4d0c-8562-a84e297e8107"/>
4482                                  <atom:link rel="edit" href="http://cmisexample.oasis-
4483          open.org/rep1/962b4fb8-aefa-4d0c-8562-a84e297e8107"/>
4484                                      <atom:link type="application/cmis+xml;type=allowableActions"
4485          rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
4486          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4487          a84e297e8107/allowableactions"/>
4488              <atom:link type="application/atom+xml;type=entry" rel="describedby"
4489          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4490          a84e297e8107/type"/>
4491                  <atom:link type="application/atomsvc+xml" rel="service"
4492          href="http://cmisexample.oasis-open.org/rep1/service"/>
4493                      <atom:published>2009-10-19T10:09:59.515-07:00</atom:published>
4494                      <atom:summary type="html">HTML summary of Entry 962b4fb8-aefa-4d0c-
4495          8562-a84e297e8107</atom:summary>
4496                          <atom:link type="application/atom+xml;type=entry" rel="up"
4497          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4498          a84e297e8107/up"/>
4499                              <atom:link type="application/atom+xml;type=feed" rel="down"
4500          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4501          a84e297e8107/children"/>
4502                                  <atom:link type="application/cmistree+xml" rel="down"
4503          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4504          a84e297e8107/tree"/>
4505                                      <atom:link type="application/atom+xml;type=feed"
4506          rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
4507          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4508          a84e297e8107/foldertree"/>
4509              <atom:link type="application/atom+xml;type=feed"
4510          rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
4511          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4512          a84e297e8107/relationships"/>
4513                  <atom:link type="application/atom+xml;type=feed"
4514          rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
4515          href="http://cmisexample.oasis-open.org/rep1/962b4fb8-aefa-4d0c-8562-
4516          a84e297e8107/policies"/>
4517                      <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
4518          open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
4519          open.org/rep1/962b4fb8-aefa-4d0c-8562-a84e297e8107/acl"/>
```

```

4520      <cmisra:object>
4521          <cmis:properties>
4522              <cmis:propertyId localName="rep-cmis:objectId"
4523      propertyDefinitionId="cmis:objectId">
4524          <cmis:value>962b4fb8-aefa-4d0c-8562-
4525      a84e297e8107</cmis:value>
4526          </cmis:propertyId>
4527      </cmis:properties>
4528  </cmisra:object>
4529      <cmisra:pathSegment>customer</cmisra:pathSegment>
4530      <cmisra:children>
4531          <atom:title type="text">CMIS Example Folder as Customer
4532      type</atom:title>
4533          <atom:author>
4534              <atom:name>Al Brown</atom:name>
4535              <atom:uri>http://www.ibm.com/</atom:uri>
4536              <atom:email>albertcbrown@us.ibm.com</atom:email>
4537          </atom:author>
4538          <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4539          <atom:id>urn:uuid:084edab9-418a-42b9-813f-cf450418181d</atom:id>
4540          <atom:link type="application/atom+xml;type=feed" rel="self"
4541 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aefa-4d0c-8562-
4542 a84e297e8107/3"/>
4543          <atom:link type="application/atomsvc+xml" rel="service"
4544 href="http://cmisexample.oasis-open.org/repl/service"/>
4545          <atom:link type="application/atom+xml;type=entry" rel="via"
4546 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aefa-4d0c-8562-
4547 a84e297e8107"/>
4548          <atom:link type="application/atom+xml;type=feed"
4549 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
4550 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aefa-4d0c-8562-
4551 a84e297e8107/foldertree"/>
4552          <atom:link type="application/atom+xml;type=feed" rel="down"
4553 href="http://cmisexample.oasis-open.org/repl/962b4fb8-aefa-4d0c-8562-
4554 a84e297e8107/children"/>
4555          <atom:link type="application/atom+xml;type=entry" rel="up"
4556 href="http://cmisexample.oasis-open.org/repl/4f942e30-895c-4c82-9b31-
4557 58f639b6c2d7"/>
4558          <atom:entry>
4559              <atom:author>
4560                  <atom:name>Al Brown</atom:name>
4561                  <atom:uri>http://www.ibm.com/</atom:uri>
4562                  <atom:email>albertcbrown@us.ibm.com</atom:email>
4563              </atom:author>
4564              <atom:content src="http://cmisexample.oasis-
4565 open.org/repl/fba295fd-c8ed-40d1-8ddc-93671a9c89a5"/>
4566              <atom:id>urn:uuid:fba295fd-c8ed-40d1-8ddc-
4567 93671a9c89a5</atom:id>
4568              <atom:title type="text">CMIS Example Doc as Invoice
4569      type</atom:title>
4570              <atom:updated>2009-10-19T10:09:59.515-07:00</atom:updated>
4571              <atom:link rel="self" href="http://cmisexample.oasis-
4572 open.org/repl/fba295fd-c8ed-40d1-8ddc-93671a9c89a5"/>
4573              <atom:link rel="edit" href="http://cmisexample.oasis-
4574 open.org/repl/fba295fd-c8ed-40d1-8ddc-93671a9c89a5"/>
4575              <atom:link type="application/cmis+xml;type=allowableActions"
4576 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
4577 href="http://cmisexample.oasis-open.org/repl/fba295fd-c8ed-40d1-8ddc-
4578 93671a9c89a5/allowableactions"/>
4579              <atom:link type="application/atom+xml;type=entry"
4580 rel="describedby" href="http://cmisexample.oasis-open.org/repl/fba295fd-c8ed-
4581 40d1-8ddc-93671a9c89a5/type"/>
4582              <atom:link type="application/atomsvc+xml" rel="service"
4583 href="http://cmisexample.oasis-open.org/repl/service"/>

```

```

4584             <atom:published>2009-10-19T10:09:59.515-07:00</atom:published>
4585             <atom:summary type="html">HTML summary of Entry fba295fd-
4586 c8ed-40d1-8ddc-93671a9c89a5</atom:summary>
4587             <atom:link rel="edit-media" href="http://cmisexample.oasis-
4588 open.org/rep1/fba295fd-c8ed-40d1-8ddc-93671a9c89a5/edit-media"/>
4589             <atom:link rel="alternate" href="http://cmisexample.oasis-
4590 open.org/rep1/fba295fd-c8ed-40d1-8ddc-93671a9c89a5/alternate"/>
4591             <atom:link type="application/atom+xml;type=feed" rel="up"
4592 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4593 93671a9c89a5/parents"/>
4594             <atom:link type="application/atom+xml;type=feed" rel="version-
4595 history" href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4596 93671a9c89a5/allversions"/>
4597             <atom:link type="application/atom+xml;type=entry"
4598 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/fba295fd-
4599 c8ed-40d1-8ddc-93671a9c89a5/latest"/>
4600             <atom:link type="application/atom+xml;type=feed"
4601 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
4602 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4603 93671a9c89a5/relationships"/>
4604             <atom:link type="application/atom+xml;type=feed"
4605 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
4606 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4607 93671a9c89a5/policies"/>
4608             <atom:link type="application/cmisacl+xml"
4609 rel="http://docs.oasis-open.org/ns/cmis/link/200908/acl"
4610 href="http://cmisexample.oasis-open.org/rep1/fba295fd-c8ed-40d1-8ddc-
4611 93671a9c89a5/acl"/>
4612             <cmisra:object>
4613                 <cmis:properties>
4614                     <cmis:propertyId localName="rep-cmis:objectId"
4615 propertyDefinitionId="cmis:objectId">
4616                         <cmis:value>fba295fd-c8ed-40d1-8ddc-
4617 93671a9c89a5</cmis:value>
4618                     </cmis:propertyId>
4619                 </cmis:properties>
4620             </cmisra:object>
4621             <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
4622         </atom:entry>
4623     </cmisra:children>
4624 </atom:entry>
4625 <atom:entry>
4626     <atom:author>
4627         <atom:name>Al Brown</atom:name>
4628         <atom:uri>http://www.ibm.com/</atom:uri>
4629         <atom:email>albertcbrown@us.ibm.com</atom:email>
4630     </atom:author>
4631     <atom:content src="http://cmisexample.oasis-open.org/rep1/affc2158-
4632 6c8f-4245-a045-688225f8c2ad"/>
4633         <atom:id>urn:uuid:affc2158-6c8f-4245-a045-688225f8c2ad</atom:id>
4634         <atom:title type="text">CMIS Example Doc as Invoice type</atom:title>
4635         <atom:updated>2009-10-19T10:09:59.531-07:00</atom:updated>
4636         <atom:link rel="self" href="http://cmisexample.oasis-
4637 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad"/>
4638         <atom:link rel="edit" href="http://cmisexample.oasis-
4639 open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad"/>
4640         <atom:link type="application/cmis+xml;type=allowableActions"
4641 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
4642 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4643 688225f8c2ad/allowableactions"/>
4644         <atom:link type="application/atom+xml;type=entry" rel="describedby"
4645 href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4646 688225f8c2ad/type"/>

```

```

4647      <atom:link type="application/atomsvc+xml" rel="service"
4648      href="http://cmisexample.oasis-open.org/rep1/service"/>
4649          <atom:published>2009-10-19T10:09:59.531-07:00</atom:published>
4650              <atom:summary type="html">HTML summary of Entry affc2158-6c8f-4245-
4651      a045-688225f8c2ad</atom:summary>
4652          <atom:link rel="edit-media" href="http://cmisexample.oasis-
4653      open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad/edit-media"/>
4654              <atom:link rel="alternate" href="http://cmisexample.oasis-
4655      open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad/alternate"/>
4656          <atom:link type="application/atom+xml;type=feed" rel="up"
4657      href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4658      688225f8c2ad/parents"/>
4659          <atom:link type="application/atom+xml;type=feed" rel="version-history"
4660      href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4661      688225f8c2ad/allversions"/>
4662              <atom:link type="application/atom+xml;type=entry" rel="current-
4663      version" href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4664      688225f8c2ad/latest"/>
4665                  <atom:link type="application/atom+xml;type=feed"
4666      rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
4667      href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4668      688225f8c2ad/relationships"/>
4669                      <atom:link type="application/atom+xml;type=feed"
4670      rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
4671      href="http://cmisexample.oasis-open.org/rep1/affc2158-6c8f-4245-a045-
4672      688225f8c2ad/policies"/>
4673                          <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
4674      open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
4675      open.org/rep1/affc2158-6c8f-4245-a045-688225f8c2ad/acl"/>
4676                              <cmisra:object>
4677                                  <cmis:properties>
4678                                      <cmis:propertyId localName="rep-cmis:objectId"
4679      propertyDefinitionId="cmis:objectId">
4680                                          <cmis:value>affc2158-6c8f-4245-a045-
4681      688225f8c2ad</cmis:value>
4682                                              </cmis:propertyId>
4683                                  </cmis:properties>
4684                              </cmisra:object>
4685                                  <cmisra:pathSegment>invoice3.pdf</cmisra:pathSegment>
4686                              </atom:entry>
4687                      </atom:feed>

```

4688

4689 Note: This media type is used on links with relation down (see section 3.2.3.2 Hierarchy Navigation
4690 Internet Draft Link Relations). When the individual resources are returned by the CMIS repository they
4691 will use the atom media type (application/atom+xml)

4692

4693 Please also see the example documents included with the schema.

4694

4695 3.3.5 CMIS ACL

4696 Media Type: application/cmisacl+xml

4697 Starting tag: cmis:acl

4698

4699 This document specifies an Access Control List based on the schema in CMIS Domain Model.

4700

4701 Example:

```

4702      <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
4703 <cmis:acl xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"  
4704   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"  
4705   xmlns:atom="http://www.w3.org/2005/Atom"  
4706   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
4707     open.org/ns/cmis/restatom/200908/">  
4708     <cmis:permission>  
4709       <cmis:principal>  
4710         <cmis:principalId>Al Brown</cmis:principalId>  
4711       </cmis:principal>  
4712       <cmis:permission>cmis:read</cmis:permission>  
4713       <cmis:permission>cmis:write</cmis:permission>  
4714       <cmis:permission>cmis:all</cmis:permission>  
4715       <cmis:permission>publish</cmis:permission>  
4716       <cmis:direct>true</cmis:direct>  
4717     </cmis:permission>  
4718   </cmis:acl>
```

4719

4720 Please also see the example documents included with the schema.

4721

4722 **3.4 Atom Extensions for CMIS**

4723 **3.4.1 Atom Element Extensions**

4724 **3.4.1.1 AtomPub Workspace**

4725 **3.4.1.1.1 cmisra:collectionType**

4726 This element is included inside the app:collection element. This specifies the cmis collection type.

4727 **3.4.1.1.2 cmisra:repositoryInfo**

4728 This element is included inside the app:workspace element. This specifies information about the CMIS
4729 repository.

4730 **3.4.1.1.3 cmis:uritemplate**

4731 This element is included inside the app:workspace element. This specifies information about URI
4732 templates

4733 **3.4.1.2 Atom Feed**

4734 **3.4.1.2.1 cmisra:numItems**

4735 This element is included inside the atom:feed element. This specifies the number of items in the feed.

4736 **3.4.1.3 Atom Entry**

4737 **3.4.1.3.1 cmisra:children**

4738 This element is included inside the atom:entry element. This includes the children of the atom entry.
4739

4740 **3.4.1.3.2 cmisra:object**

4741 This element is included inside the atom:entry element for CMIS Document, Folder, Relationship and
4742 Policy objects. This specifies the CMIS object information for the atom entry.

4743

4744 **3.4.1.3.3 cmisra:pathSegment**

4745 This element is included inside the atom:entry element for CMIS Type Definitions that are filable. This
4746 specifies the pathSegment for this object in the folder representing the feed.

4747

4748 **3.4.1.3.4 cmisra:relativePathSegment**

4749 This element is included inside the atom:entry element. This specifies the relative pathSegment for the
4750 object in that particular folder. This MUST be used only inside an object parents feed.

4751 **3.4.1.3.5 cmisra:type**

4752 This element is included inside the atom:entry element for CMIS Type Definitions. This specifies the type
4753 definition the atom entry represents.

4754 **3.4.1.3.6 cmisra:content**

4755 This element specifies the content of the atom:entry element. The content is base64 encoded.
4756 This element MUST take precedence over atom:content on submission of an atom entry to a repository.
4757 A repository MUST use the atom:content element to return back to the client the content of the document.
4758 This is required when the client has an XML document stored that is not well formed and thus would not
4759 be able to be included inside atom:content element.

4760 **3.4.2 Attributes**

4761 These attributes are in the CMIS RestAtom namespace (cmisra).

4762 **3.4.2.1 cmisra:id**

4763 This attribute is used on the atom:link element to specify the cmis id of the resource. This attribute
4764 SHOULD be on all link relations that point to a CMIS object.

4765

4766 This attribute MAY also be on cmisra:type. The value of the attribute on cmis:type MUST be the same as
4767 the type definition id.

4768

4769 Example:

```
4770 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4771 <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/" 
4772   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/" 
4773   xmlns:atom="http://www.w3.org/2005/Atom" 
4774   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4775   open.org/ns/cmis/restatom/200908/" type="application/atom+xml;type=feed"
4776   rel="down" href="http://cmisexample.oasis-open.org/rep1//children/ba2cb793-
4777   6d6d-48cd-b6c2-62dcddedc4b5/1" cmisra:id="ba2cb793-6d6d-48cd-b6c2-
4778   62dcddedc4b5" />
```

4779

4780 Please also see the example documents included with the schema.

4781

4782 **3.4.2.2 cmisra:renditionKind**

4783 This attribute is used on the atom:link element with relation alternate to specify the renditionKind of the
4784 resource. This attribute SHOULD be on all link elements with relation alternate that are a CMIS rendition.

4785

4786 Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
  xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
  open.org/ns/cmis/restatom/200908/" type="text/html" rel="alternate"
  href="http://cmisexample.oasis-open.org/repl1/rendition/ba2cb793-6d6d-48cd-
  b6c2-62dcddedc4b5/1" cmisra:renditionKind="cmis:thumbnail"/>
```

4795

4796 Please also see the example documents included with the schema.

4797 **3.4.3 CMIS Link Relations**

4798 The listing below outlines the different link relation types in CMIS. This is in addition to the link relations
4799 specified by Atom and Atom Publishing Protocol. The registry for link relations is located at
4800 <http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

4801

4802 The link element with a specified relation MUST be included if client can perform the operation. The
4803 repository SHOULD omit the link relation if the operation is not available. The operation may not be
4804 available due to a variety of reasons such as access control, administrative policies, or other
4805 mechanisms.

4806

4807 Links may have the following attribute in addition to the ones specified by Atom and Atom Publishing
4808 Protocol:

- 4809 • (CMIS) id: Specifies the CMIS ID of the resource referenced by the link. Repositories SHOULD
4810 include this attribute for elements such as atom:link that point to CMIS resources that have an id.

4811

4812 These are the link relation types specified by CMIS:

4813 **3.4.3.1 Existing Link Relations**

4814 Existing link relations should be used where appropriate by the implementation. In addition, the following
4815 link relations are leveraged for the CMIS specification:

4816

- self
 - 4817 ○ This link relation provides the URI to retrieve this resource again.
 - 4818 ○ Service: The appropriate service that generated the atom entry or feed.
 - 4819 ○ Resources: All except AllowableActions, ACL and Content Streams
- 4820 • service
 - 4821 ○ The service link relation when provided on a CMIS resource MUST point to an AtomPub
4822 service document with only one workspace element. This workspace element MUST
4823 represent the repository containing that resource.
 - 4824 ○ Media Type: application/atomsvc+xml
 - 4825 ○ Resources: All except AllowableActions, ACL and Content Streams
- 4826 • describedby

- 4827 o When used on a CMIS resource, this link relation MUST point to an atom entry that
4828 describes the type of that resource.
- 4829 o Service: getTypeDefinition on specified object
- 4830 o Media Type: application/atom+xml;type=entry
- 4831 o Resources: CMIS Document, CMIS Folder, CMIS Relationship, CMIS Policy objects and
4832 CMIS Types
- 4833 • via
 - 4834 o When used on an Atom Feed document, this link relation MUST point to the atom entry
4835 representing the CMIS resource from whom this feed is derived.
 - 4836 o Media Type: application/atom+xml;type=entry
 - 4837 o Resources: All CMIS Feeds and Collections
- 4838 • edit-media
 - 4839 o When used on a CMIS document resource, this link relation MUST point to the URI for
4840 content stream of the CMIS document. This URI MUST be used to set or delete the
4841 content stream. This URI MAY be used to retrieve the content stream for the document.
 - 4842 o Service: setContentStream (PUT) , deleteContentStream (DELETE)
 - 4843 o Media Type: Specific to resource
 - 4844 o Resources: CMIS Document
- 4845 • edit
 - 4846 o When used on a CMIS resource, this link relation MUST provide an URI that can be used
4847 with the HTTP PUT method to modify the atom:entry for the CMIS resource
 - 4848 o Service: getObject (GET), updateProperties (PUT)
 - 4849 o Media Type: application/atom+xml;type=entry
 - 4850 o Resources: CMIS Documents, CMIS Folders, CMIS Relationships and CMIS Policies
- 4851 • alternate
 - 4852 o This is used to express Renditions on a CMIS resource. See section 3.2.7 Renditions.
 - 4853 o Service: getContentStream for specified rendition
 - 4854 o Resources: CMIS Document, CMIS Folder and CMIS Policies
- 4855 • first
 - 4856 o This is used for Paging. Please see the AtomPub specification.
 - 4857 o Media Type: application/atom+xml;type=feed
 - 4858 o Resources: All Feeds
- 4859 • previous
 - 4860 o This is used for Paging. Please see the AtomPub specification.
 - 4861 o Media Type: application/atom+xml;type=feed
 - 4862 o Resources: All Feeds
- 4863 • next
 - 4864 o This is used for Paging. Please see the AtomPub specification.
 - 4865 o Media Type: application/atom+xml;type=feed
 - 4866 o Resources: All Feeds
- 4867 • last
 - 4868 o This is used for Paging. Please see the AtomPub specification.
 - 4869 o Media Type: application/atom+xml;type=feed

- 4870 ○ Resources: All Feeds
- 4871
- 4872
- 4873 Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on
4874 these link relations.
- 4875 **3.4.3.2 Hierarchy Navigation Internet Draft Link Relations**
- 4876 Please refer to the Internet Draft here: <http://www.ietf.org/internet-drafts/draft-divilly-atom-hierarchy-03.txt>.
- 4877
- 4878
- 4879 CMIS leverages the following link relations from the Internet Draft:
- 4880 • up
- 4881 ○ Service: getFolderParent, getObjectParents, getTypeDefinition, getObject
- 4882 ○ Media Type: application/atom+xml;type=feed, application/atom+xml;type=entry
- 4883 ○ Resources: CMIS Document, CMIS Folder, CMIS Type Definitions, CMIS Folder
4884 Children, CMIS Folder Descendants, CMIS FolderTree, CMIS Type Children, CMIS Type
4885 Descendants
- 4886 ▪ This link relation is not included on CMIS Base Type Definitions or the CMIS
4887 Root Folder
- 4888 • down
- 4889 ○ Service: getChildren, getDescendants, getTypeChildren, getTypeDescendants
- 4890 ○ Media Type:
- 4891 ▪ For children: application/atom+xml;type=feed
- 4892 ▪ For descendants: application/cmistree+xml
- 4893 • The descendants feed resource when retrieved from the CMIS repository
4894 will use the Atom Feed Media Type (application/atom+xml;type=feed)
- 4895 ○ Resources: CMIS Folder, Type
- 4896 **3.4.3.3 Versioning Internet Draft Link Relations**
- 4897 Please refer to the Internet Draft here: <http://www.ietf.org/internet-drafts/draft-brown-versioning-link-relationships-01.txt>.
- 4898
- 4899
- 4900 CMIS leverages the following link relations from the Internet Draft:
- 4901 • version-history
- 4902 ○ Service: getAllVersions
- 4903 ○ Media Type: application/atom+xml;type=feed
- 4904 ○ Resources: CMIS Document
- 4905 • current-version
- 4906 ○ Service: getObjectForLatestVersion
- 4907 ○ Media Type: application/atom+xml;type=entry
- 4908 ○ Resources: CMIS Document
- 4909 • working-copy
- 4910 ○ Service: getObject for private-working-copy specified by
4911 cmis:versionSeriesCheckedOutId property
- 4912 ○ Media Type: application/atom+xml;type=entry

- 4913 ○ Resources: CMIS Document
- 4914 **3.4.3.4 CMIS Specific Link Relations**
- 4915 CMIS defines the following link relations:
- 4916 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>
- 4917 ▪ This link relation MUST point to a resource containing a CMIS AllowableActions document for the CMIS resource containing this link relation.
- 4918 ▪ Service: getAllowableActions
- 4919 ▪ Media Type: application/cmisallowableactions+xml
- 4920 ▪ Resources: CMIS Documents, CMIS Folders, CMIS Policies, and CMIS Relationships
- 4921 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>
- 4922 ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS relationship resources for the CMIS resource containing this link relation.
- 4923 ▪ Service: getObjectRelationships
- 4924 ▪ Media Type: application/atom+xml;type=feed
- 4925 ▪ Resources: CMIS Documents, CMIS Folders, and CMIS Policies
- 4926 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/source>
- 4927 ▪ When used on a CMIS Relationship resource, this link relation MUST point to an atom entry document for the CMIS Resource specified by the cmis:sourcelId property on the relationship.
- 4928 ▪ Source Link on Relationship
- 4929 ▪ Media Type: application/atom+xml;type=entry
- 4930 ▪ Resources: CMIS Relationships
- 4931 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/target>
- 4932 ▪ When used on a CMIS Relationship resource, this link relation MUST point to an atom entry document for the CMIS Resource specified by the cmis:targetId property on the relationship.
- 4933 ▪ Target Link on Relationship
- 4934 ▪ Media Type: application/atom+xml;type=entry
- 4935 ▪ Resources: CMIS Relationships
- 4936 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/policies>
- 4937 ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS Policy resources for the CMIS resource containing this link relation.
- 4938 ▪ Service: getAppliedPolicies
- 4939 ▪ Media Type: application/atom+xml;type=feed
- 4940 ▪ Resources: CMIS Documents and CMIS Folders
- 4941 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/acl>
- 4942 ▪ This link relation MUST point to a resource containing a CMIS ACL document for the CMIS resource containing this link relation.
- 4943 ▪ Service: getACL
- 4944 ▪ Media Type: application/cmisacl+xml
- 4945 ▪ Resources: CMIS Documents, CMIS Folders, CMIS Relationships, and CMIS Policies that are securable
- 4946 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/changes>

- 4957 ▪ This link relation MUST point to an Atom Feed containing the set of changes
 4958 ▪ Service: getContentChanges
 4959 ▪ Media Type: application/atom+xml;type=feed
 4960 ▪ Resources: AtomPub Workspace Element in Service Document
- 4961 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>
 4962 ▪ Used in AtomPub Service Document to identify the folder tree for a specified
 4963 folder
 4964 ▪ Service: getFolderTree
 4965 ▪ Media Type: application/atom+xml;type=feed
 4966 ▪ Resources: CMIS Folder, also used in AtomPub Service Document for root folder
- 4967 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>
 4968 ▪ Used in AtomPub Service Document to identify the base types descendants
 4969 ▪ Service: getTypeDescendants
 4970 ▪ Media Type: application/atom+xml;type=feed
 4971 ▪ Resources: AtomPub Workspace Element in Service Document
- 4972 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/rootdescendants>
 4973 ▪ Used in AtomPub Service Document to identify the root folder descendants
 4974 ▪ Service: getDescendants for root folder
 4975 ▪ Media Type: application/atom+xml;type=feed
 4976 ▪ Resources: AtomPub Workspace Element in Service Document
- 4977

4978 **3.5 Atom Resources**

4979 For all Atom Resources used in this specification, the following MUST be followed:

4980 **3.5.1 Feeds**

- 4981 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis objects:
- 4982 • atom:updated SHOULD be the latest time the folder or its contents was updated. If unknown by
 4983 the underlying repository, it MUST be the current time.
 - 4984 • atom:author/atom:name MUST be the CMIS property cmis:createdBy
 - 4985 • atom:title MUST be the CMIS property cmis:name
 - 4986 • The atom:link with relation self MUST be generated to return the URI of the feed. If paging or any
 4987 other mechanism is used to filter, sort, or change the representation of the feed, the URI MUST
 4988 point back a resource with the same representation.
 - 4989 • A feed SHOULD contain the element app:collection, describing the appropriate media types
 4990 supported for creation of new entries in the feed
 - 4991 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's
 4992 specification and be a valid URI.
 - 4993 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
 4994 contained in the feed, then a link with the relation next MUST be included in the feed.
- 4995
- 4996 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis types:
- 4997 • atom:updated SHOULD be the latest time type definition was updated. If unknown by the
 4998 underlying repository, it MUST be the current time.

- 4999 • atom:author/atom:name is repository specific
5000 • atom:title MUST be the displayName attribute of the CMIS Type Definition.
5001 • The atom:link with relation self MUST be generated to return the URI of the feed
5002 • atom:id SHOULD be derived from the id attribute of the CMIS Type Definition. This id MUST be
5003 compliant with atom's specification and be a valid URI.
5004 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
5005 contained in the feed, then a link with the relation next MUST be included in the feed.

5006

5007 If on the root type, all fields are repository specific.

5008

5009 Ordering of entries in a feed is repository-specific if orderBy argument is not specified. If orderBy
5010 argument is specified, the order of the entries in the feed SHOULD conform to the ordering specified by
5011 the orderBy argument.

5012

5013 Note: Please see feedvalidator.org to validate Atom compliance.

5014 **3.5.2 Entries**

5015 At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry
5016 document and conform to the guidelines below for a cmis object:

- 5017 • atom:title MUST be the cmis:name property
5018 • app:edited MUST be cmis:lastModifiedDate
5019 • atom:updated MUST be cmis:lastModifiedDate
5020 • atom:published MUST be cmis:createdDate
5021 • atom:author/atom:name MUST be cmis:createdBy
5022 • All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are
5023 duplicated in an atom element
5024 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's
5025 specification and be a valid URI.
5026 • The repository SHOULD populate the atom:summary tag with text that best represents a
5027 summary of the object. For example, an HTML table containing the properties and their values or
5028 the description of the document if available.

5029

5030 For Documents that support Content Streams:

5031 The repository SHOULD use the atom:content/src attribute to point to the content stream.
5032 The client SHOULD use cmisra:content if the content is not well-formed or would have
5033 trouble fitting inside an atom:content element. The repository MUST use the
5034 cmisra:content element if provided by the client over the atom:content element.

5035

5036 Other Objects (Folders, Relationships, and other Document Types that do not support Content
5037 Streams, etc.):

5038 The repository MUST comply with the atom specification and have an atom:content
5039 element. This is repository specific. Any value in the content field MUST be ignored if the
5040 atom entry represents a non-document object by the CMIS repository when the atom
5041 entry is POST'ed to a collection or sent to the repository via a PUT.

5042

5043 When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding
5044 writable CMIS property. For example, atom:title will overwrite cmis:name.

5045

5046 At any point where an Atom document of CMIS Type is sent or returned, it must be a valid Atom Entry
5047 document and conform to the guidelines below for a cmis type definition:

- 5048 • atom:title MUST be the cmis:displayName
- 5049 • The repository SHOULD populate the atom:summary tag with text that best represents a
5050 summary of the object. For example, the type description if available.
- 5051 • The repository MUST comply with the atom specification and have an atom:content element. This
5052 is repository specific. Any value in the content field MUST be ignored if the atom entry represents
5053 a non-document object by the CMIS repository when the atom entry is POST'ed to a collection or
5054 sent to the repository via a PUT.

5055

5056

5057 Any atom element that is not specified is repository-specific.

5058 **3.5.2.1 Hierarchical Atom Entries**

5059 The repository SHOULD NOT provide any links to hierarchical objects if those capabilities are not
5060 supported with the exception of getTypeDescendants which is required

5061

5062 For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate
5063 a cmisra:children element in the atom:entry with the enclosing feed of its direct children. This pattern
5064 continues until the depth is satisfied.

5065

5066 The cmisra:children element that MUST be included in an atom entry:

5067 `<xs:element name="children" type="atom:feedType" />`

5068

5069 If an entry does not contain cmisra:children element, then the entry MAY have children even though it is
5070 not represented in the atom entry.

5071

5072 For Example, here is a minimal Atom Entry with CMIS Children Extension Element:

```
5073 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5074 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/" 
5075   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/" 
5076   xmlns:atom="http://www.w3.org/2005/Atom"
5077   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5078     open.org/ns/cmis/restatom/200908/">
5079   <atom:author>
5080     <atom:name>Al Brown</atom:name>
5081   </atom:author>
5082   <atom:content src="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-
5083     4233-85d3-39cad56fa091"/>
5084   <atom:id>urn:uuid:78a66015-ffd3-4233-85d3-39cad56fa091</atom:id>
5085   <atom:title type="text">CMIS Example Folder as Customer type</atom:title>
5086   <atom:updated>2009-10-19T10:09:58.906-07:00</atom:updated>
5087   <cmisra:object>
5088     <cmis:properties>
5089       <cmis:propertyId localName="rep-cmis:objectId"
5090         propertyDefinitionId="cmis:objectId">
5091         <cmis:value>78a66015-ffd3-4233-85d3-39cad56fa091</cmis:value>
5092       </cmis:propertyId>
5093     </cmis:properties>
```

```

5094     </cmisra:object>
5095     <cmisra:pathSegment>customer</cmisra:pathSegment>
5096     <cmisra:children>
5097         <atom:title type="text">CMIS Example Folder as Customer
5098 type</atom:title>
5099     <atom:author>
5100         <atom:name>Al Brown</atom:name>
5101         <atom:uri>http://www.ibm.com/</atom:uri>
5102         <atom:email>albertcbrown@us.ibm.com</atom:email>
5103     </atom:author>
5104     <atom:updated>2009-10-19T10:09:58.906-07:00</atom:updated>
5105     <atom:id>urn:uuid:d05alc90-bb02-42e2-baba-f51366e701c1</atom:id>
5106     <atom:link type="application/atom+xml;type=feed" rel="self"
5107 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5108 39cad56fa091/3"/>
5109     <atom:link type="application/atomsvc+xml" rel="service"
5110 href="http://cmisexample.oasis-open.org/rep1/service"/>
5111     <atom:link type="application/atom+xml;type=entry" rel="via"
5112 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5113 39cad56fa091"/>
5114     <atom:link type="application/atom+xml;type=feed"
5115 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
5116 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5117 39cad56fa091/foldertree"/>
5118     <atom:link type="application/atom+xml;type=feed" rel="down"
5119 href="http://cmisexample.oasis-open.org/rep1/78a66015-ffd3-4233-85d3-
5120 39cad56fa091/children"/>
5121     <atom:link type="application/atom+xml;type=entry" rel="up"
5122 href="http://cmisexample.oasis-open.org/rep1/c84ae842-15a2-4571-93fe-
5123 65cca95faa24"/>
5124     <atom:entry>
5125         <atom:author>
5126             <atom:name>Al Brown</atom:name>
5127         </atom:author>
5128         <atom:content src="http://cmisexample.oasis-
5129 open.org/rep1/78a66015-ffd3-4233-85d3-39cad56fa091"/>
5130         <atom:id>urn:uuid:78a66015-ffd3-4233-85d3-39cad56fa091</atom:id>
5131         <atom:title type="text">CMIS Example Child of Folder</atom:title>
5132         <atom:updated>2009-10-19T10:09:58.906-07:00</atom:updated>
5133         <cmisra:object>
5134             <cmis:properties>
5135                 <cmis:propertyId localName="rep-cmis:objectId"
5136 propertyDefinitionId="cmis:objectId">
5137                     <cmis:value>78a66015-ffd3-4233-85d3-
5138 39cad56fa091</cmis:value>
5139                     </cmis:propertyId>
5140                 </cmis:properties>
5141             </cmisra:object>
5142             <cmisra:pathSegment>document</cmisra:pathSegment>
5143         </atom:entry>
5144     </cmisra:children>
5145 </atom:entry>

```

5147 Please also see the example documents included with the schema.

5148 3.6 AtomPub Service Document (Repository)

5149 The AtomPub Service Document contains the set of repositories that are available. Each repository is
5150 mapped to a app:workspace element in the AtomPub Service document.

5151

5152 CMIS Services exposed:

5153 GET: getRepositories, getRepositoryInfo

5154
5155 Media Type: application/atomsvc+xml
5156
5157 How the client will get the initial AtomPub (APP) service document or the URI for the service document is
5158 repository specific. Examples are via URI, or loading the service document from disk.
5159
5160 The service document will be available from Atom Entry and Atom Feed documents via a link relationship,
5161 service. That AtomPub service document MUST contain only one workspace element which MUST be
5162 the workspace representing the repository containing the Atom Entry or Atom Feed document.
5163
5164 A workspace element for a CMIS repository MUST have a collection element for each of following
5165 collections: Each collection MUST also contain a cmisra:collectionType attribute with the given value:
5166

- Root Folder Collection: Root folder of the Repository
 - ‘root’ for the children collection of the root folder
 - cmisra:collectiontype='root'
- Types Collection: Collection containing all the types in the repository
 - ‘types’ for the children collection
 - cmisra:collectiontype='types'

5167
5168
5169
5170
5171
5172
5173 The workspace element SHOULD contain these collections if the repository supports this functionality:
5174

- CheckedOut collection: collection containing all checked out documents user can see
 - ‘checkedout’
 - cmisra:collectiontype='checkedout'
- Query collection: Collection for posting queries to be executed
 - ‘query’
 - cmisra:collectiontype='query'
- Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
 - ‘unfiled’
 - cmisra:collectiontype='unfiled'

5175
5176
5177
5178
5179
5180
5181
5182
5183
5184 The repository MUST include the URI templates in the workspace elements.
5185
5186 The workspace element MUST also contain the following link element with the relation:
5187

- <http://docs.oasis-open.org/ns/cmis/link/200908/typesdescendants>: This link relation points to the
5188 types descendants for the base types in the repository.

5189
5190 The workspace element MUST contain the following link relations for those services which are supported
5191 by the repository:
5192

- <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: This link relation points to the folder
5193 tree of the root folder. See Folder Tree resource for more information.
- <http://docs.oasis-open.org/ns/cmis/link/200908/rootdescendants>: This link relation points to the
5195 descendants feed for the root folder.
- <http://docs.oasis-open.org/ns/cmis/link/200908/changes>: This link relation points to the changes
5197 feed for the repository.

5198

5199 The workspace element may include app:collection element for the collections that represent folders in
5200 the repository. However, an alternative approach, especially for a repository with many folders, is to not
5201 enumerate those collections here, but include the app:collection element per RFC5023 in the Atom Feed
5202 document.

5203 **3.6.1 URI Templates**

5204

5205 CMIS defines the following URI Templates:

- 5206
 - objectbyid
 - objectbypath
 - query
 - typebyid

5210

5211 Repositories MUST provide the following URI Templates:

- 5212
 - objectbyid
 - objectbypath
 - typebyid

5215

5216 Repositories MUST provide the URI Template query if the repository supports query.

5217

5218 Repositories MAY extend that set of templates. Those URI Template Types will be repository specific.
5219 Repositories MAY have more than one entry per URI Template type if the entries have different media
5220 types.

5221

5222 URI Templates are simple replacement of the template parameter with the specified value. If a client
5223 does not want to specify a value for some of these variables, then the client MUST substitute an empty
5224 string for the variable.

5225

5226 For example, if the URI template that supports the variable {id} is

5227 `http://example.org/repl/getbyid/{id}`

5228

5229 If the client wants to find the entry for an object with an id of 'obj_1' then the URI would be:

5230 `http://example.org/repl/getbyid/obj_1`

5231

5232 Arguments that are substituted for URI template parameters MUST be percent escaped according to
5233 RFC3986. Please see that RFC for more information.

5234

5235 All variables MUST be in the template.

5236

5237 Structure of URI Template:

```
5238 <xs:complexType name="cmisUriTemplateType">
5239   <xs:sequence>
5240     <xs:element name="template" type="xs:string" />
5241     <xs:element name="type" type="xs:string" />
5242     <xs:element name="mediatype" type="xs:string" />
5243     <xs:any processContents="lax" namespace="##other"
5244       minOccurs="0" />
```

```
5245                               maxOccurs="unbounded" />
5246                         </xs:sequence>
5247                   </xs:complexType>
```

5248

5249

5250 Example of URI Template element in an AtomPub Workspace Element:

```
5251 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5252   <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5253     open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5254     open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5255     xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5256     open.org/ns/cmis/restatom/200908/">
5257     <cmisra:template>http://cmisexample.oasis-
5258     open.org/rep1/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
5259     ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
5260     ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
5261     <cmisra:type>objectbyid</cmisra:type>
5262     <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5263   </cmisra:uritemplate>
```

5264

5265 Please also see the example documents included with the schema.

5266

5267 3.6.1.1 Object By Id

5268 This URI template provides a method for creating an URI that directly accesses an atom entry
5269 representing documents, folders, policies or relationship objects. See section 3.8 for more information.

5270

5271 Type: objectbyid

5272 Media Type: application/atom+xml;type=entry

5273

5274 Service: getObjectById

5275

5276 Variables that are supported by the template:

- 5277 • {id}: Id of object
- 5278 • {filter}: Property Filter
- 5279 • {includeAllowableActions}
 - 5280 ○ Valid values: true, false
- 5281 • {includePolicyIds}: Include Policy Ids:
 - 5282 ○ Valid values: true, false
- 5283 • {includeRelationships}: Include relationships
 - 5284 ○ Valid values: See enumIncludeRelationships
- 5285 • {includeACL}: Include ACLs
 - 5286 ○ Valid values: true, false
- 5287 • {renditionFilter}
 - 5288 ○ Valid values: Please see renditionFilter in CMIS Domain Model

5289

5290 Example:

```
5291   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

5292 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5293 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5294 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5295 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5296 open.org/ns/cmis/restatom/200908/">
5297   <cmisra:template>http://cmisexample.oasis-
5298 open.org/repl/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
5299 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
5300 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
5301   <cmisra:type>objectbyid</cmisra:type>
5302   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5303 </cmisra:uritemplate>
5304

```

5305
5306 Please also see the example documents included with the schema.

5307 **3.6.1.2 Object By Path**

5308 This URI template provides a method for creating an URI that directly accesses an atom entry
5309 representing documents, folders or policy objects. See section 3.8 for more information.

5310
5311 Type: objectbypath

5312 Media Type: application/atom+xml;type=entry

5313
5314 Service: getObjectByPath

5315
5316 Variables that are supported by the template:

- 5317 • {path}: Path of Object
- 5318 • {filter}: Property Filter
- 5319 • {includeAllowableActions}: Boolean for include Allowable Actions
 - 5320 ○ Valid values: true, false
- 5321 • {includePolicyIds}: Include Policy Ids:
 - 5322 ○ Valid values: true, false
- 5323 • {includeRelationships}: Include relationships
 - 5324 ○ Valid values: See enumIncludeRelationships
- 5325 • {includeACL}: Include ACLs
 - 5326 ○ Valid values: true, false
- 5327 • {renditionFilter}
 - 5328 ○ Valid values: Please see renditionFilter in CMIS Domain Model

5329
5330
5331 Example:

```

5332 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5333 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5334 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5335 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5336 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5337 open.org/ns/cmis/restatom/200908/">
```

```
5338      <cmisra:template>http://cmisexample.oasis-
5339      open.org/rep1/objectbypath?p={path}&filter={filter}&includeAllowableAc-
5340      tions={includeAllowableActions}&includePolicyIds={includePolicyIds}&in-
5341      cludeRelationships={includeRelationships}&includeACL={includeACL}</cmisra:
5342      template>
5343          <cmisra:type>objectbypath</cmisra:type>
5344          <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5345      </cmisra:uritemplate>
5346
```

5347

5348 Please also see the example documents included with the schema.

5349

5350 **3.6.1.3 Query**

5351 Type: query

5352 Media Type: application/atom+xml;type=feed

5353

5354 Service: query

5355

5356 Variables that are supported by the template:

- {q}: CMIS Query Statement
- {searchAllVersions}: Boolean, true if to search all versions
- {maxItems}: Integer, Max items to return
- {skipCount}: Integer, Items to skip
- {includeAllowableActions}: Boolean
- {includeRelationships}: Boolean

5363

5364

5365 Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<cmisra:uritemplate xmlns:cmis="http://docs.oasis-
open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
open.org/ns/cmis/restatom/200908/">
    <cmisra:template>http://cmisexample.oasis-
open.org/rep1/query?q={q}&searchAllVersions={searchAllVersions}&maxIte-
ms={maxItems}&skipCount={skipCount}&includeAllowableActions={includeAl-
lowableActions}&includeRelationships={includeRelationships}</cmisra:tem-
plate>
        <cmisra:type>query</cmisra:type>
        <cmisra:mediatype>application/atom+xml;type=feed</cmisra:mediatype>
    </cmisra:uritemplate>
```

5381

5382 Please also see the example documents included with the schema.

5383

5384 **3.6.1.4 Type By Id**

5385 Type: typebyid

5386 Media Type: application/atom+xml;type=entry
5387
5388 Service: getTypeDefinition
5389
5390 Variables that are supported by the template:
5391 • {id}: CMIS Type Id
5392
5393
5394 Example:
5395 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5396 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5397 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5398 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5399 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5400 open.org/ns/cmis/restatom/200908/">
5401 <cmisra:template>http://cmisexample.oasis-
5402 open.org/repl?type?id={id}</cmisra:template>
5403 <cmisra:type>query</cmisra:type>
5404 <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5405 </cmisra:uritemplate>

5406
5407 Please also see the example documents included with the schema.
5408
5409

5410 3.6.2 HTTP Methods

5411 3.6.2.1 GET

5412 This retrieves the AtomPub Service document for a specified repository. This exposes the capabilities
5413 defined in getRepositories and getRepositoryInfo in the Domain Model.

5414
5415 The optional argument MAY be specified:

- 5416 • repositoryId:
- 5417 ○ This query parameter allows a client to specify a different repository than the one that is
5418 referenced by the URI.
 - 5419 ○ If specified, the repository MUST return the AtomPub services document for the specified
5420 repository if that repository exists.
 - 5421 ○ If not specified, the repository MUST return the service document for the repository that is
5422 referenced by URI.

5424 3.7 Service Collections

5425 These are the collections that are included on an AtomPub Service document in the workspace element.

5426 3.7.1 Root Folder Collection

5427 This is a collection described in the service document. Please see [Folder Children](#).

5428 **3.7.2 Query Collection**

5429 This is a collection for processing queries. If the implementation supports GET on this collection, then the
5430 implementation SHOULD at least return a feed consisting of zero or more atom entries. These atom
5431 entries should represent persisted objects related to query such as persisted queries, long running
5432 queries or search templates.

5433

5434 CMIS Services exposed via HTTP verbs:

5435 POST: Query

5436

5437 Media Type: application/atom+xml;type=feed

5438 Accept:

- 5439 • MUST support CMIS Query document,
- 5440 • MAY support other media type

5441

5442 Link Relations on resulting feed from Query Collection:

- 5443 • service: Points to service document containing the CMIS repository. The service document
5444 MUST contain only one workspace element.
 - 5445 o Media Type: application/atomsrv+xml
- 5446 • paging link relations as appropriate: first, next, previous, last

5447

5448 The following CMIS Atom extension element MAY be included inside the atom feed:

- 5449 • cmisra:numItems

5450

5451 The following CMIS Atom extension element MUST be included inside the atom entries:

- 5452 • cmisra:object inside atom:entry

5453

5454 **3.7.2.1 POST**

5455 This collection MUST accept CMIS Query documents (application/cmisquery+xml).

5456

5457 Upon submission (creation) of a query document, a response must be returned with a Location header
5458 representing the feed for that query. If the query cannot be performed and an atom feed returned, the
5459 repository MUST return the appropriate HTTP status code. In addition, the server SHOULD return the
5460 feed directly. If the server does so, the server should also return the Content-Location header.

5461

5462 The feed returned MUST contain a set of atom entries representing the result set from the query.

5463

5464 The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The
5465 atom entries MUST contain the CMIS extension element (cmis:object) containing the properties specified
5466 by the query in the select clause of the query statement.

5467

5468 If all the selected properties can be mapped to the same type reference, then the repository MAY include
5469 additional information in the atom entry.

5470

5471 Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

5472
5473 Status Codes:
5474 • 201 Success
5475
5476 Headers returned:
5477 • Location Header
5478 • Content-Location Header
5479
5480 Link Relations on resulting feed from POST to Query Collection:
5481 • service: Points to service document containing the CMIS repository. The service document
5482 MUST contain only one workspace element.
5483 ◦ Media Type: application/atomsrv+xml
5484 • paging link relations as appropriate: first, next, previous, last
5485
5486 Example client request:
5487 POST /Query HTTP/1.1
5488 Host: example.org
5489 Content-Length: 756
5490 Content-Type: application/cmiskey+xml
5491
5492
5493 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5494 <cmis:query xmlns:app="http://www.w3.org/2007/app"
5495 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5496 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5497 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5498 open.org/ns/cmis/restatom/200908/">
5499 <cmis:statement>SELECT cmis:objectId FROM cmis:document</cmis:statement>
5500 <cmis:searchAllVersions>true</cmis:searchAllVersions>
5501 <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
5502 <cmis:includeRelationships>none</cmis:includeRelationships>
5503 <cmis:renditionFilter>*</cmis:renditionFilter>
5504 <cmis:maxItems>50</cmis:maxItems>
5505 <cmis:skipCount>0</cmis:skipCount>
5506 </cmis:query>

5507
5508
5509
5510 Example server response:
5511 HTTP/1.1 201 Created
5512 Date: Mon, 19 Oct 2009 10:10:01 -0700
5513 Content-Length: 1830
5514 Content-Type: application/atom+xml;type=feed
5515 Content-Location: http://cmisexample.oasis-open.org/repl/queryresult/6e23b7c6-
5516 e377-4a6b-b21c-7450f28b4b5d
5517 Location: http://cmisexample.oasis-open.org/repl/queryresult/6e23b7c6-e377-
5518 4a6b-b21c-7450f28b4b5d
5519
5520
5521 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5522 <atom:feed xmlns:app="http://www.w3.org/2007/app"
5523 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5524 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5525 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5526 open.org/ns/cmis/restatom/200908/">
5527 <atom:title type="text">CMIS Query Result for SELECT cmis:objectId FROM
5528 cmis:document</atom:title>
5529 <atom:author>

```

5530      <atom:name>Al Brown</atom:name>
5531      <atom:uri>http://www.ibm.com</atom:uri>
5532      <atom:email>albertcbrown@us.ibm.com</atom:email>
5533    </atom:author>
5534    <atom:updated>2009-10-19T10:10:01.078-07:00</atom:updated>
5535    <atom:id>urn:uuid:7fd19974-9597-4eb9-88ca-3a2297cd893c</atom:id>
5536    <atom:link type="application/atom+xml;type=feed" rel="self"
5537 href="http://cmisexample.oasis-open.org/repl/c7f915fb-c222-45c9-8fa2-
5538 5062102c62b9/3"/>
5539    <atom:link type="application/atomsvc+xml" rel="service"
5540 href="http://cmisexample.oasis-open.org/repl//service"/>
5541    <atom:entry>
5542      <atom:author>
5543        <atom:name>Al Brown</atom:name>
5544      </atom:author>
5545      <atom:content src="http://cmisexample.oasis-open.org/repl/b4519062-
5546 6b13-4160-8abe-1c43bfbfe32e"/>
5547      <atom:id>urn:uuid:b4519062-6b13-4160-8abe-1c43bfbfe32e</atom:id>
5548      <atom:title type="text">Resulting Document</atom:title>
5549      <atom:updated>2009-10-19T10:10:01.078-07:00</atom:updated>
5550      <cmisra:object>
5551        <cmis:properties>
5552          <cmis:propertyId queryName="cmis:objectId" localName="rep-
5553 cmis:objectId" propertyDefinitionId="cmis:objectId">
5554            <cmis:value>b4519062-6b13-4160-8abe-
5555 1c43bfbfe32e</cmis:value>
5556          </cmis:propertyId>
5557        </cmis:properties>
5558      </cmisra:object>
5559    </atom:entry>
5560  </atom:feed>
5561
```

5562

5563 Please also see the example documents included with the schema.

5564

5565 3.7.3 Checked Out Collection

5566 This is a collection described in the service document that contains all the checkedout documents

5567 CMIS Services:

5568 GET: getCheckedOutDocs

5569 POST: checkOut

5570 Media Type: application/atom+xml;type=feed

5571 Accept:

- 5572 • MUST support Atom Entry Documents with CMIS extensions

5573 ○ application/atom+xml;type=entry or

5574 ○ application/cmisatom+xml

- 5575 • MAY support other media type

5576

5577 Link Relations:

- 5578 • service: Points to service document containing the CMIS repository. The service document
- 5579 MUST contain only one workspace element.

5580 ○ Media Type: application/atomsvc+xml

- 5581 • paging link relations as appropriate: first, next, previous, last

5582
5583 The following CMIS Atom extension element MAY be included inside the atom feed:
5584 • cmisra:numItems
5585
5586 The following CMIS Atom extension element MUST be included inside the atom entries:
5587 • cmisra:object inside atom:entry
5588

5589 **3.7.3.1 GET**

5590 The following arguments may be supplied. Please see the domain model for more information:
5591 • filter
5592 • folderId
5593 • maxItems
5594 • skipCount
5595 • includeAllowableActions
5596 • includeRelationships

5597 **3.7.3.2 POST**

5598 When an atom entry is POST'ed to this collection, the atom entry will be checked out. A Content-
5599 Location header MUST be returned containing the location of the private working copy.
5600

5601 Example client request:

```
5602 POST /CheckedOut HTTP/1.1
5603 Host: example.org
5604 Content-Length: 1044
5605 Content-Type: application/atom+xml;type=entry
5606
5607
5608 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5609 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5610 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5611 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5612 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5613 open.org/ns/cmis/restatom/200908/">
5614     <atom:author>
5615         <atom:name>Al Brown</atom:name>
5616     </atom:author>
5617     <atom:content src="http://cmisexample.oasis-open.org/rep1/46559af4-db97-
5618 471d-b229-d9b27322bf43"/>
5619     <atom:id>urn:uuid:46559af4-db97-471d-b229-d9b27322bf43</atom:id>
5620     <atom:title type="text">CMIS Example Document to checkout</atom:title>
5621     <atom:updated>2009-10-19T10:10:01.031-07:00</atom:updated>
5622     <cmisra:object>
5623         <cmis:properties>
5624             <cmis:propertyId localName="rep-cmis:objectId"
5625             propertyDefinitionId="cmis:objectId">
5626                 <cmis:value>46559af4-db97-471d-b229-d9b27322bf43</cmis:value>
5627             </cmis:propertyId>
5628         </cmis:properties>
5629     </cmisra:object>
5630 </atom:entry>
```

5631
5632

5633 Example server response:

```
5634 HTTP/1.1 201 Created
5635 Date: Mon, 19 Oct 2009 10:10:01 -0700
5636 Content-Length: 7846
5637 Content-Type: application/atom+xml;type=entry
5638 Content-Location: http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-
5639 bf6b-e9e30341581f
5640 Location: http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
5641 e9e30341581f
5642
5643
5644 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5645 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5646 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5647 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5648 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5649 open.org/ns/cmis/restatom/200908"/>
5650     <atom:author>
5651         <atom:name>Al Brown</atom:name>
5652         <atom:uri>http://www.ibm.com/</atom:uri>
5653         <atom:email>albertcbrown@us.ibm.com</atom:email>
5654     </atom:author>
5655     <atom:content src="http://cmisexample.oasis-open.org/repl/64f55634-f2de-
5656 443c-bf6b-e9e30341581f"/>
5657     <atom:id>urn:uuid:64f55634-f2de-443c-bf6b-e9e30341581f</atom:id>
5658     <atom:title type="text">CMIS Example Child of Folder</atom:title>
5659     <atom:updated>2009-10-19T10:10:01.046-07:00</atom:updated>
5660     <atom:link rel="self" href="http://cmisexample.oasis-
5661 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f"/>
5662     <atom:link rel="edit" href="http://cmisexample.oasis-
5663 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f"/>
5664     <atom:link type="application/cmis+xml;type=allowableActions"
5665 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
5666 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
5667 e9e30341581f/allowableactions"/>
5668     <atom:link type="application/atom+xml;type=entry" rel="describedby"
5669 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
5670 e9e30341581f/type"/>
5671     <atom:link type="application/atomsvc+xml" rel="service"
5672 href="http://cmisexample.oasis-open.org/repl/service"/>
5673     <atom:published>2009-10-19T10:10:01.046-07:00</atom:published>
5674     <atom:summary type="html">HTML summary of Entry 64f55634-f2de-443c-bf6b-
5675 e9e30341581f</atom:summary>
5676     <atom:link rel="edit-media" href="http://cmisexample.oasis-
5677 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/edit-media"/>
5678     <atom:link rel="alternate" href="http://cmisexample.oasis-
5679 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/alternate"/>
5680     <atom:link type="application/atom+xml;type=feed" rel="up"
5681 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
5682 e9e30341581f/parents"/>
5683     <atom:link type="application/atom+xml;type=feed" rel="version-history"
5684 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
5685 e9e30341581f/allversions"/>
5686     <atom:link type="application/atom+xml;type=entry" rel="current-version"
5687 href="http://cmisexample.oasis-open.org/repl/64f55634-f2de-443c-bf6b-
5688 e9e30341581f/latest"/>
5689     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
5690 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
5691 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/relationships"/>
5692     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
5693 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
5694 open.org/repl/64f55634-f2de-443c-bf6b-e9e30341581f/policies"/>
```

```

5695      <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
5696      open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
5697      open.org/rep1/64f55634-f2de-443c-bf6b-e9e30341581f/acl"/>
5698      <atom:link type="application/atom+xml;type=feed" rel="working-copy"
5699      href="http://cmisexample.oasis-open.org/rep1/64f55634-f2de-443c-bf6b-
5700      e9e30341581f/pwc"/>
5701      <cmisra:object>
5702          <cmis:properties>
5703              <cmis:propertyId localName="rep-cmis:objectId"
5704              propertyDefinitionId="cmis:objectId">
5705                  <cmis:value>64f55634-f2de-443c-bf6b-e9e30341581f</cmis:value>
5706              </cmis:propertyId>
5707              <cmis:propertyId localName="rep-cmis:objectTypeId"
5708              propertyDefinitionId="cmis:objectTypeId">
5709                  <cmis:value>customer</cmis:value>
5710              </cmis:propertyId>
5711              <cmis:propertyString localName="rep-cmis:name"
5712              propertyDefinitionId="cmis:name">
5713                  <cmis:value>CMIS Example Child of Folder</cmis:value>
5714              </cmis:propertyString>
5715              <cmis:propertyDateTime localName="rep-cmis:creationDate"
5716              propertyDefinitionId="cmis:creationDate">
5717                  <cmis:value>2009-10-19T10:10:01.046-07:00</cmis:value>
5718              </cmis:propertyDateTime>
5719              <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
5720              propertyDefinitionId="cmis:lastModificationDate">
5721                  <cmis:value>2009-10-19T10:10:01.046-07:00</cmis:value>
5722              </cmis:propertyDateTime>
5723              <cmis:propertyId localName="rep-cmis:baseTypeId"
5724              propertyDefinitionId="cmis:baseTypeId">
5725                  <cmis:value>cmis:document</cmis:value>
5726              </cmis:propertyId>
5727              <cmis:propertyString localName="rep-cmis:lastModifiedBy"
5728              propertyDefinitionId="cmis:lastModifiedBy">
5729                  <cmis:value>Al Brown</cmis:value>
5730              </cmis:propertyString>
5731              <cmis:propertyString localName="rep-cmis:createdBy"
5732              propertyDefinitionId="cmis:createdBy">
5733                  <cmis:value>Al Brown</cmis:value>
5734              </cmis:propertyString>
5735              <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
5736              propertyDefinitionId="cmis:isLatestVersion">
5737                  <cmis:value>true</cmis:value>
5738              </cmis:propertyBoolean>
5739              <cmis:propertyBoolean localName="rep-
5740              cmis:isVersionSeriesCheckedOut"
5741              propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
5742                  <cmis:value>true</cmis:value>
5743              </cmis:propertyBoolean>
5744              <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
5745              propertyDefinitionId="cmis:isMajorVersion">
5746                  <cmis:value>false</cmis:value>
5747              </cmis:propertyBoolean>
5748              <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
5749              propertyDefinitionId="cmis:isLatestMajorVersion">
5750                  <cmis:value>false</cmis:value>
5751              </cmis:propertyBoolean>
5752              <cmis:propertyBoolean localName="rep-cmis:isImmutable"
5753              propertyDefinitionId="cmis:isImmutable">
5754                  <cmis:value>false</cmis:value>
5755              </cmis:propertyBoolean>
5756              <cmis:propertyString localName="rep-cmis:checkinComment"
5757              propertyDefinitionId="cmis:checkinComment">
5758                  <cmis:value>Checkin comment</cmis:value>

```

```

5759         </cmis:propertyString>
5760             <cmis:propertyString localName="rep-cmis:versionLabel">
5761                 <cmis:value>0.1</cmis:value>
5762             </cmis:propertyString>
5763             <cmis:propertyString localName="rep-cmis:contentStreamMimeType">
5764                 <cmis:value>text/plain</cmis:value>
5765             </cmis:propertyString>
5766             <cmis:propertyString localName="rep-cmis:contentStreamFileName">
5767                 <cmis:value>text.txt</cmis:value>
5768             </cmis:propertyString>
5769             <cmis:propertyInteger localName="rep-cmis:contentStreamLength">
5770                 <cmis:value>4234</cmis:value>
5771             </cmis:propertyInteger>
5772             <cmis:propertyString displayName="Keywords for Document">
5773                 localName="keywords" propertyDefinitionId="keywords">
5774                     <cmis:value>document</cmis:value>
5775                     <cmis:value>example</cmis:value>
5776                     <cmis:value>sample</cmis:value>
5777                     <cmis:value>cmis</cmis:value>
5778                 </cmis:propertyString>
5779                 <cmis:propertyId localName="rep-cmis:versionSeriesCheckedOutId">
5780                     <cmis:value>64f55634-f2de-443c-bf6b-e9e30341581f</cmis:value>
5781                 </cmis:propertyId>
5782                 <cmis:propertyString localName="rep-
5783 cmis:versionSeriesCheckedOutBy">
5784                     <cmis:value>Al Brown</cmis:value>
5785                 </cmis:propertyString>
5786             </cmis:properties>
5787         </cmisra:object>
5788     </atom:entry>
5789
5790
5791
5792
5793
5794
5795

```

5796
5797 Please also see the example documents included with the schema.
5798

5799 **3.7.4 Unfiled Collection**

5800 This is a collection described in the service document that contains all the unfiled documents in the
5801 repository. if un filing is supported by the repository

5802 CMIS Services:

5803 GET: getUnfiled
5804 POST: removeObjectFromFolder
5805 Media Type: application/atom+xml;type=feed
5806 Accept:

- MUST support Atom Entry Documents with CMIS extensions
 - application/atom+xml;type=entry or
 - application/cmisatom+xml
- MAY support other media type

5811
5812 Link Relations:

- 5813 • service: Points to service document containing the CMIS repository. The service document
5814 MUST contain only one workspace element.
5815 ◦ Media Type: application/atomsrv+xml
5816 • paging link relations as appropriate: first, next, previous, last

5817
5818 The following CMIS Atom extension element MAY be included inside the atom feed:

- 5819 • cmisra:numItems

5820
5821 The following CMIS Atom extension element MUST be included inside the atom entries:

- 5822 • cmisra:object inside atom:entry

5824 **3.7.4.1 GET**

5825 The following arguments may be supplied. Please see the domain model for more information:

- 5826 • filter
5827 • folderId
5828 • maxItems
5829 • skipCount
5830 • includeAllowableActions
5831 • includeRelationships

5832 **3.7.4.2 POST**

5833 This removes the object from all folders in the repository by default. If the optional argument removeFrom
5834 is specified, the object will only be removed from that folder only.

5835
5836 If the Atom Entry POST'ed, does not have the CMIS extensions with a valid cmis:objectId, the document
5837 does not exist, or the document is not in that folder, the appropriate HTTP status code MUST be returned.

5838
5839 This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 5840 • HTTP Success: 201
5841 • Location Header

5842
5843 The following arguments may be supplied. Please see the domain model for more information:

- 5844 • removeFrom: For repositories which support multi-filing, this parameter identifies which folder to
5845 remove this object from. If specified, it indicates the folder from which the object shall be moved.
5846 If not specified, the object will be removed from all folders.

5847
5848 Example client request:

```
5849    POST /Unfiled HTTP/1.1
5850    Host: example.org
5851    Content-Length: 1043
5852    Content-Type: application/atom+xml;type=entry
5853
5854
5855   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

5856 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5857   xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5858   open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5859   open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5860   open.org/ns/cmis/restatom/200908/">
5861   <atom:author>
5862     <atom:name>Al Brown</atom:name>
5863   </atom:author>
5864   <atom:content src="http://cmisexample.oasis-open.org/repl/1f8ceb22-cc15-
5865   4d75-9221-00588cd22bdc"/>
5866   <atom:id>urn:uuid:1f8ceb22-cc15-4d75-9221-00588cd22bdc</atom:id>
5867   <atom:title type="text">CMIS Example Document to unfiled</atom:title>
5868   <atom:updated>2009-10-19T10:10:01.078-07:00</atom:updated>
5869   <cmisra:object>
5870     <cmis:properties>
5871       <cmis:propertyId localName="rep-cmis:objectId"
5872       propertyDefinitionId="cmis:objectId">
5873         <cmis:value>1f8ceb22-cc15-4d75-9221-00588cd22bdc</cmis:value>
5874       </cmis:propertyId>
5875     </cmis:properties>
5876   </cmisra:object>
5877 </atom:entry>
5878
5879

```

5880 Example server response:

```

5881 HTTP/1.1 201 Created
5882 Date: Mon, 19 Oct 2009 10:10:01 -0700
5883 Content-Length: 7234
5884 Content-Type: application/atom+xml;type=entry
5885 Content-Location: http://cmisexample.oasis-open.org/repl/queryresult/e2d2c9b3-
5886 dab4-4c65-866a-d8006d1bb16f
5887 Location: http://cmisexample.oasis-open.org/repl/queryresult/e2d2c9b3-dab4-
5888 4c65-866a-d8006d1bb16f
5889
5890
5891 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5892 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5893   xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5894   open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5895   open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5896   open.org/ns/cmis/restatom/200908/">
5897   <atom:author>
5898     <atom:name>Al Brown</atom:name>
5899     <atom:uri>http://www.ibm.com/</atom:uri>
5900     <atom:email>albertcbrown@us.ibm.com</atom:email>
5901   </atom:author>
5902   <atom:content src="http://cmisexample.oasis-open.org/repl/e2d2c9b3-dab4-
5903   4c65-866a-d8006d1bb16f"/>
5904   <atom:id>urn:uuid:e2d2c9b3-dab4-4c65-866a-d8006d1bb16f</atom:id>
5905   <atom:title type="text">CMIS Example Document to unfiled</atom:title>
5906   <atom:updated>2009-10-19T10:10:01.093-07:00</atom:updated>
5907   <atom:link rel="self" href="http://cmisexample.oasis-
5908   open.org/repl/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f"/>
5909   <atom:link rel="edit" href="http://cmisexample.oasis-
5910   open.org/repl/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f"/>
5911   <atom:link type="application/cmis+xml;type=allowableActions"
5912   rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
5913   href="http://cmisexample.oasis-open.org/repl/e2d2c9b3-dab4-4c65-866a-
5914   d8006d1bb16f/allowableactions"/>
5915   <atom:link type="application/atom+xml;type=entry" rel="describedby"
5916   href="http://cmisexample.oasis-open.org/repl/e2d2c9b3-dab4-4c65-866a-
5917   d8006d1bb16f/type"/>

```

```

5918     <atom:link type="application/atomsvc+xml" rel="service"
5919     href="http://cmisexample.oasis-open.org/rep1/service"/>
5920     <atom:published>2009-10-19T10:10:01.093-07:00</atom:published>
5921     <atom:summary type="html">HTML summary of Entry e2d2c9b3-dab4-4c65-866a-
5922 d8006d1bb16f</atom:summary>
5923     <atom:link rel="edit-media" href="http://cmisexample.oasis-
5924 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/edit-media"/>
5925     <atom:link rel="alternate" href="http://cmisexample.oasis-
5926 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/alternate"/>
5927     <atom:link type="application/atom+xml;type=feed" rel="up"
5928 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
5929 d8006d1bb16f/parents"/>
5930     <atom:link type="application/atom+xml;type=feed" rel="version-history"
5931 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
5932 d8006d1bb16f/allversions"/>
5933     <atom:link type="application/atom+xml;type=entry" rel="current-version"
5934 href="http://cmisexample.oasis-open.org/rep1/e2d2c9b3-dab4-4c65-866a-
5935 d8006d1bb16f/latest"/>
5936     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
5937 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
5938 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/relationships"/>
5939     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
5940 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
5941 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/policies"/>
5942     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
5943 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
5944 open.org/rep1/e2d2c9b3-dab4-4c65-866a-d8006d1bb16f/acl"/>
5945     <cmisra:object>
5946         <cmis:properties>
5947             <cmis:propertyId localName="rep-cmis:objectId"
5948 propertyDefinitionId="cmis:objectId">
5949                 <cmis:value>e2d2c9b3-dab4-4c65-866a-d8006d1bb16f</cmis:value>
5950             </cmis:propertyId>
5951             <cmis:propertyId localName="rep-cmis:objectTypeId"
5952 propertyDefinitionId="cmis:objectTypeId">
5953                 <cmis:value>customer</cmis:value>
5954             </cmis:propertyId>
5955             <cmis:propertyString localName="rep-cmis:name"
5956 propertyDefinitionId="cmis:name">
5957                 <cmis:value>CMIS Example Document to unfiled</cmis:value>
5958             </cmis:propertyString>
5959             <cmis:propertyDateTime localName="rep-cmis:creationDate"
5960 propertyDefinitionId="cmis:creationDate">
5961                 <cmis:value>2009-10-19T10:10:01.093-07:00</cmis:value>
5962             </cmis:propertyDateTime>
5963             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
5964 propertyDefinitionId="cmis:lastModificationDate">
5965                 <cmis:value>2009-10-19T10:10:01.093-07:00</cmis:value>
5966             </cmis:propertyDateTime>
5967             <cmis:propertyId localName="rep-cmis:baseTypeId"
5968 propertyDefinitionId="cmis:baseTypeId">
5969                 <cmis:value>cmis:document</cmis:value>
5970             </cmis:propertyId>
5971             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
5972 propertyDefinitionId="cmis:lastModifiedBy">
5973                 <cmis:value>Al Brown</cmis:value>
5974             </cmis:propertyString>
5975             <cmis:propertyString localName="rep-cmis:createdBy"
5976 propertyDefinitionId="cmis:createdBy">
5977                 <cmis:value>Al Brown</cmis:value>
5978             </cmis:propertyString>
5979             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
5980 propertyDefinitionId="cmis:isLatestVersion">
5981                 <cmis:value>true</cmis:value>

```

```

5982             </cmis:propertyBoolean>
5983                 <cmis:propertyBoolean localName="rep-
5984 cmis:isVersionSeriesCheckedOut">
5985                     <cmis:value>false</cmis:value>
5986                 </cmis:propertyBoolean>
5987                     <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
5988 propertyDefinitionId="cmis:isMajorVersion">
5989                         <cmis:value>false</cmis:value>
5990                     </cmis:propertyBoolean>
5991                     <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
5992 propertyDefinitionId="cmis:isLatestMajorVersion">
5993                         <cmis:value>false</cmis:value>
5994                     </cmis:propertyBoolean>
5995                     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
5996 propertyDefinitionId="cmis:isImmutable">
5997                         <cmis:value>false</cmis:value>
5998                     </cmis:propertyBoolean>
5999                     <cmis:propertyString localName="rep-cmis:checkinComment"
6000 propertyDefinitionId="cmis:checkinComment">
6001                         <cmis:value>Checkin comment</cmis:value>
6002                     </cmis:propertyString>
6003                     <cmis:propertyString localName="rep-cmis:versionLabel"
6004 propertyDefinitionId="cmis:versionLabel">
6005                         <cmis:value>0.1</cmis:value>
6006                     </cmis:propertyString>
6007                     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6008 propertyDefinitionId="cmis:contentStreamMimeType">
6009                         <cmis:value>text/plain</cmis:value>
6010                     </cmis:propertyString>
6011                     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6012 propertyDefinitionId="cmis:contentStreamFileName">
6013                         <cmis:value>text.txt</cmis:value>
6014                     </cmis:propertyString>
6015                     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6016 propertyDefinitionId="cmis:contentStreamLength">
6017                         <cmis:value>4234</cmis:value>
6018                     </cmis:propertyInteger>
6019                     <cmis:propertyString displayName="Keywords for Document"
6020 localName="keywords" propertyDefinitionId="keywords">
6021                         <cmis:value>document</cmis:value>
6022                         <cmis:value>example</cmis:value>
6023                         <cmis:value>sample</cmis:value>
6024                         <cmis:value>cmis</cmis:value>
6025                     </cmis:propertyString>
6026                 </cmis:properties>
6027             </cmisra:object>
6028         </atom:entry>
6029
6030
6031
6032 Please also see the example documents included with the schema.
6033

```

6034 3.7.5 Types Children Collection

6035 This is a collection described in the service document that contains the types in the repository under the
6036 specified parent type. If no parent type is specified, then the base types are returned in the feed. This
6037 feed does not include any nesting and is a flat feed.

6038 CMIS Services:

6039 GET: getTypeChildren

6040 Media Type: application/atom+xml;type=feed

- 6041
- 6042 Link Relations:
- service: Points to service document containing the CMIS repository. The service document
6044 MUST contain only one workspace element.
 - Media Type: application/atomsrv+xml
 - via: points to the type definition entry whose children represent this feed
 - down: points to the atom feed document representing the descendants collection for this same
6047 type with media type of application/cmistree+xml
 - paging link relations as appropriate: first, next, previous, last
 - up: points to the parent type definition
 - If this is a children feed for a base object type, this link is not present.

6051

6052 This feed contains a set of atom entries for each child type definition.

6054

6055 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6056
- cmisra:numItems

6057

6058 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6059
- cmisra:type inside atom:entry

6060

6061

6062 **3.7.5.1 GET**

6063 The following arguments may be supplied. Please see the domain model for more information:

- 6064
- includePropertyDefinitions
 - maxItems
 - skipCount
 - typeId

6068 **3.8 Collections**

6069 **3.8.1 Relationships Collection**

6070 This is the set of relationships available (either source or target or both) from a specific item such as a
6071 document, folder or policy.

6072 CMIS Services:

6073 GET: getObjectRelationships

6074 POST: createRelationship

6075 Media Type: application/atom+xml;type=feed

6076 Accept:

- MUST support Atom Entry Documents with CMIS extensions
 - application/atom+xml;type=entry or
 - application/cmisatom+xml
- MAY support other media type

- 6081
6082 Link Relations:
- 6083 • service: Points to service document containing the CMIS repository. The service document
6084 MUST contain only one workspace element.
 - 6085 ○ Media Type: application/atomsrv+xml
 - 6086 • paging link relations as appropriate: first, next, previous, last

6087
6088 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6089 • cmisra:numItems

6090
6091 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6092 • cmisra:object inside atom:entry

6093

6094 **3.8.1.1 GET**

6095 The following arguments may be supplied. Please see the domain model for more information:

- 6096 • typeId
- 6097 • includeSubRelationshipTypes
- 6098 • relationshipDirection
- 6099 • maxItems
- 6100 • skipCount
- 6101 • filter
- 6102 • includeAllowableActions

6103 **3.8.1.2 POST**

6104 When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new
6105 CMIS relationship, then that relationship will be created.

6106 The server MUST return the appropriate HTTP status code if the source is different than the sourceld or
6107 target different than the targetId for the source and targets specified in this collection.

6108 The server MUST return the appropriate status code if the cmis:objectTypeId is not specified.

6109

6110 Example client request:

```
6111 POST /relationships/source/76a5d5bc-cdfe-42ac-9fde-dd09ad384a71 HTTP/1.1
6112 Host: example.org
6113 Content-Length: 1432
6114 Content-Type: application/atom+xml;type=entry
6115
6116
6117 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6118 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6119 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6120 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6121 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6122 open.org/ns/cmis/restatom/200908/">
6123   <atom:author>
6124     <atom:name>Al Brown</atom:name>
6125   </atom:author>
6126   <atom:content src="http://cmisexample.oasis-open.org/rep1/235d5fdc-6193-
6127 4ec0-8a78-c0da832a06d9"/>
```

```

6128     <atom:id>urn:uuid:235d5fdc-6193-4ec0-8a78-c0da832a06d9</atom:id>
6129     <atom:title type="text">New Relationship</atom:title>
6130     <atom:updated>2009-10-19T10:09:59.796-07:00</atom:updated>
6131     <cmisra:object>
6132         <cmis:properties>
6133             <cmis:propertyId localName="rep-cmis:objectTypeId"
6134             propertyDefinitionId="cmis:objectTypeId">
6135                 <cmis:value>customerRelationships</cmis:value>
6136                 </cmis:propertyId>
6137                 <cmis:propertyId localName="rep-cmis:sourceId"
6138             propertyDefinitionId="cmis:sourceId">
6139                 <cmis:value>76a5d5bc-cdfe-42ac-9fde-dd09ad384a71</cmis:value>
6140                 </cmis:propertyId>
6141                 <cmis:propertyId localName="rep-cmis:targetId"
6142             propertyDefinitionId="cmis:targetId">
6143                 <cmis:value>5f3afe79-1e78-4ff9-a3bf-782c5a492e11</cmis:value>
6144                 </cmis:propertyId>
6145             </cmis:properties>
6146         </cmisra:object>
6147     </atom:entry>
6148

```

6149

6150 Example server response:

```

6151     HTTP/1.1 201 Created
6152     Date: Mon, 19 Oct 2009 10:09:59 -0700
6153     Content-Length: 4684
6154     Content-Type: application/atom+xml;type=entry
6155     Content-Location: http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-
6156     8cd2-30900d7a73a6
6157     Location: http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-8cd2-
6158     30900d7a73a6
6159
6160
6161     <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6162     <atom:entry xmlns:app="http://www.w3.org/2007/app"
6163     xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6164     open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6165     open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6166     open.org/ns/cmis/restatom/200908/">
6167         <atom:author>
6168             <atom:name>Al Brown</atom:name>
6169             <atom:uri>http://www.ibm.com/</atom:uri>
6170             <atom:email>albertcbrown@us.ibm.com</atom:email>
6171         </atom:author>
6172         <atom:content src="http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-
6173     4e5c-8cd2-30900d7a73a6"/>
6174             <atom:id>urn:uuid:5f3287a5-7442-4e5c-8cd2-30900d7a73a6</atom:id>
6175             <atom:title type="text">New Relationship</atom:title>
6176             <atom:updated>2009-10-19T10:09:59.812-07:00</atom:updated>
6177             <atom:link rel="self" href="http://cmisexample.oasis-
6178     open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6"/>
6179             <atom:link rel="edit" href="http://cmisexample.oasis-
6180     open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6"/>
6181             <atom:link type="application/cmis+xml;type=allowableActions"
6182     rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6183     href="http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-8cd2-
6184     30900d7a73a6/allowableactions"/>
6185             <atom:link type="application/atom+xml;type=entry" rel="describedby"
6186     href="http://cmisexample.oasis-open.org/rep1/5f3287a5-7442-4e5c-8cd2-
6187     30900d7a73a6/type"/>
6188             <atom:link type="application/atomsvc+xml" rel="service"
6189     href="http://cmisexample.oasis-open.org/rep1//service"/>
6190             <atom:published>2009-10-19T10:09:59.812-07:00</atom:published>
```

```

6191      <atom:summary type="html">HTML summary of Entry 5f3287a5-7442-4e5c-8cd2-
6192      30900d7a73a6</atom:summary>
6193      <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
6194      open.org/ns/cmis/link/200908/source" href="http://cmisexample.oasis-
6195      open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/source"/>
6196      <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
6197      open.org/ns/cmis/link/200908/target" href="http://cmisexample.oasis-
6198      open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/target"/>
6199      <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6200      open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
6201      open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/policies"/>
6202      <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6203      open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
6204      open.org/rep1/5f3287a5-7442-4e5c-8cd2-30900d7a73a6/acl"/>
6205      <cmisra:object>
6206          <cmis:properties>
6207              <cmis:propertyId localName="rep-cmis:objectId"
6208                  propertyDefinitionId="cmis:objectId">
6209                  <cmis:value>5f3287a5-7442-4e5c-8cd2-30900d7a73a6</cmis:value>
6210                  </cmis:propertyId>
6211                  <cmis:propertyId localName="rep-cmis:objectTypeId"
6212                      propertyDefinitionId="cmis:objectTypeId">
6213                      <cmis:value>customerRelationships</cmis:value>
6214                      </cmis:propertyId>
6215                      <cmis:propertyString localName="rep-cmis:name"
6216                          propertyDefinitionId="cmis:name">
6217                          <cmis:value>New Relationship</cmis:value>
6218                          </cmis:propertyString>
6219                          <cmis:propertyDateTime localName="rep-cmis:creationDate"
6220                              propertyDefinitionId="cmis:creationDate">
6221                              <cmis:value>2009-10-19T10:09:59.812-07:00</cmis:value>
6222                              </cmis:propertyDateTime>
6223                              <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6224                                  propertyDefinitionId="cmis:lastModificationDate">
6225                                  <cmis:value>2009-10-19T10:09:59.812-07:00</cmis:value>
6226                                  </cmis:propertyDateTime>
6227                                  <cmis:propertyId localName="rep-cmis:baseTypeId"
6228                                      propertyDefinitionId="cmis:baseTypeId">
6229                                      <cmis:value>cmis:relationship</cmis:value>
6230                                      </cmis:propertyId>
6231                                      <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6232                                          propertyDefinitionId="cmis:lastModifiedBy">
6233                                          <cmis:value>Al Brown</cmis:value>
6234                                          </cmis:propertyString>
6235                                          <cmis:propertyString localName="rep-cmis:createdBy"
6236                                              propertyDefinitionId="cmis:createdBy">
6237                                              <cmis:value>Al Brown</cmis:value>
6238                                              </cmis:propertyString>
6239                                              <cmis:propertyId localName="rep-cmis:sourceId"
6240                                                  propertyDefinitionId="cmis:sourceId">
6241                                                  <cmis:value>36f25b8d-c920-4d1f-86ab-3fb8d7ea5f97</cmis:value>
6242                                                  </cmis:propertyId>
6243                                                  <cmis:propertyId localName="rep-cmis:targetId"
6244                                          propertyDefinitionId="cmis:targetId">
6245                                          <cmis:value>3f4b0e37-a49d-4bf0-b71d-f8ed0c865029</cmis:value>
6246                                          </cmis:propertyId>
6247                                          </cmis:properties>
6248          </cmisra:object>
6249      </atom:entry>
6250
6251
6252 Please also see the example documents included with the schema.
6253
```

6254 **3.8.2 Folder Children Collection**

6255 This is a collection comprised of all the direct children of a particular folder represented as a feed.

6256 CMIS Services:

6257 GET: getChildren

6258 POST:

6259 createDocument

6260 or createFolder

6261 or createPolicy

6262 or moveObject

6263 or addObjectToFolder

6264

6265 Media Type: application/atom+xml;type=feed

6266

6267 Accept:

6268 • MUST support Atom Entry Documents with CMIS extensions

6269 • MAY support other media type

6270

6271 Link Relations:

6272 • service: Points to service document containing the CMIS repository. The service document
6273 MUST contain only one workspace element.

6274 ◦ Media Type: application/atomsrv+xml

6275 • via: points to the atom entry of the folder generating this collection

6276 • up: points to the atom entry document for this folder's parent

6277 ◦ If the root folder, this link relation MUST NOT be included.

6278 ◦ Media Type: application/atom+xml;type=entry

6279 • down: points to the atom feed document representing the descendants feed with a media type of
6280 application/cmistree+xml

6281 ◦ If a repository does not support capabilityGetDescendants, then this link SHOULD NOT
6282 be included.

6283 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder.
6284 This is represented as a feed with CMIS hierarchy extensions.

6285 ◦ Media Type: application/atom+xml;type=feed

6286 • paging link relations as appropriate: first, next, previous, last

6287

6288 The following CMIS Atom extension element MAY be included inside the atom feed:

6289 • cmisra:numItems

6290

6291 The following CMIS Atom extension element MUST be included inside the atom entries:

6292 • cmisra:object inside atom:entry

6293 • cmisra:pathSegment inside atom:entry

6294

6295 **3.8.2.1 GET**

6296 HTTP Code:

- 200 OK (Success)

6299 The following arguments may be supplied. Please see the domain model for more information:

- maxItems
- skipCount
- filter
- includeAllowableActions
- includeRelationships
- renditionFilter
 - If specified, renditions will be returned as links with relation alternate.
- orderBy
- includePathSegment

6309 **3.8.2.2 POST**

6310 CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please
6311 see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- HTTP Success: 201
- Location Header

6314

6315 The following arguments MAY be supplied.

- sourceFolderId: This parameter indicates the folder from which the object shall be moved from to the current specified folder. This parameter is not allowed for create operations.
 - If specified moveObject will be performed.
 - If not specified, addObjectToFolder will be performed.
- versioningState: The optional argument versioningState MAY specify additional versioning behavior such as checkIn as major or minor. Please see CMIS Domain Model for more information on this parameter.

6323

6324 POSTing an Atom Entry document with CMIS markup:

6325 Adding a document to a folder:

6326 If the atom entry has a cmis property cmis:objectId that is valid for the repository, the object will
6327 be added to the folder.

6328

6329 When an object is added to the folder, in repositories that do not support multi-filing it will be
6330 removed from the previous folder and the operation treated as move. If the repository supports
6331 multiple folders, it will be added to the new folder.

6332 If the optional argument sourceFolderId is specified, then the object will be removed from the
6333 folder specified.

6334

6335 Example client request:

```
6336 POST /obj/8a7761eb-2a0c-4400-b515-964308d6cb5e?sourceFolderId=0571b0e4-8043-
6337 4e48-8a89-448206c2b365 HTTP/1.1
6338 Host: example.org
6339 Content-Length: 1227
6340 Content-Type: application/atom+xml;type=entry
```

```

6343 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6344 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6345   xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6346   open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6347   open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6348   open.org/ns/cmis/restatom/200908/">
6349   <atom:author>
6350     <atom:name>Al Brown</atom:name>
6351   </atom:author>
6352   <atom:id>urn:uuid:8a7761eb-2a0c-4400-b515-964308d6cb5e</atom:id>
6353   <atom:title type="text">Document - To Be Moved</atom:title>
6354   <atom:updated>2009-10-19T10:09:59.640-07:00</atom:updated>
6355   <atom:content src="http://cmisexample.oasis-
6356 open.org/rep1/content/8a7761eb-2a0c-4400-b515-964308d6cb5e"/>
6357   <cmisra:object>
6358     <cmis:properties>
6359       <cmis:propertyId localName="rep-cmis:objectId"
6360 propertyDefinitionId="cmis:objectId">
6361         <cmis:value>8a7761eb-2a0c-4400-b515-964308d6cb5e</cmis:value>
6362       </cmis:propertyId>
6363       <cmis:propertyId localName="rep-cmis:objectTypeId"
6364 propertyDefinitionId="cmis:objectTypeId">
6365         <cmis:value>invoice</cmis:value>
6366       </cmis:propertyId>
6367     </cmis:properties>
6368   </cmisra:object>
6369 </atom:entry>
6370

```

6371

6372 Example server response:

```

6373 HTTP/1.1 201 Created
6374 Date: Mon, 19 Oct 2009 10:09:59 -0700
6375 Content-Length: 7213
6376 Content-Type: application/atom+xml;type=entry
6377 Content-Location: http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-
6378 826b-3aa0a8eb8b15
6379 Location: http://cmisexample.oasis-open.org/rep1/69553581-2311-4aa4-826b-
6380 3aa0a8eb8b15
6381
6382
6383 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6384 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6385   xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6386   open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6387   open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6388   open.org/ns/cmis/restatom/200908/">
6389   <atom:author>
6390     <atom:name>Al Brown</atom:name>
6391     <atom:uri>http://www.ibm.com/</atom:uri>
6392     <atom:email>albertcbrown@us.ibm.com</atom:email>
6393   </atom:author>
6394   <atom:content src="http://cmisexample.oasis-open.org/rep1/69553581-2311-
6395 4aa4-826b-3aa0a8eb8b15"/>
6396   <atom:id>urn:uuid:69553581-2311-4aa4-826b-3aa0a8eb8b15</atom:id>
6397   <atom:title type="text">Document - To Be Moved</atom:title>
6398   <atom:updated>2009-10-19T10:09:59.671-07:00</atom:updated>
6399   <atom:link rel="self" href="http://cmisexample.oasis-
6400 open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15"/>
6401     <atom:link rel="edit" href="http://cmisexample.oasis-
6402 open.org/rep1/69553581-2311-4aa4-826b-3aa0a8eb8b15"/>

```

```

6403      <atom:link type="application/cmis+xml;type=allowableActions"
6404        rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6405        href="http://cmisexample.oasis-open.org/repl/69553581-2311-4aa4-826b-
6406        3aa0a8eb8b15/allowableactions"/>
6407        <atom:link type="application/atom+xml;type=entry" rel="describedby"
6408        href="http://cmisexample.oasis-open.org/repl/69553581-2311-4aa4-826b-
6409        3aa0a8eb8b15/type"/>
6410        <atom:link type="application/atomsvc+xml" rel="service"
6411        href="http://cmisexample.oasis-open.org/repl/service"/>
6412        <atom:published>2009-10-19T10:09:59.687-07:00</atom:published>
6413        <atom:summary type="html">HTML summary of Entry 69553581-2311-4aa4-826b-
6414        3aa0a8eb8b15</atom:summary>
6415        <atom:link rel="edit-media" href="http://cmisexample.oasis-
6416        open.org/repl/69553581-2311-4aa4-826b-3aa0a8eb8b15/edit-media"/>
6417        <atom:link rel="alternate" href="http://cmisexample.oasis-
6418        open.org/repl/69553581-2311-4aa4-826b-3aa0a8eb8b15/alternate"/>
6419        <atom:link type="application/atom+xml;type=feed" rel="up"
6420        href="http://cmisexample.oasis-open.org/repl/69553581-2311-4aa4-826b-
6421        3aa0a8eb8b15/parents"/>
6422        <atom:link type="application/atom+xml;type=feed" rel="version-history"
6423        href="http://cmisexample.oasis-open.org/repl/69553581-2311-4aa4-826b-
6424        3aa0a8eb8b15/allversions"/>
6425        <atom:link type="application/atom+xml;type=entry" rel="current-version"
6426        href="http://cmisexample.oasis-open.org/repl/69553581-2311-4aa4-826b-
6427        3aa0a8eb8b15/latest"/>
6428        <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6429        open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
6430        open.org/repl/69553581-2311-4aa4-826b-3aa0a8eb8b15/relationships"/>
6431        <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6432        open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
6433        open.org/repl/69553581-2311-4aa4-826b-3aa0a8eb8b15/policies"/>
6434        <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6435        open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
6436        open.org/repl/69553581-2311-4aa4-826b-3aa0a8eb8b15/acl"/>
6437        <cmisra:object>
6438          <cmis:properties>
6439            <cmis:propertyId localName="rep-cmis:objectId"
6440            propertyDefinitionId="cmis:objectId">
6441              <cmis:value>69553581-2311-4aa4-826b-3aa0a8eb8b15</cmis:value>
6442            </cmis:propertyId>
6443            <cmis:propertyId localName="rep-cmis:objectTypeId"
6444            propertyDefinitionId="cmis:objectTypeId">
6445              <cmis:value>invoice</cmis:value>
6446            </cmis:propertyId>
6447            <cmis:propertyString localName="rep-cmis:name"
6448            propertyDefinitionId="cmis:name">
6449              <cmis:value>Document - To Be Moved</cmis:value>
6450            </cmis:propertyString>
6451            <cmis:propertyDateTime localName="rep-cmis:creationDate"
6452            propertyDefinitionId="cmis:creationDate">
6453              <cmis:value>2009-10-19T10:09:59.687-07:00</cmis:value>
6454            </cmis:propertyDateTime>
6455            <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6456            propertyDefinitionId="cmis:lastModificationDate">
6457              <cmis:value>2009-10-19T10:09:59.687-07:00</cmis:value>
6458            </cmis:propertyDateTime>
6459            <cmis:propertyId localName="rep-cmis:baseTypeId"
6460            propertyDefinitionId="cmis:baseTypeId">
6461              <cmis:value>cmis:document</cmis:value>
6462            </cmis:propertyId>
6463            <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6464            propertyDefinitionId="cmis:lastModifiedBy">
6465              <cmis:value>Al Brown</cmis:value>
6466            </cmis:propertyString>

```

```

6467      <cmis:propertyString localName="rep-cmis:createdBy"
6468      propertyDefinitionId="cmis:createdBy">
6469          <cmis:value>Al Brown</cmis:value>
6470      </cmis:propertyString>
6471      <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6472      propertyDefinitionId="cmis:isLatestVersion">
6473          <cmis:value>true</cmis:value>
6474      </cmis:propertyBoolean>
6475      <cmis:propertyBoolean localName="rep-
6476      cmis:isVersionSeriesCheckedOut"
6477      propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6478          <cmis:value>false</cmis:value>
6479      </cmis:propertyBoolean>
6480      <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6481      propertyDefinitionId="cmis:isMajorVersion">
6482          <cmis:value>false</cmis:value>
6483      </cmis:propertyBoolean>
6484      <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6485      propertyDefinitionId="cmis:isLatestMajorVersion">
6486          <cmis:value>false</cmis:value>
6487      </cmis:propertyBoolean>
6488      <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6489      propertyDefinitionId="cmis:isImmutable">
6490          <cmis:value>false</cmis:value>
6491      </cmis:propertyBoolean>
6492      <cmis:propertyString localName="rep-cmis:checkinComment"
6493      propertyDefinitionId="cmis:checkinComment">
6494          <cmis:value>Checkin comment</cmis:value>
6495      </cmis:propertyString>
6496      <cmis:propertyString localName="rep-cmis:versionLabel"
6497      propertyDefinitionId="cmis:versionLabel">
6498          <cmis:value>0.1</cmis:value>
6499      </cmis:propertyString>
6500      <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6501      propertyDefinitionId="cmis:contentStreamMimeType">
6502          <cmis:value>text/plain</cmis:value>
6503      </cmis:propertyString>
6504      <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6505      propertyDefinitionId="cmis:contentStreamFileName">
6506          <cmis:value>text.txt</cmis:value>
6507      </cmis:propertyString>
6508      <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6509      propertyDefinitionId="cmis:contentStreamLength">
6510          <cmis:value>4234</cmis:value>
6511      </cmis:propertyInteger>
6512      <cmis:propertyString displayName="Keywords for Document"
6513      localName="keywords" propertyDefinitionId="keywords">
6514          <cmis:value>document</cmis:value>
6515          <cmis:value>example</cmis:value>
6516          <cmis:value>sample</cmis:value>
6517          <cmis:value>cmis</cmis:value>
6518      </cmis:propertyString>
6519      </cmis:properties>
6520  </cmisra:object>
6521</atom:entry>
6522
6523
6524 Please also see the example documents included with the schema.
6525
6526 Creating a CMIS Object (in that folder):

```

6527 If the cmis:objectId property is missing, the object will be created and then added to the folder. If
6528 the cmis:objectId property is present but not a valid object Id, the repository MUST return the
6529 appropriate HTTP status code.

6530

6531 For Documents:

6532 If Content Stream is not provided and it is required by the type definition, the repository
6533 MUST return the appropriate HTTP status code.

6534

6535 Content Streams MAY be provided by any of the following mechanisms:

- 6536 o As part of the atom entry via the src attribute on the content element (AtomPub)
 - 6537 ▪ src attribute: Implementers MAY support external references to content
 - 6538 ▪ If the URI in the src attribute is not reachable, then an appropriate http
6539 status code should be returned.
- 6540 o As part of the atom entry inlining via the content element (AtomPub)
 - 6541 ▪ Please see the AtomPub specification RFC5023 for the processing
6542 model of the content element.
- 6543 o If the cmisra:content is provided by the client inside the atom:entry, the
6544 cmisra:content element MUST take precedence over the atom:content element.
6545 (CMIS)
 - 6546 • This element cmisra:content is base64 encoded
- 6547 o At a later time (AtomPub)
 - 6548 ▪ At a later time by replacing the edit-media link with a new content

6549

6550 The optional argument versioningState MAY specify additional versioning behavior such
6551 as checkin.

6552

6553 Example client request:

```
POST /obj/4f833ce6-f53d-4fb5-a268-7c53705a0807 HTTP/1.1
Host: example.org
Content-Length: 1190
Content-Type: application/atom+xml;type=entry

<?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/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
  open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
  open.org/ns/cmis/restatom/200908/">
  <atom:author>
    <atom:name>Al Brown</atom:name>
  </atom:author>
  <atom:id>urn:uuid:4f833ce6-f53d-4fb5-a268-7c53705a0807</atom:id>
  <atom:title type="text">New Invoice</atom:title>
  <atom:updated>2009-10-19T10:09:59.718-07:00</atom:updated>
  <atom:content type="text">this is the content of the new
  document</atom:content>
  <cmisra:object>
    <cmis:properties>
      <cmis:propertyId localName="rep-cmis:objectId"
        propertyDefinitionId="cmis:objectId">
        <cmis:value>4f833ce6-f53d-4fb5-a268-7c53705a0807</cmis:value>
      </cmis:propertyId>
```

```

6580             <cmis:propertyId localName="rep-cmis:objectTypeId"
6581     propertyDefinitionId="cmis:objectId">
6582         <cmis:value>invoice</cmis:value>
6583     </cmis:propertyId>
6584     </cmis:properties>
6585   </cmisra:object>
6586 </atom:entry>
6587

6588

6589             Example server response:
6590
HTTP/1.1 201 Created
6591 Date: Mon, 19 Oct 2009 10:09:59 -0700
6592 Content-Length: 7191
6593 Content-Type: application/atom+xml;type=entry
6594 Content-Location: http://cmisexample.oasis-open.org/repl/8f714453-f9cf-40ae-
6595 8d72-ea72987af212
6596 Location: http://cmisexample.oasis-open.org/repl/8f714453-f9cf-40ae-8d72-
6597 ea72987af212
6598
6599
6600 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6601 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6602 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6603 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6604 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6605 open.org/ns/cmis/restatom/200908/">
6606     <atom:author>
6607         <atom:name>Al Brown</atom:name>
6608         <atom:uri>http://www.ibm.com/</atom:uri>
6609         <atom:email>albertcbrown@us.ibm.com</atom:email>
6610     </atom:author>
6611     <atom:content src="http://cmisexample.oasis-open.org/repl/8f714453-f9cf-
6612 40ae-8d72-ea72987af212"/>
6613     <atom:id>urn:uuid:8f714453-f9cf-40ae-8d72-ea72987af212</atom:id>
6614     <atom:title type="text">New Invoice</atom:title>
6615     <atom:updated>2009-10-19T10:09:59.734-07:00</atom:updated>
6616     <atom:link rel="self" href="http://cmisexample.oasis-
6617 open.org/repl/8f714453-f9cf-40ae-8d72-ea72987af212"/>
6618     <atom:link rel="edit" href="http://cmisexample.oasis-
6619 open.org/repl/8f714453-f9cf-40ae-8d72-ea72987af212"/>
6620     <atom:link type="application/cmis+xml;type=allowableActions"
6621 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6622 href="http://cmisexample.oasis-open.org/repl/8f714453-f9cf-40ae-8d72-
6623 ea72987af212/allowableactions"/>
6624     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6625 href="http://cmisexample.oasis-open.org/repl/8f714453-f9cf-40ae-8d72-
6626 ea72987af212/type"/>
6627     <atom:link type="application/atomsvc+xml" rel="service"
6628 href="http://cmisexample.oasis-open.org/repl//service"/>
6629     <atom:published>2009-10-19T10:09:59.734-07:00</atom:published>
6630     <atom:summary type="html">HTML summary of Entry 8f714453-f9cf-40ae-8d72-
6631 ea72987af212</atom:summary>
6632     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6633 open.org/repl/8f714453-f9cf-40ae-8d72-ea72987af212/edit-media"/>
6634     <atom:link rel="alternate" href="http://cmisexample.oasis-
6635 open.org/repl/8f714453-f9cf-40ae-8d72-ea72987af212/alternate"/>
6636     <atom:link type="application/atom+xml;type=feed" rel="up"
6637 href="http://cmisexample.oasis-open.org/repl/8f714453-f9cf-40ae-8d72-
6638 ea72987af212/parents"/>
6639     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6640 href="http://cmisexample.oasis-open.org/repl/8f714453-f9cf-40ae-8d72-
6641 ea72987af212/allversions"/>

```

```

6642      <atom:link type="application/atom+xml;type=entry" rel="current-version"
6643 href="http://cmisexample.oasis-open.org/rep1/8f714453-f9cf-40ae-8d72-
6644 ea72987af212/latest"/>
6645      <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6646 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
6647 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/relationships"/>
6648      <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6649 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
6650 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/policies"/>
6651      <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6652 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
6653 open.org/rep1/8f714453-f9cf-40ae-8d72-ea72987af212/acl"/>
6654      <cmisra:object>
6655          <cmis:properties>
6656              <cmis:propertyId localName="rep-cmis:objectId"
6657 propertyDefinitionId="cmis:objectId">
6658                  <cmis:value>8f714453-f9cf-40ae-8d72-ea72987af212</cmis:value>
6659              </cmis:propertyId>
6660              <cmis:propertyId localName="rep-cmis:objectTypeId"
6661 propertyDefinitionId="cmis:objectTypeId">
6662                  <cmis:value>invoice</cmis:value>
6663              </cmis:propertyId>
6664              <cmis:propertyString localName="rep-cmis:name"
6665 propertyDefinitionId="cmis:name">
6666                  <cmis:value>New Invoice</cmis:value>
6667              </cmis:propertyString>
6668              <cmis:propertyDateTime localName="rep-cmis:creationDate"
6669 propertyDefinitionId="cmis:creationDate">
6670                  <cmis:value>2009-10-19T10:09:59.734-07:00</cmis:value>
6671              </cmis:propertyDateTime>
6672              <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6673 propertyDefinitionId="cmis:lastModificationDate">
6674                  <cmis:value>2009-10-19T10:09:59.734-07:00</cmis:value>
6675              </cmis:propertyDateTime>
6676              <cmis:propertyId localName="rep-cmis:baseTypeId"
6677 propertyDefinitionId="cmis:baseTypeId">
6678                  <cmis:value>cmis:document</cmis:value>
6679              </cmis:propertyId>
6680              <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6681 propertyDefinitionId="cmis:lastModifiedBy">
6682                  <cmis:value>Al Brown</cmis:value>
6683              </cmis:propertyString>
6684              <cmis:propertyString localName="rep-cmis:createdBy"
6685 propertyDefinitionId="cmis:createdBy">
6686                  <cmis:value>Al Brown</cmis:value>
6687              </cmis:propertyString>
6688              <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6689 propertyDefinitionId="cmis:isLatestVersion">
6690                  <cmis:value>true</cmis:value>
6691              </cmis:propertyBoolean>
6692              <cmis:propertyBoolean localName="rep-
6693 cmis:isVersionSeriesCheckedOut"
6694 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6695                  <cmis:value>false</cmis:value>
6696              </cmis:propertyBoolean>
6697              <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6698 propertyDefinitionId="cmis:isMajorVersion">
6699                  <cmis:value>false</cmis:value>
6700              </cmis:propertyBoolean>
6701              <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6702 propertyDefinitionId="cmis:isLatestMajorVersion">
6703                  <cmis:value>false</cmis:value>
6704              </cmis:propertyBoolean>

```

```

6705             <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6706             propertyDefinitionId="cmis:isImmutable">
6707                 <cmis:value>false</cmis:value>
6708             </cmis:propertyBoolean>
6709             <cmis:propertyString localName="rep-cmis:checkinComment"
6710             propertyDefinitionId="cmis:checkinComment">
6711                 <cmis:value>Checkin comment</cmis:value>
6712             </cmis:propertyString>
6713             <cmis:propertyString localName="rep-cmis:versionLabel"
6714             propertyDefinitionId="cmis:versionLabel">
6715                 <cmis:value>0.1</cmis:value>
6716             </cmis:propertyString>
6717             <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6718             propertyDefinitionId="cmis:contentStreamMimeType">
6719                 <cmis:value>text/plain</cmis:value>
6720             </cmis:propertyString>
6721             <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6722             propertyDefinitionId="cmis:contentStreamFileName">
6723                 <cmis:value>text.txt</cmis:value>
6724             </cmis:propertyString>
6725             <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6726             propertyDefinitionId="cmis:contentStreamLength">
6727                 <cmis:value>4234</cmis:value>
6728             </cmis:propertyInteger>
6729             <cmis:propertyString displayName="Keywords for Document"
6730             localName="keywords" propertyDefinitionId="keywords">
6731                 <cmis:value>document</cmis:value>
6732                 <cmis:value>example</cmis:value>
6733                 <cmis:value>sample</cmis:value>
6734                 <cmis:value>cmis</cmis:value>
6735             </cmis:propertyString>
6736         </cmis:properties>
6737     </cmisra:object>
6738 </atom:entry>
6739

```

6740

6741 Please also see the example documents included with the schema.

6742

6743 POSTing other document formats: (AtomPub)

6744 The behavior is repository specific when a non Atom entry or an atom document without the
6745 CMIS elements is posted to a folder collection.

6746 For example, the repository MAY auto-create a document with a specific type (document) the
6747 client could edit.

6748 If the repository does not support this scenario or another exception occurs, then the repository
6749 MUST return the appropriate HTTP status code.

6750

6751 Optional arguments:

- versioningState (for createDocument)
- sourceFolderId (for moveObject)

6754

6755 3.8.3 Policies Collection

6756 This is an atom feed of all the policy objects currently applied to a specific object

6757 CMIS Services:

6758 GET: getAppliedPolicies

6759 POST: applyPolicy (to object representing this collection of policies)
6760 Media Type: application/atom+xml;type=feed
6761 Accept:

- MUST support Atom Entry Documents with CMIS extensions
 - application/atom+xml;type=entry or
 - application/cmisatom+xml
- MAY support other media type

6766
6767 Link Relations:

- service: Points to service document containing the CMIS repository. The service document MUST contain only one workspace element.
 - Media Type: application/atomsrv+xml
- via: points to the atom entry of the resource generating this collection
- paging link relations as appropriate: first, next, previous, last

6773
6774 The policy entries displayed here are specific to the object generating this collection. A DELETE method
6775 on those URIs will invoke removePolicy().
6776
6777 The following CMIS Atom extension element MAY be included inside the atom feed:

- cmisra:numItems

6779
6780 The following CMIS Atom extension element MUST be included inside the atom entries:

- cmisra:object inside atom:entry

6782

6783 **3.8.3.1 GET**
6784 The following arguments may be supplied. Please see the domain model for more information:

- filter

6786 **3.8.3.2 POST**
6787 When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the
6788 object.
6789
6790 Example client request:
6791 POST /policies/f6fc3972-ed27-4b49-9412-b4ef2b0f715e HTTP/1.1
6792 Host: example.org
6793 Content-Length: 1039
6794 Content-Type: application/atom+xml;type=entry
6795
6796
6797 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6798 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6799 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6800 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6801 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6802 open.org/ns/cmis/restatom/200908/">
6803 <atom:author>
6804 <atom:name>Al Brown</atom:name>

```

6805      </atom:author>
6806      <atom:content src="http://cmisexample.oasis-open.org/rep1/f6fc3972-ed27-
6807        4b49-9412-b4ef2b0f715e"/>
6808      <atom:id>urn:uuid:f6fc3972-ed27-4b49-9412-b4ef2b0f715e</atom:id>
6809      <atom:title type="text">Security Policy for Invoices</atom:title>
6810      <atom:updated>2009-10-19T10:09:59.765-07:00</atom:updated>
6811      <cmisra:object>
6812          <cmis:properties>
6813              <cmis:propertyId localName="rep-cmis:objectId"
6814                propertyDefinitionId="cmis:objectId">
6815                  <cmis:value>f6fc3972-ed27-4b49-9412-b4ef2b0f715e</cmis:value>
6816              </cmis:propertyId>
6817          </cmis:properties>
6818      </cmisra:object>
6819  </atom:entry>
6820
6821 F

```

6822

6823 Example server response:

```

6824 HTTP/1.1 201 Created
6825 Date: Mon, 19 Oct 2009 10:09:59 -0700
6826 Content-Length: 4043
6827 Content-Type: application/atom+xml;type=entry
6828 Content-Location: http://cmisexample.oasis-open.org/rep1/lee01165-98cf-45bf-
6829 9bce-b6238caa5616
6830 Location: http://cmisexample.oasis-open.org/rep1/lee01165-98cf-45bf-9bce-
6831 b6238caa5616
6832
6833
6834 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6835 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6836   xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6837   open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6838   open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6839   open.org/ns/cmis/restatom/200908/">
6840     <atom:author>
6841       <atom:name>Al Brown</atom:name>
6842       <atom:uri>http://www.ibm.com/</atom:uri>
6843       <atom:email>albertcbrown@us.ibm.com</atom:email>
6844     </atom:author>
6845     <atom:content src="http://cmisexample.oasis-open.org/rep1/lee01165-98cf-
6846 45bf-9bce-b6238caa5616"/>
6847     <atom:id>urn:uuid:lee01165-98cf-45bf-9bce-b6238caa5616</atom:id>
6848     <atom:title type="text">Security Policy for Invoices</atom:title>
6849     <atom:updated>2009-10-19T10:09:59.765-07:00</atom:updated>
6850     <atom:link rel="self" href="http://cmisexample.oasis-
6851   open.org/rep1/lee01165-98cf-45bf-9bce-b6238caa5616"/>
6852     <atom:link rel="edit" href="http://cmisexample.oasis-
6853   open.org/rep1/lee01165-98cf-45bf-9bce-b6238caa5616"/>
6854     <atom:link type="application/cmis+xml;type=allowableActions"
6855       rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6856       href="http://cmisexample.oasis-open.org/rep1/lee01165-98cf-45bf-9bce-
6857 b6238caa5616/allowableactions"/>
6858     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6859       href="http://cmisexample.oasis-open.org/rep1/lee01165-98cf-45bf-9bce-
6860 b6238caa5616/type"/>
6861     <atom:link type="application/atomsvc+xml" rel="service"
6862       href="http://cmisexample.oasis-open.org/rep1//service"/>
6863     <atom:published>2009-10-19T10:09:59.781-07:00</atom:published>
6864     <atom:summary type="html">HTML summary of Entry lee01165-98cf-45bf-9bce-
6865 b6238caa5616</atom:summary>

```

```

6866      <atom:link type="application/atom+xml;type=feed" rel="up"
6867      href="http://cmisexample.oasis-open.org/rep1/lee01165-98cf-45bf-9bce-
6868      b6238caa5616/parents"/>
6869          <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6870          open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
6871          open.org/rep1/lee01165-98cf-45bf-9bce-b6238caa5616/relationships"/>
6872              <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6873              open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
6874              open.org/rep1/lee01165-98cf-45bf-9bce-b6238caa5616/acl"/>
6875      <cmisra:object>
6876          <cmis:properties>
6877              <cmis:propertyId localName="rep-cmis:objectId"
6878                  propertyDefinitionId="cmis:objectId">
6879                  <cmis:value>lee01165-98cf-45bf-9bce-b6238caa5616</cmis:value>
6880              </cmis:propertyId>
6881              <cmis:propertyId localName="rep-cmis:objectTypeId"
6882                  propertyDefinitionId="cmis:objectTypeId">
6883                  <cmis:value>generalSecurityPolicy</cmis:value>
6884              </cmis:propertyId>
6885              <cmis:propertyString localName="rep-cmis:name"
6886                  propertyDefinitionId="cmis:name">
6887                  <cmis:value>Security Policy for Invoices</cmis:value>
6888              </cmis:propertyString>
6889              <cmis:propertyDateTime localName="rep-cmis:creationDate"
6890                  propertyDefinitionId="cmis:creationDate">
6891                  <cmis:value>2009-10-19T10:09:59.781-07:00</cmis:value>
6892              </cmis:propertyDateTime>
6893              <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6894                  propertyDefinitionId="cmis:lastModificationDate">
6895                  <cmis:value>2009-10-19T10:09:59.781-07:00</cmis:value>
6896              </cmis:propertyDateTime>
6897              <cmis:propertyId localName="rep-cmis:baseTypeId"
6898                  propertyDefinitionId="cmis:baseTypeId">
6899                  <cmis:value>cmis:policy</cmis:value>
6900              </cmis:propertyId>
6901              <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6902                  propertyDefinitionId="cmis:lastModifiedBy">
6903                  <cmis:value>Al Brown</cmis:value>
6904              </cmis:propertyString>
6905              <cmis:propertyString localName="rep-cmis:createdBy"
6906                  propertyDefinitionId="cmis:createdBy">
6907                  <cmis:value>Al Brown</cmis:value>
6908              </cmis:propertyString>
6909          </cmis:properties>
6910      </cmisra:object>
6911  </atom:entry>
6912
```

6913
6914 Please also see the example documents included with the schema.

6915 **3.9 Feeds**

6916 **3.9.1 Object Parents Feed**

6917 This is the set of parents for a specific object.

6918 CMIS Services:

6919 GET: getObjectParents

6920 Media Type: application/atom+xml;type=feed

6921

6922 Link Relations:

- 6923 • service: Points to service document containing the CMIS repository. The service document
 6924 MUST contain only one workspace element.
 6925 ◦ Media Type: application/atomsrv+xml
 6926 • via: points to the atom entry of object who's parents are represented by this collection

6927 This feed contains a set of atom entries for each parent of the object that MUST contain:

- 6929 • cmisra:object inside atom:entry
 6930 • cmisra:relativePathSegment inside atom:entry for the name of the object inside the folder

6931 Example:

```

6933 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6934 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/" 
6935   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/" 
6936   xmlns:atom="http://www.w3.org/2005/Atom" 
6937   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
6938     open.org/ns/cmis/restatom/200908/">
6939     <atom:title type="text">Parent Feed for 1b43e592-a780-4feb-a0dc-
6940       d25de6bf7ab4</atom:title>
6941       <atom:author>
6942         <atom:name>Al Brown</atom:name>
6943         <atom:uri>http://www.ibm.com/</atom:uri>
6944         <atom:email>albertcbrown@us.ibm.com</atom:email>
6945       </atom:author>
6946       <atom:updated>2009-10-19T10:10:00.562-07:00</atom:updated>
6947       <atom:id>urn:uuid:a8fc16cb-d385-4172-b994-da119636d954</atom:id>
6948       <atom:link type="application/atom+xml;type=feed" rel="self"
6949         href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
6950           d25de6bf7ab4/3"/>
6951       <atom:link type="application/atomsrv+xml" rel="service"
6952         href="http://cmisexample.oasis-open.org/rep1/service"/>
6953         <atom:link type="application/atom+xml;type=entry" rel="via"
6954           href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
6955             d25de6bf7ab4"/>
6956         <atom:link type="application/atom+xml;type=feed" rel="first"
6957           href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
6958             d25de6bf7ab4/first"/>
6959         <atom:link type="application/atom+xml;type=feed" rel="next"
6960           href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
6961             d25de6bf7ab4/4"/>
6962         <atom:link type="application/atom+xml;type=feed" rel="previous"
6963           href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
6964             d25de6bf7ab4/2"/>
6965         <atom:link type="application/atom+xml;type=feed" rel="last"
6966           href="http://cmisexample.oasis-open.org/rep1/1b43e592-a780-4feb-a0dc-
6967             d25de6bf7ab4/last"/>
6968         <cmisra:numItems>1</cmisra:numItems>
6969       <atom:entry>
6970         <atom:author>
6971           <atom:name>Al Brown</atom:name>
6972           <atom:uri>http://www.ibm.com/</atom:uri>
6973           <atom:email>albertcbrown@us.ibm.com</atom:email>
6974         </atom:author>
6975         <atom:content src="http://cmisexample.oasis-open.org/rep1/adb70b59-
6976           db45-4da2-81c0-11062fb063bd"/>
6977           <atom:id>urn:uuid:adb70b59-db45-4da2-81c0-11062fb063bd</atom:id>
6978           <atom:title type="text">Customer Folder</atom:title>
6979           <atom:updated>2009-10-19T10:10:00.562-07:00</atom:updated>
6980           <atom:link rel="self" href="http://cmisexample.oasis-
6981             open.org/rep1/adb70b59-db45-4da2-81c0-11062fb063bd" />
```

```

6982         <atom:link rel="edit" href="http://cmisexample.oasis-
6983 open.org/rep1/adb70b59-db45-4da2-81c0-11062fb063bd"/>
6984         <atom:link type="application/cmis+xml;type=allowableActions"
6985             rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6986             href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
6987             11062fb063bd/allowableactions"/>
6988         <atom:link type="application/atom+xml;type=entry" rel="describedby"
6989             href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
6990             11062fb063bd/type"/>
6991         <atom:link type="application/atomsvc+xml" rel="service"
6992             href="http://cmisexample.oasis-open.org/rep1//service"/>
6993             <atom:published>2009-10-19T10:10:00.562-07:00</atom:published>
6994             <atom:summary type="html">HTML summary of Entry adb70b59-db45-4da2-
6995             81c0-11062fb063bd</atom:summary>
6996             <atom:link type="application/atom+xml;type=entry" rel="up"
6997                 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
6998                 11062fb063bd/up"/>
6999             <atom:link type="application/atom+xml;type=feed" rel="down"
7000                 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7001                 11062fb063bd/children"/>
7002                 <atom:link type="application/cmistree+xml" rel="down"
7003                     href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7004                     11062fb063bd/tree"/>
7005                     <atom:link type="application/atom+xml;type=feed"
7006                         rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7007                         href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7008                         11062fb063bd/foldertree"/>
7009                         <atom:link type="application/atom+xml;type=feed"
7010                             rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7011                             href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7012                             11062fb063bd/relationships"/>
7013                             <atom:link type="application/atom+xml;type=feed"
7014                                 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7015                                 href="http://cmisexample.oasis-open.org/rep1/adb70b59-db45-4da2-81c0-
7016                                 11062fb063bd/policies"/>
7017                                 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7018 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7019 open.org/rep1/adb70b59-db45-4da2-81c0-11062fb063bd/acl"/>
7020             <cmisra:object>
7021                 <cmis:properties>
7022                     <cmis:propertyId localName="rep-cmis:objectId"
7023                         propertyDefinitionId="cmis:objectId">
7024                         <cmis:value>adb70b59-db45-4da2-81c0-
7025                         11062fb063bd</cmis:value>
7026                         </cmis:propertyId>
7027                         <cmis:propertyId localName="rep-cmis:objectTypeId"
7028                             propertyDefinitionId="cmis:objectTypeId">
7029                             <cmis:value>customer</cmis:value>
7030                             </cmis:propertyId>
7031                             <cmis:propertyString localName="rep-cmis:name"
7032                                 propertyDefinitionId="cmis:name">
7033                                 <cmis:value>Customer Folder</cmis:value>
7034                                 </cmis:propertyString>
7035                                 <cmis:propertyDateTime localName="rep-cmis:creationDate"
7036                                     propertyDefinitionId="cmis:creationDate">
7037                                     <cmis:value>2009-10-19T10:10:00.562-07:00</cmis:value>
7038                                     </cmis:propertyDateTime>
7039                                     <cmis:propertyDateTime localName="rep-
7040                                         cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7041                                         <cmis:value>2009-10-19T10:10:00.562-07:00</cmis:value>
7042                                         </cmis:propertyDateTime>
7043                                         <cmis:propertyId localName="rep-cmis:baseTypeId"
7044                                         propertyDefinitionId="cmis:baseTypeId">
7045                                         <cmis:value>cmis:folder</cmis:value>
```

```

7046          </cmis:propertyId>
7047          <cmis:propertyString localName="rep-cmis:lastModifiedBy">
7048          propertyDefinitionId="cmis:lastModifiedBy">
7049              <cmis:value>Al Brown</cmis:value>
7050          </cmis:propertyString>
7051          <cmis:propertyString localName="rep-cmis:createdBy">
7052          propertyDefinitionId="cmis:createdBy">
7053              <cmis:value>Al Brown</cmis:value>
7054          </cmis:propertyString>
7055          <cmis:propertyId localName="rep-cmis:parentId">
7056          propertyDefinitionId="cmis:parentId">
7057              <cmis:value>adb70b59-db45-4da2-81c0-
7058      11062fb063bdup</cmis:value>
7059          </cmis:propertyId>
7060      </cmis:properties>
7061      </cmisra:object>
7062          <cmisra:relativePathSegment>customer1</cmisra:relativePathSegment>
7063      </atom:entry>
7064  </atom:feed>

```

7065

7066 Please also see the example documents included with the schema.

7067 **3.9.1.1 GET**

7068 The following arguments may be supplied. Please see the domain model for more information:

- 7069 • filter
- 7070 • includeAllowableActions
- 7071 • includeRelationships
- 7072 • renditionFilter
- 7073 • includeRelativePathSegment
 - 7074 ○ If true, then the cmisra:relativePathSegment element will be included in the response.

7075 **3.9.2 Changes**

7076 This is a link relationship described in the service document that contains the changes in the repository in
7077 the workspace element. The link relation pointing to this feed is <http://docs.oasis-open.org/ns/cmis/link/200908/changes>.

7079

7080 CMIS Services:

7081 GET: getContentChanges()

7082 Media Type: application/atom+xml;type=feed

7083 Link Relations:

- 7084 • service: Points to service document containing the CMIS repository. The service document
7085 MUST contain only one workspace element.
 - 7086 ○ Media Type: application/atomserv+xml
- 7087 • paging link relations as appropriate: first, next, previous, last
 - 7088 ○ ChangeLogToken is incorporated into the URI specified by the next link relation

7089

7090 This feed MUST be ordered from oldest first to newest.

7091

7092 If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is
7093 not available, the client should revisit the feed in the future and look for new items and the next link
7094 relation.

7095

7096 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7097 • cmisra:numItems

7098

7099 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7100 • cmisra:object inside atom:entry

7101

7102 Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
  xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
  open.org/ns/cmis/restatom/200908/">
  <atom:title type="text">changelog feed</atom:title>
  <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:updated>2009-10-19T10:09:59.953-07:00</atom:updated>
  <atom:id>urn:uuid:04393e94-6888-404e-b9de-765cf769b5de</atom:id>
  <atom:link type="application/atom+xml;type=feed" rel="self"
    href="http://cmisexample.oasis-open.org/rep1/oId/3"/>
    <atom:link type="application/atomsvc+xml" rel="service"
      href="http://cmisexample.oasis-open.org/rep1//service"/>
      <atom:link type="application/atom+xml;type=feed" rel="first"
        href="http://cmisexample.oasis-open.org/rep1/oId/first"/>
        <atom:link type="application/atom+xml;type=feed" rel="next"
          href="http://cmisexample.oasis-open.org/rep1/oId/4"/>
          <atom:link type="application/atom+xml;type=feed" rel="previous"
            href="http://cmisexample.oasis-open.org/rep1/oId/2"/>
            <atom:link type="application/atom+xml;type=feed" rel="last"
              href="http://cmisexample.oasis-open.org/rep1/oId/last"/>
              <cmisra:numItems>2</cmisra:numItems>
              <atom:entry>
                <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/rep1/2ff37143-
      0d0d-4c1a-948d-bd8311261c1f"/>
      <atom:id>urn:uuid:2ff37143-0d0d-4c1a-948d-bd8311261c1f</atom:id>
      <atom:title type="text">CMIS Example Folder as Customer Policy
      type</atom:title>
      <atom:updated>2009-10-19T10:09:59.953-07:00</atom:updated>
      <atom:link rel="self" href="http://cmisexample.oasis-
      open.org/rep1/2ff37143-0d0d-4c1a-948d-bd8311261c1f"/>
      <atom:link rel="edit" href="http://cmisexample.oasis-
      open.org/rep1/2ff37143-0d0d-4c1a-948d-bd8311261c1f"/>
      <atom:link type="application/cmis+xml;type=allowableActions"
        rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
        href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
      bd8311261c1f/allowableactions"/>
```

```

7150      <atom:link type="application/atom+xml;type=entry" rel="describedby"
7151      href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7152      bd8311261clf/type"/>
7153          <atom:link type="application/atomsvc+xml" rel="service"
7154          href="http://cmisexample.oasis-open.org/rep1//service"/>
7155              <atom:published>2009-10-19T10:09:59.953-07:00</atom:published>
7156                  <atom:summary type="html">HTML summary of Entry 2ff37143-0d0d-4c1a-
7157                  948d-bd8311261clf</atom:summary>
7158          <atom:link type="application/atom+xml;type=entry" rel="up"
7159          href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7160          bd8311261clf/up"/>
7161              <atom:link type="application/atom+xml;type=feed" rel="down"
7162              href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7163              bd8311261clf/children"/>
7164                  <atom:link type="application/cmistree+xml" rel="down"
7165                  href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7166                  bd8311261clf/tree"/>
7167                      <atom:link type="application/atom+xml;type=feed"
7168                      rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7169                      href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7170                      bd8311261clf/foldertree"/>
7171                          <atom:link type="application/atom+xml;type=feed"
7172                          rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7173                          href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7174                          bd8311261clf/relationships"/>
7175                              <atom:link type="application/atom+xml;type=feed"
7176                              rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7177                              href="http://cmisexample.oasis-open.org/rep1/2ff37143-0d0d-4c1a-948d-
7178                              bd8311261clf/policies"/>
7179                                  <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7180                                  open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7181                                  open.org/rep1/2ff37143-0d0d-4c1a-948d-bd8311261clf/acl"/>
7182          <cmisra:object>
7183              <cmis:properties>
7184                  <cmis:propertyId localName="rep-cmis:objectId"
7185                  propertyDefinitionId="cmis:objectId">
7186                      <cmis:value>2ff37143-0d0d-4c1a-948d-
7187                      bd8311261clf</cmis:value>
7188                      </cmis:propertyId>
7189                      <cmis:propertyId localName="rep-cmis:objectTypeId"
7190                      propertyDefinitionId="cmis:objectTypeId">
7191                          <cmis:value>customerpolicy</cmis:value>
7192                          </cmis:propertyId>
7193                          <cmis:propertyString localName="rep-cmis:name"
7194                          propertyDefinitionId="cmis:name">
7195                              <cmis:value>CMIS Example Folder as Customer Policy
7196                          type</cmis:value>
7197                              </cmis:propertyString>
7198                              <cmis:propertyDateTime localName="rep-cmis:creationDate"
7199                              propertyDefinitionId="cmis:creationDate">
7200                                  <cmis:value>2009-10-19T10:09:59.953-07:00</cmis:value>
7201                                  </cmis:propertyDateTime>
7202                                  <cmis:propertyDateTime localName="rep-
7203                                  cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7204                                      <cmis:value>2009-10-19T10:09:59.953-07:00</cmis:value>
7205                                      </cmis:propertyDateTime>
7206                                      <cmis:propertyId localName="rep-cmis:baseTypeId"
7207                                      propertyDefinitionId="cmis:baseTypeId">
7208                                          <cmis:value>cmis:folder</cmis:value>
7209                                          </cmis:propertyId>
7210                                          <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7211                                          propertyDefinitionId="cmis:lastModifiedBy">
7212                                              <cmis:value>Al Brown</cmis:value>
7213                                              </cmis:propertyString>
```

```

7214      <cmis:propertyString localName="rep-cmis:createdBy"
7215      propertyDefinitionId="cmis:createdBy">
7216          <cmis:value>Al Brown</cmis:value>
7217          </cmis:propertyString>
7218          <cmis:propertyId localName="rep-cmis:parentId"
7219          propertyDefinitionId="cmis:parentId">
7220              <cmis:value>2ff37143-0d0d-4c1a-948d-
7221      bd8311261clfup</cmis:value>
7222          </cmis:propertyId>
7223          </cmis:properties>
7224          <cmis:changeEventInfo>
7225              <cmis:changeType>updated</cmis:changeType>
7226              <cmis:changeTime>2009-10-19T10:09:59.953-
7227      07:00</cmis:changeTime>
7228          </cmis:changeEventInfo>
7229          </cmisra:object>
7230          <cmisra:pathSegment>policy</cmisra:pathSegment>
7231      </atom:entry>
7232      <atom:entry>
7233          <atom:author>
7234              <atom:name>Al Brown</atom:name>
7235              <atom:uri>http://www.ibm.com/</atom:uri>
7236              <atom:email>albertcbrown@us.ibm.com</atom:email>
7237          </atom:author>
7238          <atom:content src="http://cmisexample.oasis-open.org/repl/95715cc8-
7239      5f9c-45ce-8c54-4c2d8d7c44fc"/>
7240          <atom:id>urn:uuid:95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc</atom:id>
7241          <atom:title type="text">CMIS Example Document</atom:title>
7242          <atom:updated>2009-10-19T10:09:59.968-07:00</atom:updated>
7243          <atom:link rel="self" href="http://cmisexample.oasis-
7244      open.org/repl/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc"/>
7245          <atom:link rel="edit" href="http://cmisexample.oasis-
7246      open.org/repl/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc"/>
7247          <atom:link type="application/cmis+xml;type=allowableActions"
7248      rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7249      href="http://cmisexample.oasis-open.org/repl/95715cc8-5f9c-45ce-8c54-
7250      4c2d8d7c44fc/allowableactions"/>
7251          <atom:link type="application/atom+xml;type=entry" rel="describedby"
7252      href="http://cmisexample.oasis-open.org/repl/95715cc8-5f9c-45ce-8c54-
7253      4c2d8d7c44fc/type"/>
7254          <atom:link type="application/atomsvc+xml" rel="service"
7255      href="http://cmisexample.oasis-open.org/repl//service"/>
7256          <atom:published>2009-10-19T10:09:59.968-07:00</atom:published>
7257          <atom:summary type="html">HTML summary of Entry 95715cc8-5f9c-45ce-
7258      8c54-4c2d8d7c44fc</atom:summary>
7259          <atom:link rel="edit-media" href="http://cmisexample.oasis-
7260      open.org/repl/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc/edit-media"/>
7261          <atom:link rel="alternate" href="http://cmisexample.oasis-
7262      open.org/repl/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc/alternate"/>
7263          <atom:link type="application/atom+xml;type=feed" rel="up"
7264      href="http://cmisexample.oasis-open.org/repl/95715cc8-5f9c-45ce-8c54-
7265      4c2d8d7c44fc/parents"/>
7266          <atom:link type="application/atom+xml;type=feed" rel="version-history"
7267      href="http://cmisexample.oasis-open.org/repl/95715cc8-5f9c-45ce-8c54-
7268      4c2d8d7c44fc/allversions"/>
7269          <atom:link type="application/atom+xml;type=entry" rel="current-
7270      version" href="http://cmisexample.oasis-open.org/repl/95715cc8-5f9c-45ce-8c54-
7271      4c2d8d7c44fc/latest"/>
7272          <atom:link type="application/atom+xml;type=feed"
7273      rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7274      href="http://cmisexample.oasis-open.org/repl/95715cc8-5f9c-45ce-8c54-
7275      4c2d8d7c44fc/relationships"/>

```

```

7276      <atom:link type="application/atom+xml;type=feed"
7277      rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7278      href="http://cmisexample.oasis-open.org/rep1/95715cc8-5f9c-45ce-8c54-
7279      4c2d8d7c44fc/policies"/>
7280          <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7281          open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7282          open.org/rep1/95715cc8-5f9c-45ce-8c54-4c2d8d7c44fc/acl"/>
7283              <cmisra:object>
7284                  <cmis:properties>
7285                      <cmis:propertyId localName="rep-cmis:objectId"
7286                      propertyDefinitionId="cmis:objectId">
7287                          <cmis:value>95715cc8-5f9c-45ce-8c54-
7288                          4c2d8d7c44fc</cmis:value>
7289                      </cmis:propertyId>
7290                      <cmis:propertyId localName="rep-cmis:objectTypeId"
7291                      propertyDefinitionId="cmis:objectId">
7292                          <cmis:value>document</cmis:value>
7293                      </cmis:propertyId>
7294                      <cmis:propertyString localName="rep-cmis:name"
7295                      propertyDefinitionId="cmis:name">
7296                          <cmis:value>CMIS Example Document</cmis:value>
7297                      </cmis:propertyString>
7298                      <cmis:propertyDateTime localName="rep-cmis:creationDate"
7299                      propertyDefinitionId="cmis:creationDate">
7300                          <cmis:value>2009-10-19T10:09:59.968-07:00</cmis:value>
7301                      </cmis:propertyDateTime>
7302                      <cmis:propertyDateTime localName="rep-
7303                      cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7304                          <cmis:value>2009-10-19T10:09:59.968-07:00</cmis:value>
7305                      </cmis:propertyDateTime>
7306                      <cmis:propertyId localName="rep-cmis:baseTypeId"
7307                      propertyDefinitionId="cmis:baseTypeId">
7308                          <cmis:value>cmis:document</cmis:value>
7309                      </cmis:propertyId>
7310                      <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7311                      propertyDefinitionId="cmis:lastModifiedBy">
7312                          <cmis:value>Al Brown</cmis:value>
7313                      </cmis:propertyString>
7314                      <cmis:propertyString localName="rep-cmis:createdBy"
7315                      propertyDefinitionId="cmis:createdBy">
7316                          <cmis:value>Al Brown</cmis:value>
7317                      </cmis:propertyString>
7318                      <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
7319                      propertyDefinitionId="cmis:isLatestVersion">
7320                          <cmis:value>true</cmis:value>
7321                      </cmis:propertyBoolean>
7322                      <cmis:propertyBoolean localName="rep-
7323                      cmis:isVersionSeriesCheckedOut"
7324                      propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7325                          <cmis:value>false</cmis:value>
7326                      </cmis:propertyBoolean>
7327                      <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
7328                      propertyDefinitionId="cmis:isMajorVersion">
7329                          <cmis:value>false</cmis:value>
7330                      </cmis:propertyBoolean>
7331                      <cmis:propertyBoolean localName="rep-
7332                      cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
7333                          <cmis:value>false</cmis:value>
7334                      </cmis:propertyBoolean>
7335                      <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7336                      propertyDefinitionId="cmis:isImmutable">
7337                          <cmis:value>false</cmis:value>
7338                      </cmis:propertyBoolean>
```

```

7339 <cmis:propertyString localName="rep-cmis:checkinComment"
7340   propertyDefinitionId="cmis:checkinComment">
7341     <cmis:value>Checkin comment</cmis:value>
7342   </cmis:propertyString>
7343   <cmis:propertyString localName="rep-cmis:versionLabel"
7344     propertyDefinitionId="cmis:versionLabel">
7345       <cmis:value>0.1</cmis:value>
7346     </cmis:propertyString>
7347     <cmis:propertyString localName="rep-
7348 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
7349       <cmis:value>text/plain</cmis:value>
7350     </cmis:propertyString>
7351     <cmis:propertyString localName="rep-
7352 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
7353       <cmis:value>text.txt</cmis:value>
7354     </cmis:propertyString>
7355     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7356     propertyDefinitionId="cmis:contentStreamLength">
7357       <cmis:value>4234</cmis:value>
7358     </cmis:propertyInteger>
7359     <cmis:propertyString displayName="Keywords for Document"
7360     localName="keywords" propertyDefinitionId="keywords">
7361       <cmis:value>document</cmis:value>
7362       <cmis:value>example</cmis:value>
7363       <cmis:value>sample</cmis:value>
7364       <cmis:value>cmis</cmis:value>
7365     </cmis:propertyString>
7366   </cmis:properties>
7367   <cmis:changeEventInfo>
7368     <cmis:changeType>updated</cmis:changeType>
7369     <cmis:changeTime>2009-10-19T10:09:59.968-
7370     07:00</cmis:changeTime>
7371     </cmis:changeEventInfo>
7372   </cmisra:object>
7373   <cmisra:pathSegment>invoice.pdf</cmisra:pathSegment>
7374 </atom:entry>
7375 </atom:feed>

```

7376

7377 Please also see the example documents included with the schema.

7378 **3.9.2.1 GET**

7379 The following optional parameters may be supplied:

- 7380 • filter
- 7381 • maxItems
- 7382 • includeACL
- 7383 • includePolicyIds
- 7384 • includeProperties
- 7385 • filter

7386 **3.9.3 Folder Descendants**

7387 This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available
7388 via the link relation down with the application/cmisTree+xml media type. Please see the Hierarchical Atom
7389 Entries for more information on format.

7390

7391 If a repository does not support capabilityGetDescendants, then these resources SHOULD NOT be
7392 exposed.

7393

7394 CMIS Services:

7395 GET: getDescendants

7396 DELETE: deleteTree

7397 Media Type: application/atom+xml;type=feed

7398

7399 Link Relations:

- service: Points to service document containing the CMIS repository. The service document MUST contain only one workspace element.
 - Media Type: application/atomsvc+xml
- via: points to the atom entry of the folder generating this collection
- up: points to the atom entry document for this folder's parent
 - Media Type: application/atom+xml;type=entry
 - If the root folder, this link relation MUST not be included.
- down:
 - points to the atom feed document representing the children feed for this same folder with media type of application/atom+xml;type=entry
 - Since this is the descendants, the descendants link SHOULD NOT be included
- paging link relations MAY be included as appropriate: first, next, previous, last
 - Repositories may support these paging link relations on a particular cmisra:children element.
- <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

7400

7401

7402

7403

7404

7405

7406

7407

7408

7409

7410

7411

7412

7413

7414

7415

7416 The following CMIS Atom extension element MAY be included inside the atom feed:

7417 • cmisra:numItems

7418

7419 The following CMIS Atom extension element MUST be included inside the atom entries:

7420 • cmisra:object inside atom:entry

7421 • cmisra:pathSegment inside atom:entry

7422 • cmisra:children inside atom:entry

7423

7424 Example:

```

7425 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7426 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/" 
7427   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/" 
7428   xmlns:atom="http://www.w3.org/2005/Atom" 
7429   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7430     open.org/ns/cmis/restatom/200908/">
7431   <atom:title type="text">Feed for folder1</atom:title>
7432   <atom:author>
7433     <atom:name>Al Brown</atom:name>
7434     <atom:uri>http://www.ibm.com/</atom:uri>
7435     <atom:email>albertcbrown@us.ibm.com</atom:email>
7436   </atom:author>
7437   <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7438   <atom:id>urn:uuid:83638fc5-90e7-4d03-bd04-1a4cc5c5c98e</atom:id>
```

```

7439      <atom:link type="application/atom+xml;type=feed" rel="self"
7440 href="http://cmisexample.oasis-open.org/rep1/80dfb434-883b-4850-871d-
7441 6524c1ec2a64/3"/>
7442      <atom:link type="application/atomsvc+xml" rel="service"
7443 href="http://cmisexample.oasis-open.org/rep1//service"/>
7444      <atom:link type="application/atom+xml;type=entry" rel="via"
7445 href="http://cmisexample.oasis-open.org/rep1/80dfb434-883b-4850-871d-
7446 6524c1ec2a64"/>
7447      <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7448 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
7449 open.org/rep1/80dfb434-883b-4850-871d-6524c1ec2a64/foldertree"/>
7450      <atom:link type="application/atom+xml;type=feed" rel="down"
7451 href="http://cmisexample.oasis-open.org/rep1/80dfb434-883b-4850-871d-
7452 6524c1ec2a64/children"/>
7453      <atom:link type="application/atom+xml;type=entry" rel="up"
7454 href="http://cmisexample.oasis-open.org/rep1/ba8910f0-0e99-4cda-a1f6-
7455 befce0fa20e7"/>
7456      <cmisra:numItems>1</cmisra:numItems>
7457      <atom:entry>
7458          <atom:author>
7459              <atom:name>Al Brown</atom:name>
7460              <atom:uri>http://www.ibm.com/</atom:uri>
7461              <atom:email>albertcbrown@us.ibm.com</atom:email>
7462          </atom:author>
7463          <atom:content src="http://cmisexample.oasis-open.org/rep1/c2e574a0-
7464 2d30-4834-b8a3-6333b33181c3"/>
7465          <atom:id>urn:uuid:c2e574a0-2d30-4834-b8a3-6333b33181c3</atom:id>
7466          <atom:title type="text">CMIS Example Folder as Customer
7467 type</atom:title>
7468          <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7469          <atom:link rel="self" href="http://cmisexample.oasis-
7470 open.org/rep1/c2e574a0-2d30-4834-b8a3-6333b33181c3"/>
7471          <atom:link rel="edit" href="http://cmisexample.oasis-
7472 open.org/rep1/c2e574a0-2d30-4834-b8a3-6333b33181c3"/>
7473          <atom:link type="application/cmis+xml;type=allowableActions"
7474 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7475 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7476 6333b33181c3/allowableactions"/>
7477          <atom:link type="application/atom+xml;type=entry" rel="describedby"
7478 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7479 6333b33181c3/type"/>
7480          <atom:link type="application/atomsvc+xml" rel="service"
7481 href="http://cmisexample.oasis-open.org/rep1//service"/>
7482          <atom:published>2009-10-19T10:10:00.203-07:00</atom:published>
7483          <atom:summary type="html">HTML summary of Entry c2e574a0-2d30-4834-
7484 b8a3-6333b33181c3</atom:summary>
7485          <atom:link type="application/atom+xml;type=entry" rel="up"
7486 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7487 6333b33181c3/up"/>
7488          <atom:link type="application/atom+xml;type=feed" rel="down"
7489 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7490 6333b33181c3/children"/>
7491          <atom:link type="application/cmistree+xml" rel="down"
7492 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7493 6333b33181c3/tree"/>
7494          <atom:link type="application/atom+xml;type=feed"
7495 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7496 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7497 6333b33181c3/foldertree"/>
7498          <atom:link type="application/atom+xml;type=feed"
7499 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7500 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7501 6333b33181c3/relationships"/>

```

```

7502      <atom:link type="application/atom+xml;type=feed"
7503      rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7504      href="http://cmisexample.oasis-open.org/repl/c2e574a0-2d30-4834-b8a3-
7505      6333b33181c3/policies"/>
7506          <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7507          open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7508          open.org/repl/c2e574a0-2d30-4834-b8a3-6333b33181c3/acl"/>
7509              <cmisra:object>
7510                  <cmis:properties>
7511                      <cmis:propertyId localName="rep-cmis:objectId"
7512                      propertyDefinitionId="cmis:objectId">
7513                          <cmis:value>c2e574a0-2d30-4834-b8a3-
7514                          6333b33181c3</cmis:value>
7515                      </cmis:propertyId>
7516                      <cmis:propertyId localName="rep-cmis:objectTypeId"
7517                      propertyDefinitionId="cmis:objectTypeId">
7518                          <cmis:value>customer</cmis:value>
7519                      </cmis:propertyId>
7520                      <cmis:propertyString localName="rep-cmis:name"
7521                      propertyDefinitionId="cmis:name">
7522                          <cmis:value>CMIS Example Folder as Customer
7523                      type</cmis:value>
7524                          </cmis:propertyString>
7525                          <cmis:propertyDateTime localName="rep-cmis:creationDate"
7526                          propertyDefinitionId="cmis:creationDate">
7527                              <cmis:value>2009-10-19T10:10:00.203-07:00</cmis:value>
7528                          </cmis:propertyDateTime>
7529                          <cmis:propertyDateTime localName="rep-
7530                          cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7531                              <cmis:value>2009-10-19T10:10:00.203-07:00</cmis:value>
7532                          </cmis:propertyDateTime>
7533                          <cmis:propertyId localName="rep-cmis:baseTypeId"
7534                          propertyDefinitionId="cmis:baseTypeId">
7535                              <cmis:value>cmis:folder</cmis:value>
7536                          </cmis:propertyId>
7537                          <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7538                          propertyDefinitionId="cmis:lastModifiedBy">
7539                              <cmis:value>Al Brown</cmis:value>
7540                          </cmis:propertyString>
7541                          <cmis:propertyString localName="rep-cmis:createdBy"
7542                          propertyDefinitionId="cmis:createdBy">
7543                              <cmis:value>Al Brown</cmis:value>
7544                          </cmis:propertyString>
7545                          <cmis:propertyId localName="rep-cmis:parentId"
7546                          propertyDefinitionId="cmis:parentId">
7547                              <cmis:value>c2e574a0-2d30-4834-b8a3-
7548                              6333b33181c3up</cmis:value>
7549                                  </cmis:propertyId>
7550                                  </cmis:properties>
7551                              </cmisra:object>
7552                              <cmisra:pathSegment>customer</cmisra:pathSegment>
7553                              <cmisra:children>
7554                                  <atom:title type="text">CMIS Example Folder as Customer
7555                              type</atom:title>
7556                                  <atom:author>
7557                                      <atom:name>Al Brown</atom:name>
7558                                      <atom:uri>http://www.ibm.com/</atom:uri>
7559                                      <atom:email>albertcbrown@us.ibm.com</atom:email>
7560                                  </atom:author>
7561                                  <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7562                                  <atom:id>urn:uuid:9550c57d-64cd-4f00-9ab4-5b44903717e8</atom:id>
7563                                  <atom:link type="application/atom+xml;type=feed" rel="self"
7564 href="http://cmisexample.oasis-open.org/repl/c2e574a0-2d30-4834-b8a3-
7565 6333b33181c3/3"/>
```

```

7566      <atom:link type="application/atomsvc+xml" rel="service"
7567 href="http://cmisexample.oasis-open.org/rep1//service"/>
7568      <atom:link type="application/atom+xml;type=entry" rel="via"
7569 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7570 6333b33181c3"/>
7571      <atom:link type="application/atom+xml;type=feed"
7572 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7573 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7574 6333b33181c3/foldertree"/>
7575      <atom:link type="application/atom+xml;type=feed" rel="down"
7576 href="http://cmisexample.oasis-open.org/rep1/c2e574a0-2d30-4834-b8a3-
7577 6333b33181c3/children"/>
7578      <atom:link type="application/atom+xml;type=entry" rel="up"
7579 href="http://cmisexample.oasis-open.org/rep1/80dfb434-883b-4850-871d-
7580 6524clec2a64"/>
7581      <cmisra:numItems>1</cmisra:numItems>
7582      <atom:entry>
7583          <atom:author>
7584              <atom:name>Al Brown</atom:name>
7585              <atom:uri>http://www.ibm.com/</atom:uri>
7586              <atom:email>albertcbrown@us.ibm.com</atom:email>
7587          </atom:author>
7588          <atom:content src="http://cmisexample.oasis-
7589 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50"/>
7590          <atom:id>urn:uuid:d4bc9b02-f732-49ce-979a-
7591 da31175a4d50</atom:id>
7592          <atom:title type="text">CMIS Example Doc as Invoice
7593 type</atom:title>
7594          <atom:updated>2009-10-19T10:10:00.203-07:00</atom:updated>
7595          <atom:link rel="self" href="http://cmisexample.oasis-
7596 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50"/>
7597          <atom:link rel="edit" href="http://cmisexample.oasis-
7598 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50"/>
7599          <atom:link type="application/cmis+xml;type=allowableActions"
7600 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7601 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7602 da31175a4d50/allowableactions"/>
7603          <atom:link type="application/atom+xml;type=entry"
7604 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-
7605 49ce-979a-da31175a4d50/type"/>
7606          <atom:link type="application/atomsvc+xml" rel="service"
7607 href="http://cmisexample.oasis-open.org/rep1//service"/>
7608          <atom:published>2009-10-19T10:10:00.203-07:00</atom:published>
7609          <atom:summary type="html">HTML summary of Entry d4bc9b02-
7610 f732-49ce-979a-da31175a4d50</atom:summary>
7611          <atom:link rel="edit-media" href="http://cmisexample.oasis-
7612 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50/edit-media"/>
7613          <atom:link rel="alternate" href="http://cmisexample.oasis-
7614 open.org/rep1/d4bc9b02-f732-49ce-979a-da31175a4d50/alternate"/>
7615          <atom:link type="application/atom+xml;type=feed" rel="up"
7616 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7617 da31175a4d50/parents"/>
7618          <atom:link type="application/atom+xml;type=feed" rel="version-
7619 history" href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7620 da31175a4d50/allversions"/>
7621          <atom:link type="application/atom+xml;type=entry"
7622 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-
7623 f732-49ce-979a-da31175a4d50/latest"/>
7624          <atom:link type="application/atom+xml;type=feed"
7625 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7626 href="http://cmisexample.oasis-open.org/rep1/d4bc9b02-f732-49ce-979a-
7627 da31175a4d50/relationships"/>

```

```

7628      <atom:link type="application/atom+xml;type=feed"
7629      rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7630      href="http://cmisexample.oasis-open.org/repl/d4bc9b02-f732-49ce-979a-
7631      da31175a4d50/policies"/>
7632          <atom:link type="application/cmisacl+xml"
7633          rel="http://docs.oasis-open.org/ns/cmis/link/200908/acl"
7634          href="http://cmisexample.oasis-open.org/repl/d4bc9b02-f732-49ce-979a-
7635          da31175a4d50/acl"/>
7636              <cmisra:object>
7637                  <cmis:properties>
7638                      <cmis:propertyId localName="rep-cmis:objectId"
7639                      propertyDefinitionId="cmis:objectId">
7640                          <cmis:value>d4bc9b02-f732-49ce-979a-
7641                          da31175a4d50</cmis:value>
7642                              </cmis:propertyId>
7643                                  <cmis:propertyId localName="rep-cmis:objectTypeId"
7644                                  propertyDefinitionId="cmis:objectTypeId">
7645                                      <cmis:value>invoice</cmis:value>
7646                                      </cmis:propertyId>
7647                                          <cmis:propertyString localName="rep-cmis:name"
7648                                          propertyDefinitionId="cmis:name">
7649                                              <cmis:value>CMIS Example Doc as Invoice
7650                                              </cmis:value>
7651                                              </cmis:propertyString>
7652                                              <cmis:propertyDateTime localName="rep-
7653                                              cmis:creationDate" propertyDefinitionId="cmis:creationDate">
7654                                              <cmis:value>2009-10-19T10:10:00.218-
7655                                              07:00</cmis:value>
7656                                              </cmis:propertyDateTime>
7657                                              <cmis:propertyDateTime localName="rep-
7658                                              cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7659                                              <cmis:value>2009-10-19T10:10:00.218-
7660                                              07:00</cmis:value>
7661                                              </cmis:propertyDateTime>
7662                                              <cmis:propertyId localName="rep-cmis:baseTypeId"
7663                                              propertyDefinitionId="cmis:baseTypeId">
7664                                              <cmis:value>cmis:document</cmis:value>
7665                                              </cmis:propertyId>
7666                                              <cmis:propertyString localName="rep-
7667                                              cmis:lastModifiedBy" propertyDefinitionId="cmis:lastModifiedBy">
7668                                              <cmis:value>Al Brown</cmis:value>
7669                                              </cmis:propertyString>
7670                                              <cmis:propertyString localName="rep-cmis:createdBy"
7671                                              propertyDefinitionId="cmis:createdBy">
7672                                              <cmis:value>Al Brown</cmis:value>
7673                                              </cmis:propertyString>
7674                                              <cmis:propertyBoolean localName="rep-
7675                                              cmis:isLatestVersion" propertyDefinitionId="cmis:isLatestVersion">
7676                                              <cmis:value>true</cmis:value>
7677                                              </cmis:propertyBoolean>
7678                                              <cmis:propertyBoolean localName="rep-
7679                                              cmis:isVersionSeriesCheckedOut"
7680                                              propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7681                                              <cmis:value>false</cmis:value>
7682                                              </cmis:propertyBoolean>
7683                                              <cmis:propertyBoolean localName="rep-
7684                                              cmis:isMajorVersion" propertyDefinitionId="cmis:isMajorVersion">
7685                                              <cmis:value>false</cmis:value>
7686                                              </cmis:propertyBoolean>
7687                                              <cmis:propertyBoolean localName="rep-
7688                                              cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
7689                                              <cmis:value>false</cmis:value>
7690                                              </cmis:propertyBoolean>
```

```

7691                               <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7692   propertyDefinitionId="cmis:isImmutable">
7693     <cmis:value>false</cmis:value>
7694   </cmis:propertyBoolean>
7695   <cmis:propertyString localName="rep-
7696   cmis:checkinComment" propertyDefinitionId="cmis:checkinComment">
7697     <cmis:value>Checkin comment</cmis:value>
7698   </cmis:propertyString>
7699   <cmis:propertyString localName="rep-cmis:versionLabel"
7700   propertyDefinitionId="cmis:versionLabel">
7701     <cmis:value>0.1</cmis:value>
7702   </cmis:propertyString>
7703   <cmis:propertyString localName="rep-
7704   cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
7705     <cmis:value>text/plain</cmis:value>
7706   </cmis:propertyString>
7707   <cmis:propertyString localName="rep-
7708   cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
7709     <cmis:value>text.txt</cmis:value>
7710   </cmis:propertyString>
7711   <cmis:propertyInteger localName="rep-
7712   cmis:contentStreamLength" propertyDefinitionId="cmis:contentStreamLength">
7713     <cmis:value>4234</cmis:value>
7714   </cmis:propertyInteger>
7715   <cmis:propertyString displayName="Keywords for
7716 Document" localName="keywords" propertyDefinitionId="keywords">
7717     <cmis:value>document</cmis:value>
7718     <cmis:value>example</cmis:value>
7719     <cmis:value>sample</cmis:value>
7720     <cmis:value>cmis</cmis:value>
7721   </cmis:propertyString>
7722 </cmis:properties>
7723 </cmisra:object>
7724 <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
7725 </atom:entry>
7726 </cmisra:children>
7727 </atom:entry>
7728 </atom:feed>

```

7729
7730 Please also see the example documents included with the schema.

7731 **3.9.3.1 GET**

7732 The following arguments may be supplied. Please see the domain model for more information:

- 7733 • filter
- 7734 • depth
- 7735 • includeAllowableActions
- 7736 • includeRelationships
- 7737 • renditionFilter
- 7738 • includePathSegment

7739 **3.9.3.2 DELETE**

7740 This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the
7741 domain model for more information:

- 7742 • continueOnFailure
- 7743 • unfileObjects

7745 Status Code:

- 7746 • 200 OK if successful. Body contains entity describing the status
- 7747 • 202 Accepted, if accepted but deletion not yet taking place
- 7748 • 204 No Content, if successful with no content
- 7749 • 403 Forbidden, if permission is denied
- 7750 • 401 Unauthorized, if not authenticated
- 7751 • 500 Internal Server Error. The body SHOULD contain an entity describing the status

7752

7753 If the delete method does not delete all items, invoking GET with infinite depth on this URI will return the items not deleted. Subsequent DELETE methods can be invoked on this URI.

7755 Note: If the repository does not implement get on this resource, or the canGetDescendants is false, there
7756 is no mechanism to identify the resources that were not removed.

7757 **3.9.4 Folder Tree**

7758 This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link
7759 relation foldertree with media type application/atom+xml;type=feed. Please see the Hierarchical Atom
7760 Entries for more information on format.

7761

7762 CMIS Services:

7763 GET: getFolderTree

7764 DELETE: deleteTree

7765 Media Type: application/atom +xml;type=feed

7766

7767 Link Relations:

- 7768 • service: Points to service document containing the CMIS repository. The service document
7769 MUST contain only one workspace element.
 - 7770 o Media Type: application/atomserv+xml
- 7771 • via: points to the atom entry of the folder generating this collection
- 7772 • up: points to the atom entry document of this folder's parent
 - 7773 o If the root folder, this link relation MUST not be included.
 - 7774 o Media Type: application/atom+xml;type=entry
- 7775 • down:
 - 7776 o application/atom+xml : Points to the atom feed document representing the children feed
7777 for this same folder
 - 7778 o application/cmistree+xml: Points to the descendants feed of the same folder. If a
7779 repository does not support capabilityGetDescendants, then this link SHOULD NOT be
7780 included.
- 7781 • paging link relations MAY be included as appropriate: first, next, previous, last
 - 7782 o Repositories may support these paging link relations on a particular cmisra:children
7783 element.

7784

7785 This feed contains a set of atom entries for each sub-folder in the folder.

7786

7787 The following CMIS Atom extension element MAY be included inside the atom feed:

7788 • cmisra:numItems
 7789
 7790 The following CMIS Atom extension element MUST be included inside the atom entries:
 7791 • cmisra:object inside atom:entry
 7792 • cmisra:pathSegment inside atom:entry
 7793 • cmisra:children inside atom:entry
 7794
 7795 Example:
 7796 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
 7797 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
 7798 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
 7799 xmlns:atom="http://www.w3.org/2005/Atom"
 7800 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
 7801 open.org/ns/cmis/restatom/200908/">
 7802 <atom:title type="text">FolderTree Feed of Folder1</atom:title>
 7803 <atom:author>
 7804 <atom:name>Al Brown</atom:name>
 7805 <atom:uri>http://www.ibm.com/</atom:uri>
 7806 <atom:email>albertcbrown@us.ibm.com</atom:email>
 7807 </atom:author>
 7808 <atom:updated>2009-10-19T10:10:00.312-07:00</atom:updated>
 7809 <atom:id>urn:uuid:e8627e01-9b28-4e5b-b7b4-d88c05bae4e9</atom:id>
 7810 <atom:link type="application/atom+xml;type=feed" rel="self"
 7811 href="http://cmisexample.oasis-open.org/rep1/3bcbead0-ad9f-4e0c-a2e8-
 7812 49d025e062c5/3"/>
 7813 <atom:link type="application/atomsvc+xml" rel="service"
 7814 href="http://cmisexample.oasis-open.org/rep1//service"/>
 7815 <atom:link type="application/atom+xml;type=entry" rel="via"
 7816 href="http://cmisexample.oasis-open.org/rep1/3bcbead0-ad9f-4e0c-a2e8-
 7817 49d025e062c5"/>
 7818 <atom:link type="application/cmistree+xml" rel="down"
 7819 href="http://cmisexample.oasis-open.org/rep1/3bcbead0-ad9f-4e0c-a2e8-
 7820 49d025e062c5/tree"/>
 7821 <atom:link type="application/atom+xml;type=feed" rel="down"
 7822 href="http://cmisexample.oasis-open.org/rep1/3bcbead0-ad9f-4e0c-a2e8-
 7823 49d025e062c5/children"/>
 7824 <atom:link type="application/atom+xml;type=entry" rel="up"
 7825 href="http://cmisexample.oasis-open.org/rep1/cdaf0998-582a-445f-9179-
 7826 113a12470b04"/>
 7827 <cmisra:numItems>1</cmisra:numItems>
 7828 <atom:entry>
 7829 <atom:author>
 7830 <atom:name>Al Brown</atom:name>
 7831 <atom:uri>http://www.ibm.com/</atom:uri>
 7832 <atom:email>albertcbrown@us.ibm.com</atom:email>
 7833 </atom:author>
 7834 <atom:content src="http://cmisexample.oasis-open.org/rep1/7d9c26a2-
 7835 adle-43e5-b176-247b64e4f4bc"/>
 7836 <atom:id>urn:uuid:7d9c26a2-adle-43e5-b176-247b64e4f4bc</atom:id>
 7837 <atom:title type="text">Customer Folder</atom:title>
 7838 <atom:updated>2009-10-19T10:10:00.312-07:00</atom:updated>
 7839 <atom:link rel="self" href="http://cmisexample.oasis-
 7840 open.org/rep1/7d9c26a2-adle-43e5-b176-247b64e4f4bc"/>
 7841 <atom:link rel="edit" href="http://cmisexample.oasis-
 7842 open.org/rep1/7d9c26a2-adle-43e5-b176-247b64e4f4bc"/>
 7843 <atom:link type="application/cmis+xml;type=allowableActions"
 7844 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
 7845 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-adle-43e5-b176-
 7846 247b64e4f4bc/allowableactions"/>

```

7847      <atom:link type="application/atom+xml;type=entry" rel="describedby"
7848 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7849 247b64e4f4bc/type"/>
7850      <atom:link type="application/atomsvc+xml" rel="service"
7851 href="http://cmisexample.oasis-open.org/rep1//service"/>
7852      <atom:published>2009-10-19T10:10:00.312-07:00</atom:published>
7853      <atom:summary type="html">HTML summary of Entry 7d9c26a2-ad1e-43e5-
7854 b176-247b64e4f4bc</atom:summary>
7855      <atom:link type="application/atom+xml;type=entry" rel="up"
7856 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7857 247b64e4f4bc/up"/>
7858      <atom:link type="application/atom+xml;type=feed" rel="down"
7859 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7860 247b64e4f4bc/children"/>
7861      <atom:link type="application/cmistree+xml" rel="down"
7862 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7863 247b64e4f4bc/tree"/>
7864      <atom:link type="application/atom+xml;type=feed"
7865 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7866 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7867 247b64e4f4bc/foldertree"/>
7868      <atom:link type="application/atom+xml;type=feed"
7869 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7870 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7871 247b64e4f4bc/relationships"/>
7872      <atom:link type="application/atom+xml;type=feed"
7873 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7874 href="http://cmisexample.oasis-open.org/rep1/7d9c26a2-ad1e-43e5-b176-
7875 247b64e4f4bc/policies"/>
7876      <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7877 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7878 open.org/rep1/7d9c26a2-ad1e-43e5-b176-247b64e4f4bc/acl"/>
7879      <cmisra:object>
7880          <cmis:properties>
7881              <cmis:propertyId localName="rep-cmis:objectId"
7882 propertyDefinitionId="cmis:objectId">
7883                  <cmis:value>7d9c26a2-ad1e-43e5-b176-
7884 247b64e4f4bc</cmis:value>
7885                  </cmis:propertyId>
7886                  <cmis:propertyId localName="rep-cmis:objectTypeId"
7887 propertyDefinitionId="cmis:objectTypeId">
7888                      <cmis:value>customer</cmis:value>
7889                  </cmis:propertyId>
7890                  <cmis:propertyString localName="rep-cmis:name"
7891 propertyDefinitionId="cmis:name">
7892                      <cmis:value>Customer Folder</cmis:value>
7893                  </cmis:propertyString>
7894                  <cmis:propertyDateTime localName="rep-cmis:creationDate"
7895 propertyDefinitionId="cmis:creationDate">
7896                      <cmis:value>2009-10-19T10:10:00.328-07:00</cmis:value>
7897                  </cmis:propertyDateTime>
7898                  <cmis:propertyDateTime localName="rep-
7899 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7900                      <cmis:value>2009-10-19T10:10:00.328-07:00</cmis:value>
7901                  </cmis:propertyDateTime>
7902                  <cmis:propertyId localName="rep-cmis:baseTypeId"
7903 propertyDefinitionId="cmis:baseTypeId">
7904                      <cmis:value>cmis:folder</cmis:value>
7905                  </cmis:propertyId>
7906                  <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7907 propertyDefinitionId="cmis:lastModifiedBy">
7908                      <cmis:value>Al Brown</cmis:value>
7909                  </cmis:propertyString>

```

```

7910          <cmis:propertyString localName="rep-cmis:createdBy"
7911          propertyDefinitionId="cmis:createdBy">
7912              <cmis:value>Al Brown</cmis:value>
7913          </cmis:propertyString>
7914          <cmis:propertyId localName="rep-cmis:parentId"
7915          propertyDefinitionId="cmis:parentId">
7916              <cmis:value>7d9c26a2-ad1e-43e5-b176-
7917              247b64e4f4bcup</cmis:value>
7918          </cmis:propertyId>
7919      </cmis:properties>
7920  </cmisra:object>
7921      <cmisra:pathSegment>customer</cmisra:pathSegment>
7922  </atom:entry>
7923</atom:feed>

```

7924

7925 Please also see the example documents included with the schema.

7926 **3.9.4.1 GET**

7927 The following arguments may be supplied. Please see the domain model for more information:

- 7928 • filter
- 7929 • depth
- 7930 • includeAllowableActions
- 7931 • includeRelationships
- 7932 • renditionFilter

7933 **3.9.4.2 DELETE**

7934 This is the same as DELETE on Folder Descendants. Please see that section.

7935 **3.9.5 AllVersions Feed**

7936 This is a feed comprised of all the versions of the given document.

7937 CMIS Services:

```

7938      GET: getAllVersions
7939      DELETE: deleteAllVersions
7940 Media Type: application/atom+xml;type=feed
7941

```

7942 The feed SHOULD contain the newest versions at the beginning of the feed.

7943

7944 Link Relations:

- 7945 • service: Points to service document containing the CMIS repository. The service document
7946 MUST contain only one workspace element.
 - 7947 ○ Media Type: application/atomsrv+xml
- 7948 • via: points to the atom entry of the resource generating this collection
- 7949 • paging link relations as appropriate: first, next, previous, last

7950

7951 This feed contains a set of atom entries for each version in the version series

- 7952 • cmisra:object inside atom:entry
- 7953 • cmisra:children inside atom:entry if atom:entry represents a CMIS Folder

7954

7955 Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
  xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
  xmlns:atom="http://www.w3.org/2005/Atom"
  xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
  open.org/ns/cmis/restatom/200908/">
  <atom:title type="text">AllVersions for Document c3141e18-fb48-4742-a6f8-
  adef86dea79a</atom:title>
  <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:updated>2009-10-19T10:09:59.828-07:00</atom:updated>
  <atom:id>urn:uuid:309c322f-1e10-4306-85ba-98f1870c5882</atom:id>
  <atom:link type="application/atom+xml;type=feed" rel="self"
  href="http://cmisexample.oasis-open.org/repl/c3141e18-fb48-4742-a6f8-
  adef86dea79a/3"/>
  <atom:link type="application/atomsvc+xml" rel="service"
  href="http://cmisexample.oasis-open.org/repl//service"/>
    <atom:link type="application/atom+xml;type=entry" rel="via"
  href="http://cmisexample.oasis-open.org/repl/c3141e18-fb48-4742-a6f8-
  adef86dea79a"/>
    <cmisra:numItems>1</cmisra:numItems>
    <atom:entry>
      <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/repl/9c7860a7-
  be25-439e-b473-8da032402a6d"/>
        <atom:id>urn:uuid:9c7860a7-be25-439e-b473-8da032402a6d</atom:id>
        <atom:title type="text">Invoice (Version1)</atom:title>
        <atom:updated>2009-10-19T10:09:59.828-07:00</atom:updated>
        <atom:link rel="self" href="http://cmisexample.oasis-
  open.org/repl/9c7860a7-be25-439e-b473-8da032402a6d"/>
        <atom:link rel="edit" href="http://cmisexample.oasis-
  open.org/repl/9c7860a7-be25-439e-b473-8da032402a6d"/>
        <atom:link type="application/cmis+xml;type=allowableActions"
  rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
  href="http://cmisexample.oasis-open.org/repl/9c7860a7-be25-439e-b473-
  8da032402a6d/allowableactions"/>
        <atom:link type="application/atom+xml;type=entry" rel="describedby"
  href="http://cmisexample.oasis-open.org/repl/9c7860a7-be25-439e-b473-
  8da032402a6d/type"/>
        <atom:link type="application/atomsvc+xml" rel="service"
  href="http://cmisexample.oasis-open.org/repl//service"/>
        <atom:published>2009-10-19T10:09:59.843-07:00</atom:published>
        <atom:summary type="html">HTML summary of Entry 9c7860a7-be25-439e-
  b473-8da032402a6d</atom:summary>
        <atom:link rel="edit-media" href="http://cmisexample.oasis-
  open.org/repl/9c7860a7-be25-439e-b473-8da032402a6d/edit-media"/>
        <atom:link rel="alternate" href="http://cmisexample.oasis-
  open.org/repl/9c7860a7-be25-439e-b473-8da032402a6d/alternate"/>
        <atom:link type="application/atom+xml;type=feed" rel="up"
  href="http://cmisexample.oasis-open.org/repl/9c7860a7-be25-439e-b473-
  8da032402a6d/parents"/>
        <atom:link type="application/atom+xml;type=feed" rel="version-history"
  href="http://cmisexample.oasis-open.org/repl/9c7860a7-be25-439e-b473-
  8da032402a6d/allversions"/>
```

```

8017      <atom:link type="application/atom+xml;type=entry" rel="current-
8018      version" href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8019      8da032402a6d/latest"/>
8020      <atom:link type="application/atom+xml;type=feed"
8021      rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
8022      href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8023      8da032402a6d/relationships"/>
8024      <atom:link type="application/atom+xml;type=feed"
8025      rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
8026      href="http://cmisexample.oasis-open.org/rep1/9c7860a7-be25-439e-b473-
8027      8da032402a6d/policies"/>
8028      <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8029      open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
8030      open.org/rep1/9c7860a7-be25-439e-b473-8da032402a6d/acl"/>
8031      <cmisra:object>
8032          <cmis:properties>
8033              <cmis:propertyId localName="rep-cmis:objectId"
8034              propertyDefinitionId="cmis:objectId">
8035                  <cmis:value>9c7860a7-be25-439e-b473-
8036                  8da032402a6d</cmis:value>
8037                  </cmis:propertyId>
8038          </cmis:properties>
8039      </cmisra:object>
8040  </atom:entry>
8041 </atom:feed>

```

8042
8043 Please also see the example documents included with the schema.

8044 **3.9.5.1 GET**

8045 The following arguments may be supplied. Please see the domain model for more information:
8046 • filter
8047 • includeAllowableActions

8048 **3.9.5.2 DELETE**

8049 This removes the entire version history of the document.
8050
8051 Success HTTP code: 204

8052 **3.9.6 Type Descendants Feed**

8053 This is a feed described in the service document that contains all the types under a specific type in the
8054 repository to a specific depth. If no parent type is specified, then the base types and their descendants
8055 are returned in the feed which is equivalent to all types in the repository if depth is infinite. The link
8056 relation is <http://docs.oasis-open.org/ns/cmis/link/200908/typesdescendants>.
8057

8058 Types are nested using the CMIS hierarchy extension. Please see section 3.2.3.2 Hierarchy Navigation
8059 Internet Draft Link Relations.

8060
8061 CMIS Services:
8062 GET: getTypeDescendants

8063
8064 Media Type: application/atom+xml;type=feed
8065

- 8066 Link Relations:
- 8067 • service: Points to service document containing the CMIS repository. The service document
8068 MUST contain only one workspace element.
 - 8069 ○ Media Type: application/atomsrv+xml
 - 8070 • via: points to the type definition whose descendants represent this feed. This link is not present if
8071 no parent type is specified.
 - 8072 • down: points to the children feed for the same type
 - 8073 • up: points to the parent type definition
 - 8074 ○ If this is a descendants feed for a base object type, this link is not present.
- 8075
- 8076 The following CMIS Atom extension element MAY be included inside the atom feed:
- 8077 • cmisra:numItems
- 8078
- 8079 Example:
- ```

8080 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8081 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
8082 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
8083 xmlns:atom="http://www.w3.org/2005/Atom"
8084 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8085 open.org/ns/cmis/restatom/200908/">
8086 <atom:title type="text">Base Types</atom:title>
8087 <atom:author>
8088 <atom:name>Al Brown</atom:name>
8089 <atom:uri>http://www.ibm.com/</atom:uri>
8090 <atom:email>albertcbrown@us.ibm.com</atom:email>
8091 </atom:author>
8092 <atom:updated>2009-10-19T10:10:00.656-07:00</atom:updated>
8093 <atom:id>urn:uuid:e1981abc-a130-4cf0-89f2-0fa26f2801fe</atom:id>
8094 <atom:link type="application/atom+xml;type=feed" rel="self"
8095 href="http://cmisexample.oasis-open.org/rep1//3"/>
8096 <atom:link type="application/atomsrv+xml" rel="service"
8097 href="http://cmisexample.oasis-open.org/rep1//service"/>
8098 <atom:link type="application/atom+xml;type=entry" rel="via"
8099 href="http://cmisexample.oasis-open.org/rep1//"/>
8100 <atom:link type="application/atom+xml;type=feed" rel="down"
8101 href="http://cmisexample.oasis-open.org/rep1//children"/>
8102 <cmisra:numItems>1</cmisra:numItems>
8103 <atom:entry>
8104 <atom:author>
8105 <atom:name>Al Brown</atom:name>
8106 <atom:uri>http://www.ibm.com/</atom:uri>
8107 <atom:email>albertcbrown@us.ibm.com</atom:email>
8108 </atom:author>
8109 <atom:content>Type Definition for cmis:document</atom:content>
8110 <atom:id>http://cmisexample.oasis-
8111 open.org/rep1/type/cmis:document</atom:id>
8112 <atom:link type="application/atom+xml;type=entry" rel="self"
8113 href="http://cmisexample.oasis-open.org/rep1/type/cmis:document"/>
8114 <atom:link type="application/atomsrv+xml" rel="service"
8115 href="http://cmisexample.oasis-open.org/rep1/type/cmis:document"/>
8116 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8117 href="http://cmisexample.oasis-open.org/rep1/type/cmis:document"/>
8118 <atom:link type="application/atom+xml;type=entry" rel="up"
8119 href="http://cmisexample.oasis-open.org/rep1/type/cmis:document/parent"/>
8120 <atom:link type="application/atom+xml;type=feed" rel="down"
8121 href="http://cmisexample.oasis-
8122 open.org/rep1/type/cmis:document/children/flat"/>
```

```

8123 <atom:link type="application/cmistree+xml" rel="down"
8124 href="http://cmisexample.oasis-
8125 open.org/repl/type/cmis:document/children/tree"/>
8126 <atom:published>2009-10-19T10:10:00.656-07:00</atom:published>
8127 <atom:summary type="html">HTML summary of Type Definition
8128 cmis:document</atom:summary>
8129 <atom:title type="text">Type Definition - cmis:document</atom:title>
8130 <atom:updated>2009-10-19T10:10:00.656-07:00</atom:updated>
8131 <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8132 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8133 xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="cmis:document">
8134 <cmis:id>d1cmis:document</cmis:id>
8135 <cmis:localName>myrepname-cmis:document</cmis:localName>
8136 <cmis:localNamespace xsi:nil="true"/>
8137 <cmis:displayName>cmis:document</cmis:displayName>
8138 <cmis:queryName>cmis:document</cmis:queryName>
8139 <cmis:description>Description for type definition
8140 cmis:document</cmis:description>
8141 <cmis:baseId>cmis:document</cmis:baseId>
8142 <cmis:parentId>parent</cmis:parentId>
8143 <cmis:creatable>true</cmis:creatable>
8144 <cmis:fileable>true</cmis:fileable>
8145 <cmis:queryable>false</cmis:queryable>
8146 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8147
8148 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8149 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8150 <cmis:controllableACL>true</cmis:controllableACL>
8151 <cmis:versionable>true</cmis:versionable>
8152 <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8153 </cmisra:type>
8154 <cmisra:children>
8155 <atom:title type="text">Children for Document</atom:title>
8156 <atom:author>
8157 <atom:name>Al Brown</atom:name>
8158 <atom:uri>http://www.ibm.com/</atom:uri>
8159 <atom:email>albertcbrown@us.ibm.com</atom:email>
8160 </atom:author>
8161 <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8162 <atom:id>urn:uuid:5657c90b-b07f-4225-832d-28a5302f8424</atom:id>
8163 <atom:link type="application/atom+xml;type=feed" rel="self"
8164 href="http://cmisexample.oasis-open.org/repl/cmis:document/3"/>
8165 <atom:link type="application/atomsvc+xml" rel="service"
8166 href="http://cmisexample.oasis-open.org/repl/service"/>
8167 <atom:link type="application/atom+xml;type=entry" rel="via"
8168 href="http://cmisexample.oasis-open.org/repl/cmis:document"/>
8169 <atom:link type="application/atom+xml;type=feed" rel="down"
8170 href="http://cmisexample.oasis-open.org/repl/cmis:document/children"/>
8171 <atom:link type="application/atom+xml;type=entry" rel="up"
8172 href="http://cmisexample.oasis-open.org/repl/document"/>
8173 <cmisra:numItems>1</cmisra:numItems>
8174 <atom:entry>
8175 <atom:author>
8176 <atom:name>Al Brown</atom:name>
8177 <atom:uri>http://www.ibm.com/</atom:uri>
8178 <atom:email>albertcbrown@us.ibm.com</atom:email>
8179 </atom:author>
8180 <atom:content>Type Definition for invoice-
8181 document</atom:content>
8182 <atom:id>http://cmisexample.oasis-open.org/repl/type/invoice-
8183 document</atom:id>
8184 <atom:link type="application/atom+xml;type=entry" rel="self"
8185 href="http://cmisexample.oasis-open.org/repl/type/invoice-document"/>
```

```

8186 <atom:link type="application/atomsvc+xml" rel="service"
8187 href="http://cmisexample.oasis-open.org/repl/type/invoice-document"/>
8188 <atom:link type="application/atom+xml;type=entry"
8189 rel="describedby" href="http://cmisexample.oasis-
8190 open.org/repl/type/cmis:document"/>
8191 <atom:link type="application/atom+xml;type=entry" rel="up"
8192 href="http://cmisexample.oasis-open.org/repl/type/invoice-document/parent"/>
8193 <atom:link type="application/atom+xml;type=feed" rel="down"
8194 href="http://cmisexample.oasis-open.org/repl/type/invoice-
8195 document/children/flat"/>
8196 <atom:link type="application/cmistree+xml" rel="down"
8197 href="http://cmisexample.oasis-open.org/repl/type/invoice-
8198 document/children/tree"/>
8199 <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8200 <atom:summary type="html">HTML summary of Type Definition
8201 invoice-document</atom:summary>
8202 <atom:title type="text">Type Definition - invoice-
8203 document</atom:title>
8204 <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8205 <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8206 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8207 instance" xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="invoice-
8208 document">
8209 <cmis:id>dtinvoice-document</cmis:id>
8210 <cmis:localName>myrepname-invoice-
8211 document</cmis:localName>
8212 <cmis:localNamespace xsi:nil="true"/>
8213 <cmis:displayName>invoice-document</cmis:displayName>
8214 <cmis:queryName>invoice-document</cmis:queryName>
8215 <cmis:description>Description for type definition invoice-
8216 document</cmis:description>
8217 <cmis:baseId>cmis:document</cmis:baseId>
8218 <cmis:parentId>parent</cmis:parentId>
8219 <cmis:creatable>true</cmis:creatable>
8220 <cmis:fileable>true</cmis:fileable>
8221 <cmis:queryable>false</cmis:queryable>
8222 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8223
8224 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8225 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8226 <cmis:controllableACL>true</cmis:controllableACL>
8227 <cmis:versionable>true</cmis:versionable>
8228
8229 <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8230 </cmisra:type>
8231 </atom:entry>
8232 </cmisra:children>
8233 </atom:entry>
8234 <atom:entry>
8235 <atom:author>
8236 <atom:name>Al Brown</atom:name>
8237 <atom:uri>http://www.ibm.com/</atom:uri>
8238 <atom:email>albertcbrown@us.ibm.com</atom:email>
8239 </atom:author>
8240 <atom:content>Type Definition for cmis:folder</atom:content>
8241 <atom:id>http://cmisexample.oasis-
8242 open.org/repl/type/cmis:folder</atom:id>
8243 <atom:link type="application/atom+xml;type=entry" rel="self"
8244 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8245 <atom:link type="application/atomsvc+xml" rel="service"
8246 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8247 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8248 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
```

```

8249 <atom:link type="application/atom+xml;type=entry" rel="up"
8250 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/parent"/>
8251 <atom:link type="application/atom+xml;type=feed" rel="down"
8252 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/children/flat"/>
8253 <atom:link type="application/cmistree+xml" rel="down"
8254 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/children/tree"/>
8255 <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8256 <atom:summary type="html">HTML summary of Type Definition
8257 cmis:folder</atom:summary>
8258 <atom:title type="text">Type Definition - cmis:folder</atom:title>
8259 <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8260 <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8261 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8262 xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="cmis:folder">
8263 <cmis:id>dtcmis:folder</cmis:id>
8264 <cmis:localName>myrepname-cmis:folder</cmis:localName>
8265 <cmis:localNamespace xsi:nil="true"/>
8266 <cmis:displayName>cmis:folder</cmis:displayName>
8267 <cmis:queryName>cmis:folder</cmis:queryName>
8268 <cmis:description>Description for type definition
8269 cmis:folder</cmis:description>
8270 <cmis:baseId>cmis:folder</cmis:baseId>
8271 <cmis:parentId>parent</cmis:parentId>
8272 <cmis:creatable>true</cmis:creatable>
8273 <cmis:fileable>true</cmis:fileable>
8274 <cmis:queryable>false</cmis:queryable>
8275 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8276
8277 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8278 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8279 <cmis:controllableACL>true</cmis:controllableACL>
8280 </cmisra:type>
8281 <cmisra:children>
8282 <atom:title type="text">Children for Folder</atom:title>
8283 <atom:author>
8284 <atom:name>Al Brown</atom:name>
8285 <atom:uri>http://www.ibm.com/</atom:uri>
8286 <atom:email>albertcbrown@us.ibm.com</atom:email>
8287 </atom:author>
8288 <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8289 <atom:id>urn:uuid:7b556496-2a09-40ca-b2a5-3e4e5596fb89</atom:id>
8290 <atom:link type="application/atom+xml;type=feed" rel="self"
8291 href="http://cmisexample.oasis-open.org/repl/cmis:folder/3"/>
8292 <atom:link type="application/atomsvc+xml" rel="service"
8293 href="http://cmisexample.oasis-open.org/repl//service"/>
8294 <atom:link type="application/atom+xml;type=entry" rel="via"
8295 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8296 <atom:link type="application/atom+xml;type=feed" rel="down"
8297 href="http://cmisexample.oasis-open.org/repl/cmis:folder/children"/>
8298 <atom:link type="application/atom+xml;type=entry" rel="up"
8299 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8300 <cmisra:numItems>1</cmisra:numItems>
8301 <atom:entry>
8302 <atom:author>
8303 <atom:name>Al Brown</atom:name>
8304 <atom:uri>http://www.ibm.com/</atom:uri>
8305 <atom:email>albertcbrown@us.ibm.com</atom:email>
8306 </atom:author>
8307 <atom:content>Type Definition for customer-
8308 folder</atom:content>
8309 <atom:id>http://cmisexample.oasis-open.org/repl/type/customer-
8310 folder</atom:id>
8311 <atom:link type="application/atom+xml;type=entry" rel="self"
8312 href="http://cmisexample.oasis-open.org/repl/type/customer-folder"/>

```

```

8313 <atom:link type="application/atomsvc+xml" rel="service"
8314 href="http://cmisexample.oasis-open.org/repl/type/customer-folder"/>
8315 <atom:link type="application/atom+xml;type=entry"
8316 rel="describedby" href="http://cmisexample.oasis-
8317 open.org/repl/type/cmis:folder"/>
8318 <atom:link type="application/atom+xml;type=entry" rel="up"
8319 href="http://cmisexample.oasis-open.org/repl/type/customer-folder/parent"/>
8320 <atom:link type="application/atom+xml;type=feed" rel="down"
8321 href="http://cmisexample.oasis-open.org/repl/type/customer-
8322 folder/children/flat"/>
8323 <atom:link type="application/cmistree+xml" rel="down"
8324 href="http://cmisexample.oasis-open.org/repl/type/customer-
8325 folder/children/tree"/>
8326 <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8327 <atom:summary type="html">HTML summary of Type Definition
8328 customer-folder</atom:summary>
8329 <atom:title type="text">Type Definition - customer-
8330 folder</atom:title>
8331 <atom:updated>2009-10-19T10:10:00.671-07:00</atom:updated>
8332 <app:edited>2009-10-19T10:10:00.671-07:00</app:edited>
8333 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8334 instance" xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="customer-
8335 folder">
8336 <cmis:id>dtcustomer-folder</cmis:id>
8337 <cmis:localName>myrepname-customer-folder</cmis:localName>
8338 <cmis:localNamespace xsi:nil="true"/>
8339 <cmis:displayName>customer-folder</cmis:displayName>
8340 <cmis:queryName>customer-folder</cmis:queryName>
8341 <cmis:description>Description for type definition
8342 customer-folder</cmis:description>
8343 <cmis:baseId>cmis:folder</cmis:baseId>
8344 <cmis:parentId>parent</cmis:parentId>
8345 <cmis:creatable>true</cmis:creatable>
8346 <cmis:fileable>true</cmis:fileable>
8347 <cmis:queryable>false</cmis:queryable>
8348 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8349
8350 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8351 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8352 <cmis:controllableACL>true</cmis:controllableACL>
8353 </cmisra:type>
8354 </atom:entry>
8355 </cmisra:children>
8356 </atom:entry>
8357 <atom:entry>
8358 <atom:author>
8359 <atom:name>Al Brown</atom:name>
8360 <atom:uri>http://www.ibm.com/</atom:uri>
8361 <atom:email>albertcbrown@us.ibm.com</atom:email>
8362 </atom:author>
8363 <atom:content>Type Definition for cmis:relationship</atom:content>
8364 <atom:id>http://cmisexample.oasis-
8365 open.org/repl/type/cmis:relationship</atom:id>
8366 <atom:link type="application/atom+xml;type=entry" rel="self"
8367 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8368 <atom:link type="application/atomsvc+xml" rel="service"
8369 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8370 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8371 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8372 <atom:link type="application/atom+xml;type=entry" rel="up"
8373 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship/parent"/>
8374 <atom:link type="application/atom+xml;type=feed" rel="down"
8375 href="http://cmisexample.oasis-
8376 open.org/repl/type/cmis:relationship/children/flat"/>
```

```

8377 <atom:link type="application/cmistree+xml" rel="down"
8378 href="http://cmisexample.oasis-
8379 open.org/repl/type/cmis:relationship/children/tree"/>
8380 <atom:published>2009-10-19T10:10:00.671-07:00</atom:published>
8381 <atom:summary type="html">HTML summary of Type Definition
8382 cmis:relationship</atom:summary>
8383 <atom:title type="text">Type Definition -
8384 cmis:relationship</atom:title>
8385 <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8386 <app:edited>2009-10-19T10:10:00.687-07:00</app:edited>
8387 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8388 xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8389 cmisra:id="cmis:relationship">
8390 <cmis:id>d1cmis:relationship</cmis:id>
8391 <cmis:localName>myrepname-cmis:relationship</cmis:localName>
8392 <cmis:localNamespace xsi:nil="true"/>
8393 <cmis:displayName>cmis:relationship</cmis:displayName>
8394 <cmis:queryName>cmis:relationship</cmis:queryName>
8395 <cmis:description>Description for type definition
8396 cmis:relationship</cmis:description>
8397 <cmis:baseId>cmis:relationship</cmis:baseId>
8398 <cmis:parentId>parent</cmis:parentId>
8399 <cmis:creatable>true</cmis:creatable>
8400 <cmis:fileable>false</cmis:fileable>
8401 <cmis:queryable>false</cmis:queryable>
8402 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8403
8404 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8405 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8406 <cmis:controllableACL>true</cmis:controllableACL>
8407 <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8408 <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8409 <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8410 </cmisra:type>
8411 <cmisra:children>
8412 <atom:title type="text">Children for Relationship</atom:title>
8413 <atom:author>
8414 <atom:name>Al Brown</atom:name>
8415 <atom:uri>http://www.ibm.com/</atom:uri>
8416 <atom:email>albertcbrown@us.ibm.com</atom:email>
8417 </atom:author>
8418 <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8419 <atom:id>urn:uuid:1bc33a0d-27ac-4976-9cab-58ea28e3ff6c</atom:id>
8420 <atom:link type="application/atom+xml;type=feed" rel="self"
8421 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/3"/>
8422 <atom:link type="application/atomsvc+xml" rel="service"
8423 href="http://cmisexample.oasis-open.org/repl//service"/>
8424 <atom:link type="application/atom+xml;type=entry" rel="via"
8425 href="http://cmisexample.oasis-open.org/repl/cmis:relationship"/>
8426 <atom:link type="application/atom+xml;type=feed" rel="down"
8427 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/children"/>
8428 <atom:link type="application/atom+xml;type=entry" rel="up"
8429 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8430 <cmisra:numItems>1</cmisra:numItems>
8431 <atom:entry>
8432 <atom:author>
8433 <atom:name>Al Brown</atom:name>
8434 <atom:uri>http://www.ibm.com/</atom:uri>
8435 <atom:email>albertcbrown@us.ibm.com</atom:email>
8436 </atom:author>
8437 <atom:content>Type Definition for customer-
8438 relationship</atom:content>
8439 <atom:id>http://cmisexample.oasis-open.org/repl/type/customer-
8440 relationship</atom:id>
```

```

8441 <atom:link type="application/atom+xml;type=entry" rel="self"
8442 href="http://cmisexample.oasis-open.org/repl/type/customer-relationship"/>
8443 <atom:link type="application/atomsvc+xml" rel="service"
8444 href="http://cmisexample.oasis-open.org/repl/type/customer-relationship"/>
8445 <atom:link type="application/atom+xml;type=entry"
8446 rel="describedby" href="http://cmisexample.oasis-
8447 open.org/repl/type/cmis:relationship"/>
8448 <atom:link type="application/atom+xml;type=entry" rel="up"
8449 href="http://cmisexample.oasis-open.org/repl/type/customer-
8450 relationship/parent"/>
8451 <atom:link type="application/atom+xml;type=feed" rel="down"
8452 href="http://cmisexample.oasis-open.org/repl/type/customer-
8453 relationship/children/flat"/>
8454 <atom:link type="application/cmistree+xml" rel="down"
8455 href="http://cmisexample.oasis-open.org/repl/type/customer-
8456 relationship/children/tree"/>
8457 <atom:published>2009-10-19T10:10:00.687-07:00</atom:published>
8458 <atom:summary type="html">HTML summary of Type Definition
8459 customer-relationship</atom:summary>
8460 <atom:title type="text">Type Definition - customer-
8461 relationship</atom:title>
8462 <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8463 <app:edited>2009-10-19T10:10:00.687-07:00</app:edited>
8464 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8465 instance" xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8466 cmisra:id="customer-relationship">
8467 <cmis:id>dtcustomer-relationship</cmis:id>
8468 <cmis:localName>myrepname-customer-
8469 relationship</cmis:localName>
8470 <cmis:localNamespace xsi:nil="true"/>
8471 <cmis:displayName>customer-relationship</cmis:displayName>
8472 <cmis:queryName>customer-relationship</cmis:queryName>
8473 <cmis:description>Description for type definition
8474 customer-relationship</cmis:description>
8475 <cmis:baseId>cmis:relationship</cmis:baseId>
8476 <cmis:parentId>parent</cmis:parentId>
8477 <cmis:creatable>true</cmis:creatable>
8478 <cmis:fileable>false</cmis:fileable>
8479 <cmis:queryable>false</cmis:queryable>
8480 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8481
8482 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8483 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8484 <cmis:controllableACL>true</cmis:controllableACL>
8485 <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8486
8487 <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8488
8489 <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8490 </cmisra:type>
8491 </atom:entry>
8492 </cmisra:children>
8493 </atom:entry>
8494 <atom:entry>
8495 <atom:author>
8496 <atom:name>Al Brown</atom:name>
8497 <atom:uri>http://www.ibm.com/</atom:uri>
8498 <atom:email>albertcbrown@us.ibm.com</atom:email>
8499 </atom:author>
8500 <atom:content>Type Definition for cmis:policy</atom:content>
8501 <atom:id>http://cmisexample.oasis-
8502 open.org/repl/type/cmis:policy</atom:id>
8503 <atom:link type="application/atom+xml;type=entry" rel="self"
8504 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
```

```

8505 <atom:link type="application/atomsvc+xml" rel="service"
8506 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8507 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8508 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8509 <atom:link type="application/atom+xml;type=entry" rel="up"
8510 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/parent"/>
8511 <atom:link type="application/atom+xml;type=feed" rel="down"
8512 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/flat"/>
8513 <atom:link type="application/cmistree+xml" rel="down"
8514 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/tree"/>
8515 <atom:published>2009-10-19T10:10:00.687-07:00</atom:published>
8516 <atom:summary type="html">HTML summary of Type Definition
8517 cmis:policy</atom:summary>
8518 <atom:title type="text">Type Definition - cmis:policy</atom:title>
8519 <atom:updated>2009-10-19T10:10:00.687-07:00</atom:updated>
8520 <app:edited>2009-10-19T10:10:00.687-07:00</app:edited>
8521 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8522 xsi:type="cmis:cmisTypePolicyDefinitionType" cmisra:id="cmis:policy">
8523 <cmis:id>d1cmis:policy</cmis:id>
8524 <cmis:localName>myrepname-cmis:policy</cmis:localName>
8525 <cmis:localNamespace xsi:nil="true"/>
8526 <cmis:displayName>cmis:policy</cmis:displayName>
8527 <cmis:queryName>cmis:policy</cmis:queryName>
8528 <cmis:description>Description for type definition
8529 cmis:policy</cmis:description>
8530 <cmis:baseId>cmis:policy</cmis:baseId>
8531 <cmis:parentId>parent</cmis:parentId>
8532 <cmis:creatable>true</cmis:creatable>
8533 <cmis:fileable>false</cmis:fileable>
8534 <cmis:queryable>false</cmis:queryable>
8535 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8536
8537 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8538 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8539 <cmis:controllableACL>true</cmis:controllableACL>
8540 </cmisra:type>
8541 </atom:entry>
8542 </atom:feed>

```

8543  
8544 Please also see the example documents included with the schema.

### 8545 **3.9.6.1 GET**

8546 The following arguments may be supplied. Please see the domain model for more information:  
8547     

- includePropertyDefinitions
- depth

8548

## 8549 **3.10 Resources**

### 8550 **3.10.1 Type Entry**

8551 This represents a type definition in the repository. This is enclosed as an atom entry

8552

8553 CMIS Services:

8554     GET: getTypeDefinition

8555 Media Type: application/atom+xml;type=entry

8556

8557 Link Relations:

- service: Points to service document containing the CMIS repository. The service document MUST contain only one workspace element.
  - Media Type: application/atomsrv+xml
- up: Points to the parent type as atom entry if applicable
- down: Points to the children of this type as atom feed if applicable
  - (Children) Media Type: application/atom +xml;type=feed points to the atom feed document representing the children feed for this same type
  - (Descendants) Media Type: application/cmistree+xml points to the atom feed document representing the descendants feed for this same type
- describedby: Points to the type definition atom entry of the base type of this type definition.

The following CMIS Atom extension element MUST be included inside the atom entry:

- cmisra:type

### 3.10.1.1 GET

There are no optional arguments for this resource.

Request:

```
GET /obj/0010d88d-d006-4d50-aad0-c8f61f80f273 HTTP/1.1
Host: example.org
```

Response:

```
HTTP/1.1 200 Ok
Date: Mon, 19 Oct 2009 10:10:01 -0700
Content-Length: 2995
Content-Type: application/atom+xml;type=entry
Location: http://cmisexample.oasis-open.org/repl/cmis:document

<?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/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
 open.org/ns/cmis/restatom/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>Type Definition for cmis:document</atom:content>
 <atom:id>http://cmisexample.oasis-
 open.org/repl/type/cmis:document</atom:id>
 <atom:link type="application/atom+xml;type=entry" rel="self"
 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
 <atom:link type="application/atomsrv+xml" rel="service"
 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
 <atom:link type="application/atom+xml;type=entry" rel="describedby"
 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
 <atom:link type="application/atom+xml;type=entry" rel="up"
 href="http://cmisexample.oasis-open.org/repl/type/cmis:document/parent"/>
 <atom:link type="application/atom+xml;type=feed" rel="down"
 href="http://cmisexample.oasis-
 open.org/repl/type/cmis:document/children/flat"/>
```

```

8612 <atom:link type="application/cmisTree+xml" rel="down"
8613 href="http://cmisexample.oasis-
8614 open.org/repl/type/cmis:document/children/tree"/>
8615 <atom:published>2009-10-19T10:10:01.015-07:00</atom:published>
8616 <atom:summary type="html">HTML summary of Type Definition
8617 cmis:document</atom:summary>
8618 <atom:title type="text">Type Definition - cmis:document</atom:title>
8619 <atom:updated>2009-10-19T10:10:01.015-07:00</atom:updated>
8620 <app:edited>2009-10-19T10:10:01.015-07:00</app:edited>
8621 <cmisra:type xsi:type="cmis:cmisTypeDocumentDefinitionType"
8622 cmisra:id="cmis:document" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8623 instance">
8624 <cmis:id>d1cmis:document</cmis:id>
8625 <cmis:localName>myrepname-cmis:document</cmis:localName>
8626 <cmis:localNamespace xsi:nil="true"/>
8627 <cmis:displayName>cmis:document</cmis:displayName>
8628 <cmis:queryName>cmis:document</cmis:queryName>
8629 <cmis:description>Description for type definition
8630 cmis:document</cmis:description>
8631 <cmis:baseId>cmis:document</cmis:baseId>
8632 <cmis:parentId>parent</cmis:parentId>
8633 <cmis:creatable>true</cmis:creatable>
8634 <cmis:fileable>true</cmis:fileable>
8635 <cmis:queryable>false</cmis:queryable>
8636 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8637 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8638 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8639 <cmis:controllableACL>true</cmis:controllableACL>
8640 <cmis:versionable>true</cmis:versionable>
8641 <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8642 </cmisra:type>
8643 </atom:entry>
8644

```

8645  
8646 Please also see the example documents included with the schema.  
8647

### 8648 **3.10.2 Document Entry**

8649 This is a CMIS Document instance.

8650

8651 CMIS Services:

8652        GET: getObject, getObjectOfLatestVersion (getObject)  
8653        PUT: updateProperties  
8654        DELETE: deleteObject

8655 Media Type: application/atom+xml;type=entry

8656

8657 Link Relations:

- 8658     • self: Points to an URI that returns the atom entry for this document. Please see Atom for more  
8659        information.
- 8660     • edit: Points to an URI that accepts PUT of atom entry. Please see AtomPub for more information.
- 8661     • service: Points to service document containing the CMIS repository. The service document  
8662        MUST contain only one workspace element.
  - 8663            o Media Type: application/atomsrv+xml

- up: Points to the atom feed containing the set of parents. If there is only one parent, the repository MAY point this link relation directly to the atom entry of the parent.
- version-history: Points to atom feed containing the versions of this document
  - If the document is not versionable, this link relation may not be on the resource
- current-version: Points to the latest version of the document
  - Uses query parameter ‘returnVersion’ and enumReturnVersion
  - If this version is the current-version, this link relation may not be on the resource
- edit-media:
  - Same as setContentStream. Allows updating the content stream on this document
  - Please see AtomPub for more information
- working-copy: Points to the private working copy if it exists.
- describedby: Points to the type definition as an atom entry for the type of this document entry.
- alternate: this is used to identify the renditions available for the specified object. Please see the Renditions section.
- <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for this object
- <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/act>: Points to ACL document for this object

The following CMIS Atom extension element MUST be included inside the atom entry:

- cmisra:object

### 3.10.2.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- returnVersion
  - Used to differentiate between getObject() and getObjectOfLatestVersion().
  - valid values are described by the schema element cmisra:enumReturnVersion
  - If not specified, return the version specified by the URI
- includeAllowableActions
- includeRelationships
- includePolicyIds
- includeACL
- filter
- renditionFilter
  - If not specified, renditions will not be included.

Request:

```
GET /obj/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08?filter=cmis:objectId HTTP/1.1
Host: example.org
```

8707

8708 Response:

```
8709 HTTP/1.1 200 Ok
8710 Date: Mon, 19 Oct 2009 10:10:00 -0700
8711 Content-Length: 3403
8712 Content-Type: application/atom+xml;type=entry
8713 Location: /obj/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08?filter=cmis:objectId
8714
8715
8716 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8717 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8718 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8719 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
8720 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8721 open.org/ns/cmis/restatom/200908/">
8722 <atom:author>
8723 <atom:name>Al Brown</atom:name>
8724 <atom:uri>http://www.ibm.com/</atom:uri>
8725 <atom:email>albertcbrown@us.ibm.com</atom:email>
8726 </atom:author>
8727 <atom:content src="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-
8728 49b9-acf7-94851ff7ad08"/>
8729 <atom:id>urn:uuid:8a7b8f30-3e9f-49b9-acf7-94851ff7ad08</atom:id>
8730 <atom:title type="text">Invoice</atom:title>
8731 <atom:updated>2009-10-19T10:10:00.843-07:00</atom:updated>
8732 <atom:link rel="self" href="http://cmisexample.oasis-
8733 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08"/>
8734 <atom:link rel="edit" href="http://cmisexample.oasis-
8735 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08"/>
8736 <atom:link type="application/cmis+xml;type=allowableActions"
8737 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
8738 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
8739 94851ff7ad08/allowableactions"/>
8740 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8741 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
8742 94851ff7ad08/type"/>
8743 <atom:link type="application/atomsvc+xml" rel="service"
8744 href="http://cmisexample.oasis-open.org/rep1/service"/>
8745 <atom:published>2009-10-19T10:10:00.843-07:00</atom:published>
8746 <atom:summary type="html">HTML summary of Entry 8a7b8f30-3e9f-49b9-acf7-
8747 94851ff7ad08</atom:summary>
8748 <atom:link rel="edit-media" href="http://cmisexample.oasis-
8749 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/edit-media"/>
8750 <atom:link rel="alternate" href="http://cmisexample.oasis-
8751 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/alternate"/>
8752 <atom:link type="application/atom+xml;type=feed" rel="up"
8753 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
8754 94851ff7ad08/parents"/>
8755 <atom:link type="application/atom+xml;type=feed" rel="version-history"
8756 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
8757 94851ff7ad08/allversions"/>
8758 <atom:link type="application/atom+xml;type=entry" rel="current-version"
8759 href="http://cmisexample.oasis-open.org/rep1/8a7b8f30-3e9f-49b9-acf7-
8760 94851ff7ad08/latest"/>
8761 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
8762 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
8763 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/relationships"/>
8764 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
8765 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
8766 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/policies"/>
8767 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8768 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
8769 open.org/rep1/8a7b8f30-3e9f-49b9-acf7-94851ff7ad08/acl"/>
```

```
8770 <cmisra:object>
8771 <cmis:properties>
8772 <cmis:propertyId localName="rep-cmis:objectId"
8773 propertyDefinitionId="cmis:objectId">
8774 <cmis:value>8a7b8f30-3e9f-49b9-acf7-94851ff7ad08</cmis:value>
8775 </cmis:propertyId>
8776 </cmis:properties>
8777 </cmisra:object>
8778 </atom:entry>
8779
```

8780  
8781 Please also see the example documents included with the schema.

8782

### 8783 **3.10.2.2 PUT**

8784 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
8785 properties are not included, the repository SHOULD NOT modify them.

8786

8787 The server SHOULD respond with:

- 8788 • HTTP Status Code 200
- 8789 • Response Body containing the updated atom entry

8790

### 8791 **3.10.2.3 DELETE**

8792 This removes the document.

8793 Success HTTP code: 204

## 8794 **3.10.3 Document Private Working Copy (PWC) Entry**

8795 This is the private working copy of the document (checkedout version of document)

8796 CMIS Services:

- 8797 GET: getObject
- 8798 PUT: updateProperties or checkIn
- 8799 DELETE: cancelCheckOut

8800 Media Type: application/atom+xml;type=entry

8801

8802 Link relations:

- 8803 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 8804 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more  
8805 information.
- 8806 • service: Points to service document containing the CMIS repository. The service document  
8807 MUST contain only one workspace element.
  - 8808 ○ Media Type: application/atomsrv+xml
- 8809 • up: Points to the atom feed containing the set of parents. If there is only one parent, the  
8810 repository MAY point this link relation directly to the atom entry of the parent.
- 8811 • version-history
  - 8812 ○ Points to an URI that returns the feed associated with the version history
- 8813 • edit-media

- 8814           ○ Same as setContentStream. Allows updating the content stream on this document  
 8815           ○ Please see AtomPub for more information
- 8816     • via: atom entry that created this private working copy
  - 8817     • describedby: Points to the type definition as an atom entry for the type of this PWC entry.
  - 8818     • alternate: this is used to identify the renditions available for the specified object. Please see the Renditions section.
  - 8819     • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions document for this object.
  - 8820     • <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for this object
  - 8821     • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
  - 8822     • <http://docs.oasis-open.org/ns/cmis/link/200908/act>: Points to ACL document for this object

8823     The following element MUST be included inside the atom entry:

- 8824       • cmisra:object

### 8830     **3.10.3.1 GET**

8831     The following arguments may be supplied. Please see the domain model for more information:

- 8832       • filter
- 8833       • includeAllowableActions
- 8834       • includeRelationships
- 8835       • renditionFilter
  - 8836           ○ If not specified, renditions will not be included.

8837     Request:

```
8839 GET /obj/3ca7d7d0-1c98-4c38-9141-604d287fb881?filter=cmis:objectId HTTP/1.1
8840 Host: example.org
```

8842     Response:

```
8844 HTTP/1.1 200 Ok
8845 Date: Mon, 19 Oct 2009 10:10:00 -0700
8846 Content-Length: 3564
8847 Content-Type: application/atom+xml;type=entry
8848 Location: /obj/3ca7d7d0-1c98-4c38-9141-604d287fb881?filter=cmis:objectId

8849
8850
8851 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8852 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8853 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8854 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
8855 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8856 open.org/ns/cmis/restatom/200908/">
8857 <atom:author>
8858 <atom:name>Al Brown</atom:name>
8859 <atom:uri>http://www.ibm.com/</atom:uri>
8860 <atom:email>albertcbrown@us.ibm.com</atom:email>
8861 </atom:author>
```

```

8862 <atom:content src="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-
8863 4c38-9141-604d287fb881"/>
8864 <atom:id>urn:uuid:3ca7d7d0-1c98-4c38-9141-604d287fb881</atom:id>
8865 <atom:title type="text">Invoice</atom:title>
8866 <atom:updated>2009-10-19T10:10:00.968-07:00</atom:updated>
8867 <atom:link rel="self" href="http://cmisexample.oasis-
8868 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881"/>
8869 <atom:link rel="edit" href="http://cmisexample.oasis-
8870 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881"/>
8871 <atom:link type="application/cmis+xml;type=allowableActions"
8872 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
8873 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
8874 604d287fb881/allowableactions"/>
8875 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8876 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
8877 604d287fb881/type"/>
8878 <atom:link type="application/atomsvc+xml" rel="service"
8879 href="http://cmisexample.oasis-open.org/rep1//service"/>
8880 <atom:published>2009-10-19T10:10:00.968-07:00</atom:published>
8881 <atom:summary type="html">HTML summary of Entry 3ca7d7d0-1c98-4c38-9141-
8882 604d287fb881</atom:summary>
8883 <atom:link rel="edit-media" href="http://cmisexample.oasis-
8884 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/edit-media"/>
8885 <atom:link rel="alternate" href="http://cmisexample.oasis-
8886 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/alternate"/>
8887 <atom:link type="application/atom+xml;type=feed" rel="up"
8888 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
8889 604d287fb881/parents"/>
8890 <atom:link type="application/atom+xml;type=feed" rel="version-history"
8891 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
8892 604d287fb881/allversions"/>
8893 <atom:link type="application/atom+xml;type=entry" rel="current-version"
8894 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
8895 604d287fb881/latest"/>
8896 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
8897 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
8898 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/relationships"/>
8899 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
8900 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
8901 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/policies"/>
8902 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8903 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
8904 open.org/rep1/3ca7d7d0-1c98-4c38-9141-604d287fb881/acl"/>
8905 <atom:link type="application/atom+xml;type=feed" rel="working-copy"
8906 href="http://cmisexample.oasis-open.org/rep1/3ca7d7d0-1c98-4c38-9141-
8907 604d287fb881/pwc"/>
8908 <cmisra:object>
8909 <cmis:properties>
8910 <cmis:propertyId localName="rep-cmis:objectId"
8911 propertyDefinitionId="cmis:objectId">
8912 <cmis:value>3ca7d7d0-1c98-4c38-9141-604d287fb881</cmis:value>
8913 </cmis:propertyId>
8914 </cmis:properties>
8915 </cmisra:object>
8916 </atom:entry>
8917

```

8918

8919 Please also see the example documents included with the schema.

8920

- 8921 **3.10.3.2 PUT**
- 8922 This does a replacement of the atom entry with the atom entry document specified. If modifiable  
8923 properties (whencheckedout or readonly) are not included, the repository SHOULD NOT modify them.
- 8924
- 8925 The following arguments may be supplied. Please see the domain model for more information:
- 8926
  - checkinComment
  - major
  - checkin
    - Used to differentiate between updateProperties() or checkin() services. If TRUE, execute  
8930 checkin service.
- 8931
- 8932 The server SHOULD respond with:
- 8933
  - HTTP Status Code 200
  - Location header of the resource (if changed via checkin)
  - Response Body containing the updated atom entry
- 8936 **3.10.3.3 DELETE**
- 8937 This removes the document entry, in this case, cancels the check out. The PWC will be removed.
- 8938
- 8939 Success HTTP code: 204
- 8940 **3.10.4 Folder Entry**
- 8941 This is a CMIS Folder instance. The properties of a folder map onto the feed tag.
- 8942 CMIS Services:
- 8943
  - GET: getObject
  - PUT: updateProperties
  - DELETE: deleteObject (this is deletion of the folder only and not any contained objects)
- 8944 Media Type: application/atom+xml;type=entry
- 8945
- 8946 Link Relations:
- 8947
  - self: Points to the URI to retrieve this atom entry. Please see Atom for more information
  - edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
  - service: Points to service document containing the CMIS repository. The service document  
8952 MUST contain only one workspace element.
    - Media Type: application/atomsrv+xml
  - describedby: Points to the type definition as an atom entry for the type of this folder entry.
  - down: Points to the children of this folder if they exist
    - application/atom+xml : Points to the atom feed document representing the children feed  
8957 for this same folder
    - application/cmistree+xml: Points to the descendants feed of the same folder
  - up: Points to the atom entry for the parent
    - If the root folder, this link will not be present

- alternate: this is used to identify the renditions available for the specified object. Please see the Renditions section.
- <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/act>: Points to ACL document for this object
- <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

The following CMIS Atom extension element MUST be included inside the atom entry:

- cmisra:object

### 3.10.4.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions
- includeRelationships
- renditionFilter
  - If not specified, renditions will not be included.

Request:

```
GET /obj/24cbe125-1ffc-4459-8089-0e6f0a500150?filter=cmis:objectId HTTP/1.1
Host: example.org
```

Response:

```
HTTP/1.1 200 Ok
Date: Mon, 19 Oct 2009 10:10:00 -0700
Content-Length: 3332
Content-Type: application/atom+xml;type=entry
Location: /obj/24cbe125-1ffc-4459-8089-0e6f0a500150?filter=cmis:objectId

<?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/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
 open.org/ns/cmis/restatom/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/rep1/24cbe125-1ffc-
 4459-8089-0e6f0a500150"/>
 <atom:id>urn:uuid:24cbe125-1ffc-4459-8089-0e6f0a500150</atom:id>
 <atom:title type="text">Customer Folder</atom:title>
 <atom:updated>2009-10-19T10:10:00.875-07:00</atom:updated>
```

```

9011 <atom:link rel="self" href="http://cmisexample.oasis-
9012 open.org/rep1/24cbe125-1ffc-4459-8089-0e6f0a500150"/>
9013 <atom:link rel="edit" href="http://cmisexample.oasis-
9014 open.org/rep1/24cbe125-1ffc-4459-8089-0e6f0a500150"/>
9015 <atom:link type="application/cmis+xml;type=allowableActions"
9016 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9017 href="http://cmisexample.oasis-open.org/rep1/24cbe125-1ffc-4459-8089-
9018 0e6f0a500150/allowableactions"/>
9019 <atom:link type="application/atom+xml;type=entry" rel="describedby"
9020 href="http://cmisexample.oasis-open.org/rep1/24cbe125-1ffc-4459-8089-
9021 0e6f0a500150/type"/>
9022 <atom:link type="application/atomsvc+xml" rel="service"
9023 href="http://cmisexample.oasis-open.org/rep1/service"/>
9024 <atom:published>2009-10-19T10:00.875-07:00</atom:published>
9025 <atom:summary type="html">HTML summary of Entry 24cbe125-1ffc-4459-8089-
9026 0e6f0a500150</atom:summary>
9027 <atom:link type="application/atom+xml;type=entry" rel="up"
9028 href="http://cmisexample.oasis-open.org/rep1/24cbe125-1ffc-4459-8089-
9029 0e6f0a500150/up"/>
9030 <atom:link type="application/atom+xml;type=feed" rel="down"
9031 href="http://cmisexample.oasis-open.org/rep1/24cbe125-1ffc-4459-8089-
9032 0e6f0a500150/children"/>
9033 <atom:link type="application/cmistree+xml" rel="down"
9034 href="http://cmisexample.oasis-open.org/rep1/24cbe125-1ffc-4459-8089-
9035 0e6f0a500150/tree"/>
9036 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9037 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
9038 open.org/rep1/24cbe125-1ffc-4459-8089-0e6f0a500150/foldertree"/>
9039 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9040 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9041 open.org/rep1/24cbe125-1ffc-4459-8089-0e6f0a500150/relationships"/>
9042 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9043 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9044 open.org/rep1/24cbe125-1ffc-4459-8089-0e6f0a500150/policies"/>
9045 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9046 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9047 open.org/rep1/24cbe125-1ffc-4459-8089-0e6f0a500150/acl"/>
9048 <cmisra:object>
9049 <cmis:properties>
9050 <cmis:propertyId localName="rep-cmis:objectId"
9051 propertyDefinitionId="cmis:objectId">
9052 <cmis:value>24cbe125-1ffc-4459-8089-0e6f0a500150</cmis:value>
9053 </cmis:propertyId>
9054 </cmis:properties>
9055 </cmisra:object>
9056 </atom:entry>
9057

```

9058  
9059 Please also see the example documents included with the schema.  
9060

### 9061 **3.10.4.2 PUT**

9062 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9063 properties are not included, the repository SHOULD NOT modify them.  
9064

9065 The server SHOULD respond with:

- 9066 • HTTP Status Code 200
- 9067 • Response Body containing the updated atom entry

9068

9069 **3.10.4.3 DELETE**

9070 This removes the object (folder) from the repository.

9071 Success HTTP code: 204

9072 **3.10.5 Relationship Entry**

9073 This is a CMIS relationship instance. These objects are exposed via ‘relationships’ link type.

9074 CMIS Services:

9075     GET: getObject

9076     PUT: updateProperties

9077     DELETE: deleteObject

9078 Media Type: application/atom+xml;type=entry

9079

9080 Link Relations:

- self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
- service: Points to service document containing the CMIS repository. The service document MUST contain only one workspace element.
  - Media Type: application/atomsrv+xml
- describedby: Points to the type definition as an atom entry for the type of this relationship entry.
- <http://docs.oasis-open.org/ns/cmis/link/200908/target>
- <http://docs.oasis-open.org/ns/cmis/link/200908/source>
- <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions document for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- <http://docs.oasis-open.org/ns/cmis/link/200908/act>: Points to ACL document for this object

9081

9082

9083

9084

9085

9086

9087

9088

9089

9090

9091

9092

9093

9094

9095 The following element MUST be included inside the atom entry:

9096     • cmisra:object

9097

9098 **3.10.5.1 GET**

9099 The following arguments may be supplied. Please see the domain model for more information:

9100     • filter

9101     • includeAllowableActions

9102

9103 Request:

9104     GET /obj/38af30f5-020d-4274-b25c-0821b6db899b?filter=cmis:objectId HTTP/1.1

9105     Host: example.org

9106

9107

9108 Response:

9109     HTTP/1.1 200 Ok

9110     Date: Mon, 19 Oct 2009 10:10:01 -0700

```

9111 Content-Length: 2861
9112 Content-Type: application/atom+xml;type=entry
9113 Location: /obj/38af30f5-020d-4274-b25c-0821b6db899b?filter=cmis:objectId
9114
9115
9116 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9117 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9118 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9119 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9120 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9121 open.org/ns/cmis/restatom/200908/">
9122 <atom:author>
9123 <atom:name>Al Brown</atom:name>
9124 <atom:uri>http://www.ibm.com/</atom:uri>
9125 <atom:email>albertcbrown@us.ibm.com</atom:email>
9126 </atom:author>
9127 <atom:content src="http://cmisexample.oasis-open.org/rep1/38af30f5-020d-
9128 4274-b25c-0821b6db899b"/>
9129 <atom:id>urn:uuid:38af30f5-020d-4274-b25c-0821b6db899b</atom:id>
9130 <atom:title type="text">Customer Relationship</atom:title>
9131 <atom:updated>2009-10-19T10:10:01.000-07:00</atom:updated>
9132 <atom:link rel="self" href="http://cmisexample.oasis-
9133 open.org/rep1/38af30f5-020d-4274-b25c-0821b6db899b"/>
9134 <atom:link rel="edit" href="http://cmisexample.oasis-
9135 open.org/rep1/38af30f5-020d-4274-b25c-0821b6db899b"/>
9136 <atom:link type="application/cmis+xml;type=allowableActions"
9137 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9138 href="http://cmisexample.oasis-open.org/rep1/38af30f5-020d-4274-b25c-
9139 0821b6db899b/allowableactions"/>
9140 <atom:link type="application/atom+xml;type=entry" rel="describedby"
9141 href="http://cmisexample.oasis-open.org/rep1/38af30f5-020d-4274-b25c-
9142 0821b6db899b/type"/>
9143 <atom:link type="application/atomsvc+xml" rel="service"
9144 href="http://cmisexample.oasis-open.org/rep1//service"/>
9145 <atom:published>2009-10-19T10:10:01.000-07:00</atom:published>
9146 <atom:summary type="html">HTML summary of Entry 38af30f5-020d-4274-b25c-
9147 0821b6db899b</atom:summary>
9148 <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9149 open.org/ns/cmis/link/200908/source" href="http://cmisexample.oasis-
9150 open.org/rep1/38af30f5-020d-4274-b25c-0821b6db899b/source"/>
9151 <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9152 open.org/ns/cmis/link/200908/target" href="http://cmisexample.oasis-
9153 open.org/rep1/38af30f5-020d-4274-b25c-0821b6db899b/target"/>
9154 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9155 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9156 open.org/rep1/38af30f5-020d-4274-b25c-0821b6db899b/policies"/>
9157 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9158 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9159 open.org/rep1/38af30f5-020d-4274-b25c-0821b6db899b/acl"/>
9160 <cmisra:object>
9161 <cmis:properties>
9162 <cmis:propertyId localName="rep-cmis:objectId"
9163 propertyDefinitionId="cmis:objectId">
9164 <cmis:value>38af30f5-020d-4274-b25c-0821b6db899b</cmis:value>
9165 </cmis:propertyId>
9166 </cmis:properties>
9167 </cmisra:object>
9168 </atom:entry>
9169
9170
9171 Please also see the example documents included with the schema.
9172
```

9173 **3.10.5.2 PUT**

9174 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9175 properties are not included, the repository SHOULD NOT modify them.

9176

9177 The server SHOULD respond with:

- 9178 • HTTP Status Code 200
- 9179 • Response Body containing the updated atom entry

9180

9181 **3.10.5.3 DELETE**

9182 This removes the relationship entry.

9183 Successful HTTP code: 204

9184 **3.10.6 Policy Entry**

9185 This is a CMIS policy instance.

9186 CMIS Services:

9187     GET: getObject

9188     PUT: updateProperties

9189     DELETE: deleteObject or removePolicy

9190 Media Type: application/atom+xml;type=entry

9191

9192 Link Relations:

- 9193 • self

- 9194 • edit

- 9195 • service: Points to service document containing the CMIS repository. The service document  
9196 MUST contain only one workspace element.

9197         o Media Type: application/atomsrv+xml

- 9198 • describedby: Points to the type definition as an atom entry for the type of this policy entry.

- 9199 • alternate: this is used to identify the renditions available for the specified object. Please see the  
9200 Renditions section.

- 9201 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions  
9202 document for this object.

- 9203 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.

- 9204 • <http://docs.oasis-open.org/ns/cmis/link/200908/act>: Points to ACL document for this object

9205

9206 The following element MUST be included inside the atom entry:

- 9207 • cmisra:object

9208

9209 **3.10.6.1 GET**

9210 The following arguments may be supplied. Please see the domain model for more information:

- 9211 • filter

- 9212 • includeAllowableActions

- 9213     • includeRelationships  
 9214     • renditionFilter  
     ○ If not specified, renditions will not be included.

9216  
 9217 Request:

```
9218 GET /obj/408fb453-4a18-4103-b495-e84a2a2761a0?filter=cmis:objectId HTTP/1.1
9219 Host: example.org
9220
```

9221

9222 Response:

```
9223 HTTP/1.1 200 Ok
9224 Date: Mon, 19 Oct 2009 10:10:00 -0700
9225 Content-Length: 2608
9226 Content-Type: application/atom+xml;type=entry
9227 Location: /obj/408fb453-4a18-4103-b495-e84a2a2761a0?filter=cmis:objectId
9228
9229
9230 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9231 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9232 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9233 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9234 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9235 open.org/ns/cmis/restatom/200908/">
9236 <atom:author>
9237 <atom:name>Al Brown</atom:name>
9238 <atom:uri>http://www.ibm.com/</atom:uri>
9239 <atom:email>albertcbrown@us.ibm.com</atom:email>
9240 </atom:author>
9241 <atom:content src="http://cmisexample.oasis-open.org/repl/408fb453-4a18-
9242 4103-b495-e84a2a2761a0"/>
9243 <atom:id>urn:uuid:408fb453-4a18-4103-b495-e84a2a2761a0</atom:id>
9244 <atom:title type="text">Security Policy</atom:title>
9245 <atom:updated>2009-10-19T10:10:00.937-07:00</atom:updated>
9246 <atom:link rel="self" href="http://cmisexample.oasis-
9247 open.org/repl/408fb453-4a18-4103-b495-e84a2a2761a0"/>
9248 <atom:link rel="edit" href="http://cmisexample.oasis-
9249 open.org/repl/408fb453-4a18-4103-b495-e84a2a2761a0"/>
9250 <atom:link type="application/cmis+xml;type=allowableActions"
9251 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9252 href="http://cmisexample.oasis-open.org/repl/408fb453-4a18-4103-b495-
9253 e84a2a2761a0/allowableactions"/>
9254 <atom:link type="application/atom+xml;type=entry" rel="describedby"
9255 href="http://cmisexample.oasis-open.org/repl/408fb453-4a18-4103-b495-
9256 e84a2a2761a0/type"/>
9257 <atom:link type="application/atomsvc+xml" rel="service"
9258 href="http://cmisexample.oasis-open.org/repl//service"/>
9259 <atom:published>2009-10-19T10:10:00.937-07:00</atom:published>
9260 <atom:summary type="html">HTML summary of Entry 408fb453-4a18-4103-b495-
9261 e84a2a2761a0</atom:summary>
9262 <atom:link type="application/atom+xml;type=feed" rel="up"
9263 href="http://cmisexample.oasis-open.org/repl/408fb453-4a18-4103-b495-
9264 e84a2a2761a0/parents"/>
9265 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9266 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9267 open.org/repl/408fb453-4a18-4103-b495-e84a2a2761a0/relationships"/>
9268 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9269 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9270 open.org/repl/408fb453-4a18-4103-b495-e84a2a2761a0/acl"/>
9271 <cmisra:object>
9272 <cmis:properties>
```

```
9273 <cmis:propertyId localName="rep-cmis:objectId"
9274 propertyDefinitionId="cmis:objectId">
9275 <cmis:value>408fb453-4a18-4103-b495-e84a2a2761a0</cmis:value>
9276 </cmis:propertyId>
9277 </cmis:properties>
9278 </cmisra:object>
9279 </atom:entry>
9280
```

9281

9282 Please also see the example documents included with the schema.

9283

### 9284 **3.10.6.2 PUT**

9285 This does a replacement of the atom entry with the atom entry document specified. If read/write  
9286 properties are not included, the repository SHOULD NOT modify them.

9287

9288 The server SHOULD respond with:

- 9289     • HTTP Status Code 200  
9290     • Response Body containing the updated atom entry

9291

### 9292 **3.10.6.3 DELETE**

9293 This removes the policy entry. If this policy entry was discovered through a policy collection on an object,  
9294 then removePolicy() is performed rather than deleteObject() on the policy itself.

9295

9296 Success HTTP code: 204

## 9297 **3.10.7 Content Stream**

9298 This is the content stream portion of the document object.

9299 CMIS Services:

9300       GET: getContentStream

9301       PUT: setContentStream

9302       DELETE: deleteContentStream

9303 Media Type: Mime/Type of resource (mime type of content stream on document)

### 9304 **3.10.7.1 GET**

9305 This returns the content stream.

9306

9307 It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED  
9308 that HTTP compression is also supported.

9309

9310 Please see RFC2616 for more information on HTTP Range requests.

### 9311 **3.10.7.2 PUT**

9312 This does a replacement of the content stream.

9313

9314 The following optional arguments may be supplied. Please see the domain model for more information:

- 9315     • overwriteFlag.  
9316         ○ If not specified, this defaults to ‘true’ in this binding and behaves consistent with  
9317             AtomPub.

9318  
9319     Success HTTP code: 201

9320  
9321     Returns headers:  
9322         • Content-Location: URI for content stream  
9323         • Location: URI for content stream

### 9324     **3.10.7.3 DELETE**

9325     This removes the content stream.

## 9326     **3.10.8 ACL Resource**

9327     CMIS Services:

9328         GET: getACL  
9329         PUT: applyACL

9330  
9331     Media Type: application/cmisacl+xml  
9332

### 9333     **3.10.8.1 GET**

9334     This returns the CMIS ACL for a specified object. The client will follow the link on the atom entry to get  
9335     the CMIS ACL for that object.

9336

9337     Request:

```
9338 GET /objacl/59fe373b-0348-493e-baaf-21089b596f0a HTTP/1.1
9339 Host: example.org
```

9341

9342     Response:

```
9343 HTTP/1.1 200 Ok
9344 Date: Mon, 19 Oct 2009 10:10:00 -0700
9345 Content-Length: 758
9346 Content-Type: application/cmisacl+xml
9347 Location: /objacl/59fe373b-0348-493e-baaf-21089b596f0a
9348
9349
9350 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9351 <cmis:acl xmlns:app="http://www.w3.org/2007/app"
9352 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9353 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9354 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9355 open.org/ns/cmis/restatom/200908/">
9356 <cmis:permission>
9357 <cmis:principal>
9358 <cmis:principalId>Al Brown</cmis:principalId>
9359 </cmis:principal>
9360 <cmis:permission>cmis:read</cmis:permission>
9361 <cmis:permission>cmis:write</cmis:permission>
9362 <cmis:permission>cmis:all</cmis:permission>
```

```
9363 <cmis:permission>publish</cmis:permission>
9364 <cmis:direct>true</cmis:direct>
9365 </cmis:permission>
9366 </cmis:acl>
9367
```

9368

9369 Please also see the example documents included with the schema.

---

## 9370 4 Web Services Binding

### 9371 4.1 Overview

9372 All services and operations defined in part I of the CMIS specification are presented in this Web Services  
9373 binding.

9374 The WSDL for these services reference two XSD documents. One defines elements for the primary data  
9375 types of documents, folders, relationships and policies as well as collections of these types of objects.  
9376 The second XSD defines the message formats for each of the CMIS services; the messages often refer  
9377 to the data types defined in the first XSD schema. The WSDL presents exactly the abstract services  
9378 defined in the services section of Part I of the CMIS specification.

9379 The normative CMIS Web Services binding is defined by the WSDL and XSD as well as the details given  
9380 here in this part of the CMIS specification except the examples.

#### 9381 4.1.1 WS-I

9382 A CMIS Web Services binding MUST comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

#### 9383 4.1.2 Authentication

9384 A CMIS Web Services binding SHOULD support WS-Security 1.1 for Username Token Profile 1.1 and  
9385 MAY also support other authentication mechanisms. A CMIS repository MAY grant access to all or a  
9386 subset of the CMIS services to unauthenticated clients.

#### 9387 4.1.3 Content Transfer

9388 A CMIS Web Services binding SHOULD support MTOM content transfers. It MUST accept content that is  
9389 base64 encoded.

#### 9390 4.1.4 Reporting Errors

9391 Services MUST report errors via SOAP faults. The CMIS-Messaging.xsd defines a basic fault structure  
9392 that includes an error code and an error message and the WSDL for each service defines specific  
9393 messages that have the basic fault format.

### 9394 4.2 Web Services Binding Mapping

9395 The Domain Model in Part I of the CMIS specification defines all services, operations, parameters and  
9396 objects of CMIS. The Web Services binding is an exact one-to-one mapping of this definition with small  
9397 exceptions that are explained in the next section. Operations and parameters are named exactly after  
9398 their counterparts in Part I. All rules and exceptions defined in Part I apply to the Web Services binding.  
9399 Optional parameters and optional return values are not set if they are missing or their value is NULL.

### 9400 4.3 Additions to Part I

#### 9401 4.3.1 updateProperties and checkIn Semantics

9402 This binding supports partial properties updates. All properties passed to updateProperties or checkIn will  
9403 be updated to their new values. Properties that are passed without a value will be set to their default  
9404 value or un-set if no default value is defined. All others property values remain untouched.

## 9405 **4.3.2 Content Ranges**

9406 This binding supports the retrieval of content ranges. The operation `getContentType` accepts two  
9407 optional parameters:

- 9408 • **Integer offset:** The first byte of the content to retrieve. Default value is 0.
- 9409 • **Integer length:** The length of the range in bytes. Default value is the size of the content minus  
9410 the offset.

9411 If the **offset** value is greater than the size of the content the repository SHOULD throw a constraint  
9412 exception.

9413 If **offset + length** is greater than the size of the content the repository should deliver the content from the  
9414 offset to the end of the content.

## 9417 **4.3.3 Extensions**

9418 On all input messages and some output messages exists an element called extension. This element is  
9419 used to provide vendor or repository-specific information between client and server.

9420 All of the types referenced by the schema also support `xs:any` for vendor or repository-specific  
9421 information.

## 9422 **4.3.4 Web Services Specific Structures**

9423 This binding requires specific structures that are not part of the general CMIS schema.

9424 Please also see the example request and response documents included with the schema.

### 9425 **4.3.4.1 cmisFaultType and cmisFault**

9426 `cmisFaultType` and `cmisFault` SHOULD be used to generate SOAP faults. See 4.1.4 Reporting  
9427 Errors.

### 9428 **4.3.4.2 cmisRepositoryEntryType**

9429 `cmisRepositoryEntryType` is the return structure of `getRepositories`. It contains the id and the name  
9430 of a repository.

### 9431 **4.3.4.3 cmisTypeContainer**

9432 `cmisTypeContainer` is the return structure of `getTypeDescendants`. It holds a type hierarchy.

### 9433 **4.3.4.4 cmisTypeDefinListType**

9434 `cmisTypeDefinListType` is the return structure of `getTypeChildren`. It contains a list of types,  
9435 the `hasMoreItems` flag and the `numItem` element.

### 9436 **4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and 9437 cmisObjectInFolderContainerType**

9438 `cmisObjectInFolderType` holds, in addition to a `cmisObjectType` object, a path segment string. It  
9439 is used in all operations that support the `includePathSegments` parameter.

9440 `cmisObjectParentsType` is similar but has a relative path segment string instead of a path segment.  
9441 For details about path segments and relative path segments see Part I section 2.5.3 Paths.

9442 `cmisObjectInFolderContainerType` contains a folder hierarchy.

9443 **4.3.4.6 cmisObjectType and cmisObjectInFolderListType**

9444 `cmisObjectType` and `cmisObjectInFolderListType` hold lists of `cmisObjectType` and  
9445 `cmisObjectInFolderType` structures. They also contain the `hasMoreItems` flag and the `numItems`  
9446 element that are returned by operations that return these lists.

9447 **4.3.4.7 cmisContentStreamType**

9448 `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.<br>If set it MUST be a positive number.<br>If the length is unknown it MUST NOT be set.                                      | SHOULD be set        | SHOULD be set        |
| MimeType | MIME Media Type of the content stream.<br>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.<br>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.<br>MUST be present even if the content stream has 0 bytes.                                                                                      | MUST be set          | MUST be set          |

9449

9450 **4.3.4.8 cmisACLType**

9451 `cmisACLType` is the return structure of `getACL` and `applyACL`. It contains the current Access Control List  
9452 (ACL) of the object and the `exact` flag that indicates if the ACL fully describes the permission of this  
9453 object.

9454 **4.3.4.9 cmisExtensionType**

9455 `cmisExtensionType` is a placeholder for extensions. See 4.3.3 Extensions.

9456

---

9457    **5 IANA Considerations**

9458    **5.1 Content-Type Registration**

9459    **5.1.1 CMIS Query**

9460    A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

9461

9462    MIME media type name: application

9463    MIME subtype name: cmisquery +xml

9464    Mandatory parameters: None

9465    Optional parameters:

9466        "charset": This parameter has semantics identical to the charset parameter of the  
9467        "application/xml" media type as specified in [RFC3023].

9468    Encoding considerations:

9469        Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

9470    Security considerations: As defined in this specification.

9471        In addition, as this media type uses the "+xml" convention, it shares the same security  
9472        considerations as described in [RFC3023], Section 10.

9473    Interoperability considerations:

9474        There are no known interoperability issues.

9475    Published specification: This specification.

9476    Applications that use this media type:

9477        No known applications currently use this media type.

9478    Additional information:

9479    Magic number(s):

9480        As specified for "application/xml" in [RFC3023], Section 3.2.

9481    File extension: .cmisquery

9482    Fragment identifiers:

9483        As specified for "application/xml" in [RFC3023], Section 5.

9484    Base URI:

9485        As specified in [RFC3023], Section 6.

9486    Macintosh File Type code: TEXT

9487    Person and email address to contact for further information:

9488        Al Brown <albertcbrown@us.ibm.com>

9489    Intended usage: COMMON

9490    Author/Change controller: IESG

9491    **5.1.2 CMIS AllowableActions**

9492    A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following  
9493    media type:

9494

9495   MIME media type name: application  
9496   MIME subtype name: cmisallowableactions +xml  
9497   Mandatory parameters: None.  
9498   Optional parameters:  
9499       "charset": This parameter has semantics identical to the charset parameter of the  
9500       "application/xml" media type as specified in [RFC3023].  
9501   Encoding considerations:  
9502       Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9503   Security considerations: As defined in this specification.  
9504       In addition, as this media type uses the "+xml" convention, it shares the same security  
9505       considerations as described in [RFC3023], Section 10.  
9506   Interoperability considerations:  
9507       There are no known interoperability issues.  
9508   Published specification: This specification.  
9509   Applications that use this media type:  
9510       No known applications currently use this media type.  
9511   Additional information:  
9512   Magic number(s):  
9513       As specified for "application/xml" in [RFC3023], Section 3.2.  
9514   File extension: .cmisallowableactions  
9515   Fragment identifiers:  
9516       As specified for "application/xml" in [RFC3023], Section 5.  
9517   Base URI:  
9518       As specified in [RFC3023], Section 6.  
9519   Macintosh File Type code: TEXT  
9520   Person and email address to contact for further information:  
9521       Al Brown <albertbrown@us.ibm.com>  
9522   Intended usage: COMMON  
9523   Author/Change controller: IESG  
9524

### 9525   **5.1.3 CMIS Tree**

9526   A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:  
9527  
9528   MIME media type name: application  
9529   MIME subtype name: cmistree +xml  
9530   Mandatory parameters: None.  
9531   Optional parameters:  
9532       "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9533       type as specified in [RFC3023].  
9534   Encoding considerations:  
9535       Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9536   Security considerations: As defined in this specification.

9537 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9538 described in [RFC3023], Section 10.

9539 Interoperability considerations:

9540 There are no known interoperability issues.

9541 Published specification: This specification.

9542 Applications that use this media type:

9543 No known applications currently use this media type.

9544 Additional information:

9545 Magic number(s):

9546 As specified for "application/xml" in [RFC3023], Section 3.2.

9547 File extension: .cmistree

9548 Fragment identifiers:

9549 As specified for "application/xml" in [RFC3023], Section 5.

9550 Base URI:

9551 As specified in [RFC3023], Section 6.

9552 Macintosh File Type code: TEXT

9553 Person and email address to contact for further information:  
9554 Al Brown <albertcbrown@us.ibm.com>

9555 Intended usage: COMMON

9556 Author/Change controller: IESG

9557

#### 9558 **5.1.4 CMIS Atom**

9559 A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:

9560

9561 MIME media type name: application

9562 MIME subtype name: cmisatom +xml

9563 Mandatory parameters: None.

9564 Optional parameters:

9565 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9566 type as specified in [RFC3023].

9567 "type": This parameter has semantics identical to the type parameter of the "application/atom+xml" as  
9568 specified in [RFC4287]

9569 Encoding considerations:

9570 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

9571 Security considerations: As defined in this specification.

9572 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9573 described in [RFC3023], Section 10.

9574 Interoperability considerations:

9575 There are no known interoperability issues.

9576 Published specification: This specification.

9577 Applications that use this media type:

9578 No known applications currently use this media type.

9579 Additional information:

9580     Magic number(s):  
9581     As specified for "application/xml" in [RFC3023], Section 3.2.  
9582     File extension: .cmisatom  
9583     Fragment identifiers:  
9584     As specified for "application/xml" in [RFC3023], Section 5.  
9585     Base URI:  
9586     As specified in [RFC3023], Section 6.  
9587     Macintosh File Type code: TEXT  
9588     Person and email address to contact for further information:  
9589     AI Brown <albertcbrown@us.ibm.com>  
9590     Intended usage: COMMON  
9591     Author/Change controller: IESG  
9592  
9593     Please see section 3.1.1 on why this media type is needed above the Atom Media Type.

## 9594     **5.1.5 CMIS ACL**

9595     A CMIS ACL Document, when serialized as XML 1.0, can be identified with the following media type:  
9596  
9597     MIME media type name: application  
9598     MIME subtype name: cmisacl +xml  
9599     Mandatory parameters: None.  
9600     Optional parameters:  
9601     "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9602     type as specified in [RFC3023].  
9603     Encoding considerations:  
9604     Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9605     Security considerations: As defined in this specification.  
9606     In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9607     described in [RFC3023], Section 10.  
9608     Interoperability considerations:  
9609     There are no known interoperability issues.  
9610     Published specification: This specification.  
9611     Applications that use this media type:  
9612     No known applications currently use this media type.  
9613     Additional information:  
9614     Magic number(s):  
9615     As specified for "application/xml" in [RFC3023], Section 3.2.  
9616     File extension: .cmisacl  
9617     Fragment identifiers:  
9618     As specified for "application/xml" in [RFC3023], Section 5.  
9619     Base URI:  
9620     As specified in [RFC3023], Section 6.  
9621     Macintosh File Type code: TEXT

9622 Person and email address to contact for further information:  
9623 Al Brown <albertcbrown@us.ibm.com>  
9624 Intended usage: COMMON  
9625 Author/Change controller: IESG  
9626

---

## 6 Conformance

9627 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level requirements defined within this specification.

9629

9630 Specification:

9631 This specification references a number of other specifications (see the table above). In order to comply with this specification, an implementation MUST implement the portions of referenced specifications necessary to comply with the required provisions of this specification. Additionally, the implementation of the portions of the referenced specifications that are specifically cited in this specification MUST comply with the rules for those portions as established in the referenced specification.

9632

9633

9634

9635

9636

9637

9638 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level requirements defined within this specification.

9639

9640

9641

9642 Domain Model:

9643 Normative text within this specification takes precedence over the CMIS Core XML Schema.

9644 That is, the normative text in this specification further constrains the schemas and/or WSDL that are part of this specification; and this specification contains further constraints on the elements defined in referenced schemas.

9645

9646

9647

9648 Clients:

9649 Client implementations MAY implement either Restful AtomPub Binding or the Web Services Binding.

9650

9651

9652 Repositories:

9653 Repositories MUST implement the following CMIS protocol bindings:

9654     ▪ Restful AtomPub Binding

9655     ▪ Web Services Binding

9656

9657 Rest Binding:

9658 This specification references a number of other specifications. In order to comply with this specification, an implementation MUST implement the portions of referenced specifications necessary to comply with the required provisions of this specification. Additionally, the implementation of the portions of the referenced specifications that are specifically cited in this specification MUST comply with the rules for those portions as established in the referenced specification.

9659

9660

9661

9662

9663

9664 Additionally normative text within this specification takes precedence over the CMIS RestAtom XML Schema. That is, the normative text in this specification further constrains the schemas and/or WSDL that are part of this specification; and this specification contains further constraints on the elements defined in referenced schemas.

9665

9666

9667

9668 The CMIS RestAtom XML takes precedence over any examples or non-normative outlines included either in this document or as standalone examples.

9669

9670

9671 Web Services Binding:

9672        Normative text within this specification takes precedence over the CMIS Messaging XML and  
9673        CMIS WSDL. That is, the normative text in this specification further constrains the schemas and  
9674        WSDL that are part of this specification; and this specification contains further constraints on the  
9675        elements defined in referenced schemas.

9676        The CMIS Messaging XML and CMIS WSDL takes precedence over any examples or non-  
9677        normative outlines included either in this document or as standalone examples.

---

## 9678 A. Acknowledgements

9679 The following individuals have participated in the creation of this specification and are gratefully  
9680 acknowledged:

9681

### 9682 Participants:

9683 Philippe Allart, Adullact  
9684 Florian Bartels, fme AG  
9685 Fred Boiscuvier, Exalead, Inc.  
9686 Al Brown, IBM  
9687 Jay Brown, IBM  
9688 Mark Carlson, Sun Microsystems  
9689 Derek Carr, IBM  
9690 David Caruana, Alfresco Software  
9691 Eric Chan, Oracle Corporation  
9692 Sameer Charles, Magnolia International AG  
9693 David Choy, EMC Corporation  
9694 Scott Conroy, Individual  
9695 Cornelia Davis, EMC Corporation  
9696 Kevin Dorr, Flatirons Solutions Corporation  
9697 Betsy Fanning, AIIM  
9698 Steffen Frederiksen, Content Technologies ApS  
9699 Stephan Friedl, Quark  
9700 Dustin Friesenhahn, Microsoft Corporation  
9701 Gary Gershon, Individual  
9702 Paul Goetz, SAP AG  
9703 Gregory Grefenstette, Exalead, Inc.  
9704 Florent Guillaume, Nuxeo  
9705 Ethan Gur-esh, Microsoft Corporation  
9706 Martin Hermes, SAP AG  
9707 Jens Huebel, Open Text Corporation  
9708 David Izatt, Structured Software Systems Limited (3SL)  
9709 Gershon Janssen, Individual  
9710 Volker John, Saperion AG  
9711 Shane Johnson, Citytech, Inc.  
9712 Christophe Kijewska, Adullact  
9713 Ijonas Kisselbach, Vamosa  
9714 Mark Klamerus, Individual  
9715 Stephan Klevenz, SAP AG  
9716 Boris Kraft, Magnolia International AG  
9717 Alison Macmillan, Oracle Corporation  
9718 Michael Marth, Day Software  
9719 Ryan McVeigh, Oracle Corporation  
9720 Juerg Meier, fme AG  
9721 Gregory Melahn, IBM  
9722 Pat Miller, Microsoft Corporation  
9723 Florian Mueller, Open Text Corporation  
9724 Thomas Mueller, Day Software  
9725 John Newton, Alfresco Software  
9726 David Nuescheler, Day Software  
9727 Dominique Pfister, Day Software  
9728 Peeter Piegaze, Day Software  
9729 David Pitfield, Oracle Corporation  
9730 Norrie Quinn, EMC Corporation

9731           Craig Randall, Adobe Corporation  
9732           Celso Rodriguez, ASG Software Solutions  
9733           Steve Roth, Oracle Corporation  
9734           Patrick Ryan, IBM  
9735           Angela Schreiber, Day Software  
9736           Spencer Shearer, Exalead, Inc.  
9737           Madi Solomon, Pearson PLC  
9738           Wojciech Specht, fme AG  
9739           Maik Uhlenberg, fme AG  
9740           Oliver Walthard, Day Software  
9741           Patrick Ward, Booz Allen Hamilton

9742  
9743         Original Authors of the initial contribution:

9744           Al Brown, IBM  
9745           David Choy, EMC  
9746           Cornelia Davis, EMC  
9747           Ethan Gur-Esh, Microsoft

9748  
9749         Original Acknowledgements of the initial contribution:

9750           Al Brown, IBM  
9751           David Caruana, Alfresco  
9752           Derek Carr, IBM  
9753           David Choy, EMC  
9754           Cornelia Davis, EMC  
9755           Paul Goetz, SAP  
9756           Ethan Gur-Esh, Microsoft  
9757           Martin Hermes, SAP  
9758           Jens Hubel, OpenText  
9759           Jay Brown, IBM  
9760           Ryan McVeigh, Oracle  
9761           Gregory Melahn, IBM  
9762           Florian Mueller, OpenText  
9763           John Newton, Alfresco  
9764           Norrie Quinn, EMC  
9765           Steve Roth, Oracle  
9766           Craig Randall, EMC

## B. Non-Normative Text

---

9768

## C. Revision History

9769

| Revision | Date | Editor   | Changes Made        |
|----------|------|----------|---------------------|
| 1.0      |      | Al Brown | First specification |

9770