

Content Management Interoperability Services (CMIS) Version 1.0

Committee Specification 01

12 March 2010

Specification URIs:

This Version:

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

Previous Version:

<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cd07/cmisis-spec-v1.0.doc> (Authoritative)
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cd07/cmisis-spec-v1.0.html>
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/cd07/cmisis-spec-v1.0.pdf>

Latest Version:

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

Technical Committee:

OASIS Content Management Interoperability Services (CMIS) TC

Chair(s):

David Choy, EMC

Editor(s):

Al Brown, IBM
Ethan Gur-Esh, Microsoft
Ryan McVeigh, Oracle
Florian Müller, OpenText

Related work:

N/A

Declared XML Namespace(s):

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

Abstract:

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

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

Status:

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

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

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/cmisis/ipr.php>).

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

Notices

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

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

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

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

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

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

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

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

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

Table of Contents

Committee Specification 01	1
Specification URIs:	1
This Version:	1
Previous Version:	1
Latest Version:	1
Technical Committee:	1
Chair(s):	1
David Choy, EMC	1
Editor(s):	1
Al Brown, IBM	1
Related work:	1
N/A	1
Declared XML Namespace(s):	1
Abstract:	2
Status:	2
Notices	3
Table of Contents	4
1 Introduction	15
1.1 Terminology	15
1.2 Normative References	15
1.3 Non-Normative References	15
2 Domain Model	16
2.1 Data Model	16
2.1.1 Repository	16
2.1.1.1 Optional Capabilities	16
Navigation Capabilities:	16
Object Capabilities:	17
Filing Capabilities:	17
Ability for an application to file a document or other file-able object in more than one folder	17
Ability for an application to leave a document or other file-able object not filed in any folder	17
Ability for an application to file individual versions (i.e., not all versions) of a document in a folder	17
Versioning Capabilities:	17
Ability for an application to update the “Private Working Copy” of a checked-out document	18
Query Capabilities:	18
ACL Capabilities:	18
Indicates the level of support for ACLs by the repository	18
See Section: 2.8 Access Control	19
2.1.1.2 Implementation Information	19
2.1.2 Object	19
2.1.2.1 Property	20
2.1.3 Object-Type	21
2.1.3.1 Object-Type Hierarchy and Inheritance	21
2.1.3.2 Object-Type Attributes	22

2.1.3.3 Object-Type Property Definitions	24
2.1.4 Document Object.....	28
2.1.4.1 Content Stream.....	29
2.1.4.2 Renditions	29
2.1.4.3 Document Object-Type Definition	30
2.1.5 Folder Object.....	38
2.1.5.1 File-able Objects	38
2.1.5.2 Folder Hierarchy	39
2.1.5.3 Paths.....	40
2.1.5.4 Folder Object-Type Definition	41
2.1.6 Relationship Object	45
2.1.6.1 Relationship Object-Type Definition.....	46
2.1.7 Policy Object.....	51
2.1.7.1 Policy Object-Type Definition	52
2.1.8 Access Control	55
2.1.8.1 ACL, ACE, Principal, and Permission	55
2.1.8.2 CMIS Permissions	55
2.1.8.3 ACL Capabilities	56
Navigation Services:	57
Object Services:	58
Filing Services:	61
Versioning Services:	61
Relationship Services:	62
ACL Services:	63
2.1.9 Versioning.....	65
2.1.9.1 Version Series.....	65
2.1.9.2 Latest Version	65
2.1.9.3 Major Versions	65
2.1.9.4 Services that modify Version Series	66
2.1.9.5 Versioning Properties on Document Objects	67
2.1.9.6 Document Creation and Initial Versioning State.....	68
2.1.9.7 Version Specific/Independent membership in Folders	68
2.1.9.8 Version Specific/Independent membership in Relationships	68
2.1.9.9 Versioning visibility in Query Services	69
2.1.10 Query	69
2.1.10.1 Relational View Projection of the CMIS Data Model	70
2.1.10.2 Query Language Definition	71
<table reference> [<join type>] JOIN <table reference> <join specification>	73
<quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>	73
<text search predicate> ::= CONTAINS "("	73
Decimal	75
Integer	76
Boolean	76
DateTime.....	76
ID.....	76
ID (IN).....	76
Supported Operators: [NOT] IN	76

URI	76
URI (IN)	76
Supported Operators: [NOT] IN	76
URI	76
Supported Operators: [NOT] LIKE	77
Operations on the SCORE() output MUST be treated the same as decimal operations.	77
The SQL-92 <quantifier> is restricted to ANY only.	77
Example:	78
Inputs:	78
Return value:	78
Constraints:	78
BNF grammar structure: SCORE ()	79
Return value:	79
Constraints:	79
Inputs:	79
Return value:	79
Inputs:	80
Return value:	80
2.1.10.3 Escaping	80
2.1.11 Change Log	80
2.1.11.1 Completeness of the Change Log	81
2.1.11.2 Change Log Token	81
2.1.11.3 Change Event	81
2.2 Services	81
2.2.1 Common Service Elements	82
2.2.1.1 Paging	82
Input Parameters:	82
Output Parameters:	82
2.2.1.2 Retrieving additional information on objects in CMIS service calls	82
Optional Input Parameter:	82
Optional Input Parameter:	83
Output Parameter for each object:	83
Optional Input Parameter:	83
Output Parameter or each object:	83
Optional Input Parameter:	83
Output Parameter for each object:	83
Optional Input Parameter:	84
Output Parameter for each object:	84
Optional Input Parameter:	84
Output Parameter for each object:	84
2.2.1.3 Change Tokens	84
2.2.1.4 Exceptions	85
2.2.1.5 ACLs	88
2.2.2 Repository Services	88
2.2.2.1 getRepositories	88
2.2.2.2 getRepositoryInfo	89

Required:	89
2.2.2.3 getTypeChildren.....	90
Required:	90
Optional:	90
Optional:	91
2.2.2.4 getTypeDescendants.....	91
Notes:	91
Required:	91
Optional:	91
2.2.2.5 getTypeDefinition.....	92
Required:	92
2.2.3 Navigation Services.....	92
2.2.3.1 getChildren	92
Notes:	92
Required:	92
Optional:	92
Optional:	93
2.2.3.2 getDescendants.....	93
Notes:	93
Required:	93
Optional:	94
2.2.3.3 getFolderTree	94
Notes:	94
Required:	94
Optional:	95
2.2.3.4 getFolderParent	95
Required:	95
Optional:	95
2.2.3.5 getObjectParents	96
Required:	96
Optional:	96
2.2.3.6 getCheckedOutDocs.....	97
Required:	97
Optional:	97
Optional:	97
2.2.4 Object Services	97
2.2.4.1 createDocument.....	98
Required:	98
Optional:	98
2.2.4.2 createDocumentFromSource.....	99
Required:	99
Optional:	99
2.2.4.3 createFolder.....	100
Required:	100
Optional:	100
2.2.4.4 createRelationship	101

Required:	101
Optional:	101
2.2.4.5 createPolicy	102
Required:	102
Optional:	102
2.2.4.6 getAllowableActions.....	103
Required:	103
2.2.4.7 getObject	103
Required:	103
Optional:	104
2.2.4.8 getProperties.....	104
Required:	104
Optional:	104
2.2.4.9 getObjectByPath	104
Required:	105
Optional:	105
2.2.4.10 getContentStream	105
Required:	105
Optional:	105
2.2.4.11 getRenditions	106
Required:	106
Optional:	106
2.2.4.12 updateProperties.....	106
Notes:	106
Required:	106
Optional:	106
2.2.4.13 moveObject.....	107
Required:	107
2.2.4.14 deleteObject.....	108
Required:	108
Optional:	108
2.2.4.15 deleteTree.....	108
Notes:	108
Required:	108
Optional:	108
2.2.4.16 setContentStream	109
Required:	109
Optional:	109
2.2.4.17 deleteContentStream	110
Required:	110
Optional:	110
2.2.5 Multi-filing Services	110
2.2.5.1 addObjectToFolder	110
Required:	110
Optional:	110
2.2.5.2 removeObjectFromFolder	111
Required:	111

Optional:	111
2.2.6 Discovery Services.....	111
2.2.6.1 query.....	111
Required:	111
Optional:	111
Optional:	112
2.2.6.2 getContentChanges.....	112
Notes:	112
Required:	112
Optional:	112
Optional:	113
2.2.7 Versioning Services.....	113
2.2.7.1 checkOut.....	113
Required:	114
2.2.7.2 cancelCheckOut.....	114
Required:	114
2.2.7.3 checkIn	114
Notes:	114
Required:	115
Optional:	115
2.2.7.4 getObjectOfLatestVersion.....	115
Required:	115
Optional:	115
2.2.7.5 getPropertiesOfLatestVersion.....	116
Required:	116
Optional:	116
2.2.7.6 getAllVersions.....	117
Notes:	117
Required:	117
Optional:	117
2.2.8 Relationship Services.....	117
2.2.8.1 getObjectRelationships.....	117
Required:	117
Optional:	118
Optional:	118
2.2.9 Policy Services.....	118
2.2.9.1 applyPolicy.....	119
Required:	119
2.2.9.2 removePolicy	119
Required:	119
2.2.9.3 getAppliedPolicies.....	119
Required:	119
Optional:	119
2.2.10 ACL Services.....	120
2.2.10.1 getACL.....	120
Required:	120

Optional:	120
Optional:	120
2.2.10.2 applyACL	120
Required:	120
Optional:	120
Optional:	121
3 Restful AtomPub Binding	122
3.1 Overview	122
3.1.1 Namespaces.....	122
3.1.2 Authentication.....	122
3.1.3 Response Formats	122
3.1.4 Optional Arguments.....	123
3.1.5 Errors and Exceptions.....	123
3.1.6 Renditions.....	123
The following attributes MAY be included.....	123
3.1.7 Content Streams	123
3.1.8 Paging of Feeds	123
3.1.9 Services not Exposed.....	123
3.1.9.1 removePolicy	124
3.2 HTTP.....	124
3.2.1 Entity Tag	124
3.2.2 HTTP Range.....	124
3.2.3 HTTP OPTIONS Method.....	124
3.2.4 HTTP Status Codes	124
3.2.4.1 General CMIS Exceptions.....	124
3.2.4.2 Notable HTTP Status Codes.....	125
Please see RFC2616 Section 10 for more information.	125
3.3 Media Types	125
3.3.1 CMIS Atom	125
3.3.2 CMIS Query.....	126
This document contains the representation of a query to be executed in a CMIS repository.	126
Please also see the example documents included with the schema.....	127
3.3.3 CMIS Allowable Actions	127
Please also see the example documents included with the schema.....	127
3.3.4 CMIS Tree	128
This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.	128
Please see Section 3.3.2.1 for more information.....	128
Please also see the example documents included with the schema.....	132
3.3.5 CMIS ACL.....	132
This document specifies an Access Control List based on the schema in CMIS Domain Model.....	132
Please also see the example documents included with the schema.....	133
3.4 Atom Extensions for CMIS.....	133
3.4.1 Atom Element Extensions	133
3.4.1.1 AtomPub Workspace	133
3.4.1.2 Atom Feed	133

3.4.1.3 Atom Entry	133
3.4.2 Attributes	134
3.4.2.1 cmisra:id	134
Please also see the example documents included with the schema.....	134
3.4.2.2 cmisra:renditionKind	134
3.4.3 CMIS Link Relations.....	135
3.4.3.1 Existing Link Relations.....	135
3.4.3.2 Hierarchy Navigation Internet Draft Link Relations	137
3.4.3.3 Versioning Internet Draft Link Relations.....	137
3.4.3.4 CMIS Specific Link Relations.....	137
3.5 Atom Resources	139
3.5.1 Feeds.....	139
If on the root type, all fields are repository specific.	139
3.5.2 Entries	140
3.5.2.1 Hierarchical Atom Entries	141
3.6 AtomPub Service Document (Repository).....	142
Media Type: application/atomsvc+xml.....	142
The repository MUST include the URI templates in the workspace elements.....	143
3.6.1 URI Templates	144
Repositories MUST provide the URI Template query if the repository supports query.	144
For example, if the URI template that supports the variable {id} is.....	144
All variables MUST be in the template.	144
<xs:complexType name="cmisUriTemplateType">	144
</xs:sequence>	144
Please also see the example documents included with the schema.....	145
3.6.1.1 Object By Id	145
Type: objectbyid	145
Service: getObjectById.....	145
3.6.1.2 Object By Path	146
Type: objectbypath.....	146
Service: getObjectByPath	146
Please also see the example documents included with the schema.....	147
3.6.1.3 Query	147
Service: query	147
Please also see the example documents included with the schema.....	147
3.6.1.4 Type By Id.....	147
Service: getTypeDefinition	147
Please also see the example documents included with the schema.....	148
3.6.2 HTTP Methods	148
3.6.2.1 GET	148
3.7 Service Collections	148
3.7.1 Root Folder Collection.....	148
3.7.2 Query Collection.....	148
Media Type: application/atom+xml;type=feed	149
3.7.2.1 POST	149
The feed returned MUST contain a set of atom entries representing the result set from the query.....	149

Please see http://tools.ietf.org/html/rfc5023#section-5.3	149
Please also see the example documents included with the schema.....	151
3.7.3 Checked Out Collection.....	151
3.7.3.1 GET	152
3.7.3.2 POST	152
Please also see the example documents included with the schema.....	155
3.7.4 Unfiled Collection	155
3.7.4.1 POST	156
Please also see the example documents included with the schema.....	159
3.7.5 Types Children Collection	159
This feed contains a set of atom entries for each child type definition.	160
3.7.5.1 GET	160
3.8 Collections	160
3.8.1 Relationships Collection	160
3.8.1.1 GET	161
3.8.1.2 POST	161
Please also see the example documents included with the schema.....	163
3.8.2 Folder Children Collection.....	163
Media Type: application/atom+xml;type=feed	164
3.8.2.1 GET	164
3.8.2.2 POST	165
Please also see the example documents included with the schema.....	168
Please also see the example documents included with the schema.....	172
3.8.3 Policies Collection	172
GET: getAppliedPolicies	172
Media Type: application/atom+xml;type=feed	172
3.8.3.1 GET	173
3.8.3.2 POST	173
3.8.3.3 DELETE	175
3.9 Feeds.....	175
3.9.1 Object Parents Feed	175
3.9.1.1 GET	178
3.9.2 Changes	178
This feed MUST be ordered from oldest first to newest.	179
3.9.2.1 GET	183
3.9.3 Folder Descendants	184
3.9.3.1 GET	189
3.9.3.2 DELETE	190
3.9.4 Folder Tree	190
This feed contains a set of atom entries for each sub-folder in the folder.	191
3.9.4.1 GET	193
3.9.4.2 DELETE	193
3.9.5 AllVersions Feed	193
The feed SHOULD contain the newest versions at the beginning of the feed.	193
This feed contains a set of atom entries for each version in the version series	194
3.9.5.1 GET	195
3.9.5.2 DELETE	195

Success HTTP code: 204	195
3.9.6 Type Descendants Feed	195
Media Type: application/atom+xml;type=feed	196
3.9.6.1 GET	204
3.10 Resources.....	204
3.10.1 Type Entry	204
3.10.1.1 GET	204
Please also see the example documents included with the schema.....	206
3.10.2 Document Entry.....	206
3.10.2.1 GET	207
Please also see the example documents included with the schema.....	208
3.10.2.2 PUT.....	208
3.10.2.3 DELETE	209
3.10.3 Document Private Working Copy (PWC) Entry.....	209
3.10.3.1 GET	209
Please also see the example documents included with the schema.....	211
3.10.3.2 PUT.....	211
3.10.3.3 DELETE.....	211
Success HTTP code: 204	212
3.10.4 Folder Entry.....	212
3.10.4.1 GET	212
Please also see the example documents included with the schema.....	214
3.10.4.2 PUT.....	214
3.10.4.3 DELETE.....	214
3.10.5 Relationship Entry	214
3.10.5.1 GET	215
Please also see the example documents included with the schema.....	216
3.10.5.2 PUT.....	216
3.10.5.3 DELETE	216
3.10.6 Policy Entry.....	216
3.10.6.1 GET	217
Please also see the example documents included with the schema.....	218
3.10.6.2 PUT.....	218
3.10.6.3 DELETE.....	219
Success HTTP code: 204	219
3.10.7 Content Stream	219
3.10.7.1 GET	219
3.10.7.2 PUT.....	219
Success HTTP code: 200 (with content), 204 (without content) or 201 if a new resource is created. Please see the HTTP specification for more information.....	219
3.10.7.3 DELETE	219
3.10.8 ACL Resource	219
Media Type: application/cmisacl+xml	220
3.10.8.1 GET	220
4 Web Services Binding	221
4.1 Overview	221
4.1.1 WS-I.....	221

4.1.2 Authentication.....	221
4.1.3 Content Transfer	221
4.1.4 Reporting Errors	221
4.2 Web Services Binding Mapping.....	221
4.3 Additions to the Services section.....	221
4.3.1 updateProperties and checkIn Semantics.....	221
4.3.2 Content Ranges	221
4.3.3 Extensions	222
4.3.4 Web Services Specific Structures	222
4.3.4.1 cmisFaultType and cmisFault	222
4.3.4.2 cmisRepositoryEntryType	222
4.3.4.3 cmisTypeContainer	222
4.3.4.4 cmisTypeDefinitionListType	222
4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and cmisObjectInFolderContainerType	222
4.3.4.6 cmisObjectListType and cmisObjectInFolderListType	222
4.3.4.7 cmisContentStreamType	223
4.3.4.8 cmisACLType.....	223
4.3.4.9 cmisExtensionType.....	223
5 IANA Considerations	224
5.1 Content-Type Registration.....	224
5.1.1 CMIS Query.....	224
MIME media type name: application.....	224
OASIS CMIS TC <cmis@lists.oasis-open.org>	224
5.1.2 CMIS AllowableActions	224
MIME media type name: application.....	225
OASIS CMIS TC <cmis@lists.oasis-open.org>	225
5.1.3 CMIS Tree	225
MIME media type name: application.....	225
OASIS CMIS TC <cmis@lists.oasis-open.org>	226
5.1.4 CMIS Atom	226
MIME media type name: application.....	226
OASIS CMIS TC <cmis@lists.oasis-open.org>	227
5.1.5 CMIS ACL.....	227
MIME media type name: application.....	227
File extension: .cmisacl	227
Macintosh File Type code: TEXT.....	227
OASIS CMIS TC <cmis@lists.oasis-open.org>	228
6 Conformance.....	229
▪ Restful AtomPub Binding	229
A. Acknowledgements	231
Participants:	231
B. Non-Normative Text	233
C. Revision History.....	234

1 Introduction

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

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

1.1 Terminology

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

1.2 Normative References

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

1.3 Non-Normative References

2 Domain Model

2.1 Data Model

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

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

2.1.1 Repository

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

2.1.1.1 Optional Capabilities

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

Navigation Capabilities:

capabilityGetDescendants

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

[See section: 2.2.3.2 getDescendants](#)

capabilityGetFolderTree

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

[See section: 2.2.3.1 getFolderTree](#)

Object Capabilities:

capabilityContentStreamUpdatability (**enumCapabilityContentStreamUpdates**)

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

- **none**: The content stream may never be updated.
- **anytime**: The content stream may be updated any time.
- **pwconly**: The content stream may be updated only when checked out. The abbreviation PWC is described in section 2.1.9 Versioning.

See Section: 2.1.4.1 Content Stream

capabilityChanges (**enumCapabilityChanges**)

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

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

See Section: 2.1.11 Change Log

capabilityRenditions (**enumCapabilityRendition**)

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

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

Filing Capabilities:

capabilityMultifiling

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

See Section: 2.1.5 Folder Object

capabilityUnfiling

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

See Section: 2.1.5 Folder Object

capabilityVersionSpecificFiling

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

See Section: 2.1.9 Versioning

Versioning Capabilities:

capabilityPWCUpdatable

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

See Section: 2.1.9 Versioning

capabilityPWCSearchable

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

See Section: 2.1.9 Versioning

capabilityAllVersionsSearchable

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

See Section: 2.1.9 Versioning

Query Capabilities:

capabilityQuery (enumCapabilityQuery)

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

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

See Section: 2.1.10 Query

capabilityJoin (enumCapabilityJoin)

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

- **none:** The repository cannot fulfill any queries that include any JOIN clauses.
- **inneronly:** The repository can fulfill queries that include an INNER JOIN clause, but cannot fulfill queries that include other types of JOIN clauses.
- **innerandouter:** The repository can fulfill queries that include any type of JOIN clause defined by the CMIS query grammar.

See Section: 2.1.10 Query

ACL Capabilities:

capabilityACL (enumCapabilityACL)

Indicates the level of support for ACLs by the repository

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

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

See Section: 2.8 Access Control

2.1.1.2 Implementation Information

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

2.1.2 Object

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

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

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

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

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

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

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

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

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

208 2.1.2.1 Property

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

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

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

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

- 223 • `string` (xsd:string)
- 224 • `boolean` (xsd:boolean)
- 225 • `decimal` (see section 2.1.3.3.4 Attributes specific to Decimal Object-Type Property Definitions)
- 226 • `integer` (xsd:integer)
- 227 • `datetime` (xsd:dateTime and see section 2.1.3.3.4 Attributes specific to Decimal Object-Type
228 Property Definitions)
- 229 • `uri` (xsd:anyURI)

230

231

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

- 233 • `id`
- 234 • `html`

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

236

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

240

241 The string MUST NOT contain:

- 242 • whitespace “ ”,
- 243 • comma “,”
- 244 • double quotes “”
- 245 • single quotes “”
- 246 • backslash “\”
- 247 • the period “.” character or,

- the open “(“ or close “)” parenthesis characters.

2.1.2.1.1 ID Property

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

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

2.1.2.1.2 HTML Property

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

2.1.3 Object-Type

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.

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.

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

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

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.

2.1.3.1 Object-Type Hierarchy and Inheritance

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

- A CMIS repository MUST have these base types:
 - *cmis:document* object-type

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

316 2.1.3.2 Object-Type Attributes

317 2.1.3.2.1 Attributes common to ALL Object-Type Definitions

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

- | | | |
|-----|---|-------------------|
| 319 | id | ID |
| 320 | This opaque attribute identifies this object-type in the repository. | |
| 321 | | |
| 322 | localName | String (optional) |
| 323 | This attribute represents the underlying repository's name for the object-type. This field is | |
| 324 | opaque and has no uniqueness constraint imposed by this specification. | |
| 325 | Two properties with the same localName and localNamespace MUST have the same semantic | |
| 326 | equality. | |
| 327 | | |
| 328 | localNamespace | String (optional) |
| 329 | This attribute allows repositories to represent the internal namespace of the underlying | |
| 330 | repository's name for the object-type. | |

331

332 **queryName** String

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

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

335

336 The string MUST NOT contain:

337

- whitespace “ ”,
- 338 • comma “,”
- 339 • double quotes “”
- 340 • single quotes “”
- 341 • backslash “\”
- 342 • the period “.” character or,
- 343 • the open “(“ or close “)” parenthesis characters.

344

345 **displayName** String (optional)

346 Used for presentation by application.

347

348 **baseId** Enum

349 A value that indicates whether the base type for this Object-Type is the Document, Folder,

350 Relationship, or Policy base type.

351

352 **parentId** ID

353 The ID of the Object-Type’s immediate parent type.

354 It MUST be “not set” for a base type.

355

356 **description** String (optional)

357 Description of this object-type, such as the nature of content, or its intended use. Used for

358 presentation by application.

359

360 **creatable** Boolean

361 Indicates whether new objects of this type MAY be created. If the value of this attribute is FALSE,

362 the repository MAY contain objects of this type already, but MUST NOT allow new objects of this

363 type to be created.

364

365 **fileable** Boolean

366 Indicates whether or not objects of this type are [file-able](#).

367

368 **queryable** Boolean

369 Indicates whether or not this object-type can appear in the FROM clause of a query statement. A

370 non-queryable object-type is not visible through the relational view that is used for query, and

371 CAN NOT appear in the FROM clause of a query statement.

372

373 **controllablePolicy** Boolean

374 Indicates whether or not objects of this type are controllable via policies. Policy objects can only
 375 be applied to controllablePolicy objects.

376

377 **controllableACL** Boolean

378 This attribute indicates whether or not objects of this type are controllable by ACL's. Only objects
 379 that are controllableACL can have an ACL.

380

381 **fulltextIndexed** Boolean

382 Indicates whether objects of this type are indexed for full-text search for querying via the
 383 CONTAINS() query predicate.

384

385 **includedInSupertypeQuery** Boolean

386 Indicates whether this type and its subtypes appear in a query of this type's ancestor types.

387 For example: if Invoice is a sub-type of cmis:document, if this is TRUE on Invoice then for a query
 388 on cmis:document, instances of Invoice will be returned if they match.

389 If this attribute is FALSE, no instances of Invoice will be returned even if they match the query.

390 **2.1.3.3 Object-Type Property Definitions**

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

395 **2.1.3.3.1 Property Types**

396 Property types are defined in section 2.1.2.1 Property.

397 **2.1.3.3.2 Attributes common to ALL Object-Type Property Definitions**

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

399 **id** ID

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

402

403 **localName** String (optional)

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

406

407 **localNamespace** String (optional)

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

410

411 **queryName** String

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

414

415 **displayName** String (optional)
416 Used for presentation by application.
417
418 **description** String (optional)
419 This is an optional attribute containing a description of the property
420
421 **propertyType** Enum
422 This attribute indicates the type of this property. It MUST be one of the allowed property types.
423 (See section 2.1.2.1 Property.)
424
425 **cardinality** Enum
426 Indicates whether the property can have “zero or one” or “zero or more” values.
427 Values:
428

- **single**: Property can have zero or one values (if property is not required), or exactly one value (if property is required)

429

- **multi**: Property can have zero or more values (if property is not required), or one or more values (if property is required).

430 Repositories SHOULD preserve the ordering of values in a multi-valued property. That is, the
431 order in which the values of a multi-valued property are returned in get operations SHOULD be
432 the same as the order in which they were supplied during previous create/update operation.
433
434
435
436 **updatability** Enum
437 Indicates under what circumstances the value of this property MAY be updated.
438 Values:
439

- **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.
 - 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.)

440

- **readwrite**: The property value can be modified using the *updateProperties* service.

441

- **whencheckedout**: The property value MUST only be update-able using a “private working copy” Document.
 - I.e. the update is either made on a “private working copy” object or made using a “check in” service.

442

- **oncreate**: The property value MUST only be update-able during the Create operation on that Object.

443
444
445
446
447
448
449
450
451
452
453 **inherited** Boolean
454 Indicates whether the property definition is inherited from the parent-type when TRUE or it is explicitly defined for this object-type when FALSE.
455
456
457 **required** Boolean
458

This attribute is only applicable to non-system properties, i.e. properties whose value is provided by the application.

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.

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.

This attribute is not applicable when the “updatability” attribute is “readonly”. In that case, “required” SHOULD be set to FALSE.

Note: For CMIS-defined object types, the value of a system property (such as cmis:objectId, cmis:createdBy) MUST be set by the repository. However, the property’s “required” attribute SHOULD be FALSE because it is read-only to applications.

queryable Boolean

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

This attribute MUST have a value of FALSE if the Object-type’s attribute for “Queryable” is set to FALSE.

orderable Boolean

Indicates whether the property can appear in the ORDER BY clause of a CMIS query statement or an ORDERBY parameter.

This property MUST be FALSE for any property whose cardinality is “multi”.

choices <PropertyChoiceType list> (multi-valued)

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

If the cardinality of the property definition is “single” and the “openChoice” attribute is FALSE, then the property value MUST be at most one of the values listed in this attribute.

If the cardinality of the property definition is “single” and the “openChoice” attribute is TRUE, then the property value MAY be one of the values listed in this attribute.

If the cardinality of the property definition is “multi” and the “openChoice” attribute is FALSE, then the property value MUST be zero, one or more than one of the values listed in this attribute.

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

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

Choices MAY be hierarchically presented. For example: a value of “choices” for a geographic location would be represented as follows:

- Europe:
 - England
 - France
 - Germany
- North America
 - Canada
 - USA

506 ▪ Mexico

507 **openChoice** Boolean

508 This attribute is only applicable to properties that provide a value for the “Choices” attribute.

509 If FALSE, then the data value for the property MUST only be one of the values specified in the

510 “Choices” attribute. If TRUE, then values other than those included in the “Choices” attribute may

511 be set for the property.

512

513 **defaultValue** <PropertyType>

514 The value that the repository MUST set for the property if a value is not provided by an

515 application when the object is created.

516 If no default value is specified and an application creates an object of this type without setting a

517 value for the property, the repository MUST attempt to store a “value not set” state for the

518 property value. If this occurs for a property that is defined to be required, then the creation

519 attempt MUST throw an exception.

520 The attributes on the default value element are the same as the attributes on the property

521 definition.

522 2.1.3.3.3 Attributes specific to Integer Object-Type Property Definitions

523 The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is

524 “Integer”, in addition to the common attributes specified above. A repository MAY provide additional

525 guidance on what values can be accepted. If the following attributes are not present the repository

526 behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

527 **minValue** Integer

528 The minimum value allowed for this property.

529 If an application tries to set the value of this property to a value lower than **minValue**, the

530 repository MUST throw a **constraint** exception.

531

532 **maxValue** Integer

533 The maximum value allowed for this property.

534 If an application tries to set the value of this property to a value higher than **maxValue**, the

535 repository MUST throw a **constraint** exception.

536

537 2.1.3.3.4 Attributes specific to DateTime Object-Type Property Definitions

538 The following Object **attributes** MUST only apply to Property-Type definitions whose *propertyType* is

539 “DateTime”, in addition to the common attributes specified above. A repository MAY provide additional

540 guidance on what values can be accepted. If the following attributes are not present the repository

541 behavior is undefined and it MAY throw an exception if a runtime constraint is encountered.

542 **resolution** String Enumeration

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

544

- Year: Year resolution is persisted
- Date: Date resolution is persisted
- Time: Time resolution is persisted

546

547

2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions

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

precision Integer Enumeration

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

- 32: 32-bit precision (“single” as specified in IEEE-754-1985).
- 64: 64-bit precision (“double” as specified in IEEE-754-1985.)

minValue Decimal

The minimum value allowed for this property.

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

maxValue Decimal

The maximum value allowed for this property.

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

2.1.3.3.6 Attributes specific to String Object-Type Property Definitions

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

maxLength Integer

The maximum length (in characters) allowed for a value of this property.

If an application attempts to set the value of this property to a string larger than the specified maximum length, the repository MUST throw a **constraint** exception.

2.1.4 Document Object

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

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

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

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

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

588 2.1.4.1 Content Stream

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

593 CMIS provides basic CRUD services for content-stream, using the ID of a content-stream's containing
594 document object for identification. A content stream also has a `streamId` which is used for access to the
595 stream. The "Set Content-Stream" service (`setContentStream`) either creates a new content-stream for a
596 document object or replaces an existing content-stream. The "Get Content-Stream" service
597 (`getContentStream`) retrieves a content-stream. The "Delete Content-Stream" service
598 (`deleteContentStream`) deletes a content-stream from a document object. In addition, the
599 "CreateDocument" and "Check-in" services MAY also take a content-stream as an optional input. A
600 content stream MUST be specified if required by the type definition. These are the only services that
601 operate on content-stream. The "Get Properties" and "Query" services, for example, do not return a
602 content-stream.

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

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

608 2.1.4.2 Renditions

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

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

621 2.1.4.2.1 Rendition Attributes

622 A rendition has the following attributes:

623	streamId	ID
624	Identifies the rendition stream.	
625		
626	contentType	String
627	The MIME type of the rendition stream.	
628		
629	length	Integer (optional)
630	The length of the rendition stream in bytes.	

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

2.1.4.3.2 Attribute Values

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

Notes:

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

id

Value: cmis:document

localName

Value: <repository-specific>

localNamespace

Value: <repository-specific>

queryName

Value: cmis:document

displayName

Value: <repository-specific>

baseId

Value: cmis:document

parentId

Value: Not set

description

Value: <repository-specific>

creatable

Value: <repository-specific>

fileable

Value: TRUE

queryable

714 Value: SHOULD be TRUE
715
716 **controllablePolicy**
717 Value: <repository-specific>
718
719 **includedInSupertypeQuery**
720 Value: <repository-specific>
721
722 **versionable**
723 Value: <repository-specific>
724
725 **contentStreamAllowed**
726 Value: <repository-specific>
727
728 **controllableACL**
729 Value: <repository-specific>
730
731 **fulltextIndexed**
732 Value: <repository-specific>

733 2.1.4.3.3 Property Definitions

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

738		
739	cmis:name	Name of the object
740	Inherited:	False
741	Property Type:	String
742	Cardinality:	Single
743		
744	cmis:objectId	Id of the object
745	Required:	False
746	Inherited:	False
747	Property Type:	ID
748	Cardinality:	Single
749	Updatability:	Read Only
750	Choices:	Not Applicable
751	Open Choice:	Not Applicable
752	Repository MUST return this property with non-empty values when an object is requested and the	
753	property filter does not exclude them	
754		
755		
756	cmis:baseTypeId	Id of the base object-type for the object

757	Required:	False
758	Inherited:	False
759	Property Type:	ID
760	Cardinality:	Single
761	Updatability:	Read Only
762	Choices:	Not Applicable
763	Open Choice:	Not Applicable
764	Repository MUST return this property with non-empty values when an object is requested and the	
765	property filter does not exclude them	
766		
767	cmis:objectTypeId	Id of the object's type
768	Required:	True
769	Inherited:	False
770	Property Type:	ID
771	Cardinality:	Single
772	Updatability:	oncreate
773	Choices:	Not Applicable
774	Open Choice:	Not Applicable
775	Repository MUST return this property with non-empty values when an object is requested and the	
776	property filter does not exclude them	
777		
778	cmis:createdBy	User who created the object.
779	Required:	False
780	Inherited:	False
781	Property Type:	String
782	Cardinality:	Single
783	Updatability:	Read Only
784	Choices:	Not Applicable
785	Open Choice:	Not Applicable
786	Queryable:	True
787	Orderable:	True
788	Repository MUST return this property with non-empty values when an object is requested and the	
789	property filter does not exclude them	
790		
791	cmis:creationDate	DateTime when the object was created.
792	Required:	False
793	Inherited:	False
794	Property Type:	DateTime
795	Cardinality:	Single
796	Updatability:	Read Only
797	Choices:	Not Applicable
798	Open Choice:	Not Applicable
799	Queryable:	True

800	Orderable:	True
801	Repository MUST return this property with non-empty values when an object is requested and the	
802	property filter does not exclude them	
803		
804	cmis:lastModifiedBy	User who last modified the object.
805	Required:	False
806	Inherited:	False
807	Property Type:	String
808	Cardinality:	Single
809	Updatability:	Read Only
810	Choices:	Not Applicable
811	Open Choice:	Not Applicable
812	Queryable:	True
813	Orderable:	True
814	Repository MUST return this property with non-empty values when an object is requested and the	
815	property filter does not exclude them	
816		
817	cmis:lastModificationDate	DateTime when the object was last modified.
818	Required:	False
819	Inherited:	False
820	Property Type:	DateTime
821	Cardinality:	Single
822	Updatability:	Read Only
823	Choices:	Not Applicable
824	Open Choice:	Not Applicable
825	Queryable:	True
826	Orderable:	True
827	Repository MUST return this property with non-empty values when an object is requested and the	
828	property filter does not exclude them	
829		
830	cmis:changeToken	Opaque token used for optimistic locking & concurrency
831	checking. (see section 2.2.1.3 Change Tokens)	
832	Required:	False
833	Inherited:	False
834	Property Type:	String
835	Cardinality:	Single
836	Updatability:	Read Only
837	Choices:	Not Applicable
838	Open Choice:	Not Applicable
839	Repository MUST return this property with non-empty values when an object is requested and the	
840	property filter does not exclude them	
841		
842	cmis:isImmutable	TRUE if the repository MUST throw an error at any attempt to
843	update or delete the object.	

844	Required:	False
845	Inherited:	False
846	Property Type:	Boolean
847	Cardinality:	Single
848	Updatability:	Read Only
849	Choices:	Not Applicable
850	Open Choice:	Not Applicable
851	Repository MUST return this property with non-empty values when an object is requested and the	
852	property filter does not exclude them	
853		
854	cmis:isLatestVersion	See section 2.1.9 Versioning.
855	Required:	False
856	Inherited:	False
857	Property Type:	Boolean
858	Cardinality:	Single
859	Updatability:	Read Only
860	Choices:	Not Applicable
861	Open Choice:	Not Applicable
862	Repository MUST return this property with non-empty values when an object is requested and the	
863	property filter does not exclude them. Version Property Values are repository-specific when a	
864	document is defined as non-versionable.	
865		
866	cmis:isMajorVersion	See section 2.1.9 Versioning.
867	Required:	False
868	Inherited:	False
869	Property Type:	Boolean
870	Cardinality:	Single
871	Updatability:	Read Only
872	Choices:	Not Applicable
873	Open Choice:	Not Applicable
874	Repository MUST return this property with non-empty values when an object is requested and the	
875	property filter does not exclude them. Version Property Values are repository-specific when a	
876	document is defined as non-versionable.	
877		
878	cmis:isLatestMajorVersion	See section 2.1.9 Versioning.
879	Required:	False
880	Inherited:	False
881	Property Type:	Boolean
882	Cardinality:	Single
883	Updatability:	Read Only
884	Choices:	Not Applicable
885	Open Choice:	Not Applicable

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

889

890 **cmis:versionLabel** See section 2.1.9 Versioning.

891 Required: False

892 Inherited: False

893 Property Type: String

894 Updatability: Read Only

895 Choices: Not Applicable

896 Open Choice: Not Applicable

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

900

901 **cmis:versionSeriesId** See section 2.1.9 Versioning.

902 Required: False

903 Inherited: False

904 Property Type: ID

905 Cardinality: Single

906 Updatability: Read Only

907 Choices: Not Applicable

908 Open Choice: Not Applicable

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

912

913 **cmis:isVersionSeriesCheckedOut** See section 2.1.9 Versioning.

914 Required: False

915 Inherited: False

916 Property Type: Boolean

917 Cardinality: Single

918 Updatability: Read Only

919 Choices: Not Applicable

920 Open Choice: Not Applicable

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

924

925 **cmis:versionSeriesCheckedOutBy** See section 2.1.9 Versioning.

926 Required: False

927 Inherited: False

928 Property Type: String

929 Cardinality: Single

930	Updatability:	Read Only
931	Choices:	Not Applicable
932	Open Choice:	Not Applicable
933	Version Property Values are repository-specific when a document is defined as non-versionable.	
934		
935	cmis:versionSeriesCheckedOutId	See section 2.1.9 Versioning.
936	Required:	False
937	Inherited:	False
938	Property Type:	ID
939	Cardinality:	Single
940	Updatability:	Read Only
941	Choices:	Not Applicable
942	Open Choice:	Not Applicable
943	Version Property Values are repository-specific when a document is defined as non-versionable.	
944		
945	cmis:checkinComment	See section 2.1.9 Versioning.
946	Required:	False
947	Inherited:	False
948	Property Type:	String
949	Cardinality:	Single
950	Updatability:	Read Only
951	Choices:	Not Applicable
952	Open Choice:	Not Applicable
953	Version Property Values are repository-specific when a document is defined as non-versionable.	
954		
955	cmis:contentStreamLength	Length of the content stream (in bytes).
956	Required:	False
957	Inherited:	False
958	Property Type:	Integer
959	Cardinality:	Single
960	Updatability:	Read Only
961	Choices:	Not Applicable
962	Open Choice:	Not Applicable
963	Repository MUST return this property with non-empty values when an object is requested and the	
964	property filter does not exclude them and if the document has a content stream	
965		
966	cmis:contentStreamMimeType	MIME type of the Content Stream
967	Required:	False
968	Inherited:	False
969	Property Type:	String
970	Cardinality:	Single
971	Updatability:	Read Only

972	Choices:	Not Applicable
973	Open Choice:	Not Applicable
974	Repository MUST return this property with non-empty values when an object is requested and the	
975	property filter does not exclude them and if the document has a content stream	
976		
977	cmis:contentStreamFileName	File name of the Content Stream
978	Required:	False
979	Inherited:	False
980	Property Type:	String
981	Cardinality:	Single
982	Repository MUST return this property with non-empty values when an object is requested and the	
983	property filter does not exclude them and if the document has a content stream	
984		
985	cmis:contentStreamId	Id of the stream
986	Required:	False
987	Inherited:	False
988	Property Type:	ID
989	Cardinality:	Single
990	Updatability:	Read Only
991	Choices:	Not Applicable
992	Open Choice:	Not Applicable

993 2.1.5 Folder Object

994 A folder object serves as the anchor for a collection of *file-able* objects. The folder object has an *implicit*
995 hierarchical relationship with each object in its collection, with the anchor folder object being the **Parent**
996 object and each object in the collection being a **Child** object. This implicit relationship has specific
997 containment semantics which MUST be maintained by the repository with implicit referential integrity.
998 (That is, there will never be a dangling parent-relationship or a dangling child-relationship. Furthermore,
999 object A is a parent of object B if and only if object B is a child of object A.) This system-maintained
1000 implicit relationship is distinct from an *explicit* relationship which is instantiated by an application-
1001 maintained Relationship Object. (See section 2.1.6 Relationship Object.)

1002 A folder object does not have a content-stream and is not version-able. A folder object MAY be
1003 associated with zero or more renditions (see section 2.1.4.2 Renditions).

1004 2.1.5.1 File-able Objects

1005 A *file-able* object is one that MAY be “filed” into a folder. That is, it MAY be a child object of a folder
1006 object. The following list defines whether the base CMIS Object-types are file-able:

1007	cmis:folder
1008	MUST be file-able
1009	
1010	cmis:document
1011	MUST be file-able
1012	
1013	cmis:relationship

1014 MUST NOT be file-able

1015

1016 **cmis:policy**

1017 MAY be file-able

1018 2.1.5.1.1 Document Version Series and Filing

1019 Since document objects are versionable, a document object's membership in a folder MAY be version-
1020 specific or version-independent. That is, the folder membership MAY be restricted to that particular
1021 version of the document or MAY apply to all versions of the document. Whether or not a repository
1022 supports version-specific filing is discoverable via the "Get Repository Information" service
1023 (*getRepositoryInfo*).

1024 When the child objects of a folder are retrieved, a specific version of a document MAY be returned. If the
1025 repository supports version-specific filing, the specific version filed in that folder is returned. If the
1026 repository does not support version-specific filing, the latest version of the document is returned.

1027 Likewise, this version sensitivity in child-binding also affects the behavior of parent retrieval for a
1028 document object, as well as the scope of the IN_FOLDER() and IN_TREE() function calls in a query. For
1029 non-versionable fileable objects, their membership in a folder does not have version sensitivity.

1030 2.1.5.1.2 Filing Restrictions by Object-Type

1031 A folder collection's membership MAY be restricted by object-type. Each folder object has a multi-valued
1032 *AllowedChildObjectTypes* property, which specifies that only objects of these types are allowed to be
1033 its children. If this property is "not set", then objects of any file-able type MAY be filed in the Folder. It is
1034 repository-specific if subtypes of the types listed in the *AllowedChildObjectTypes* property MAY be filed
1035 in the folder.

1036 Because of these filing constraints, when a new folder object is created, an existing folder object MUST
1037 be specified as its parent.

1038 When a non-file-able object is created, a parent folder MUST NOT be specified.

1039 When a file-able object is deleted, it is removed from any folder collection in which the object is a
1040 member. In other words, when an object is deleted, all implicit parent-child relationships with the deleted
1041 object as a child cease to exist.

1042 2.1.5.2 Folder Hierarchy

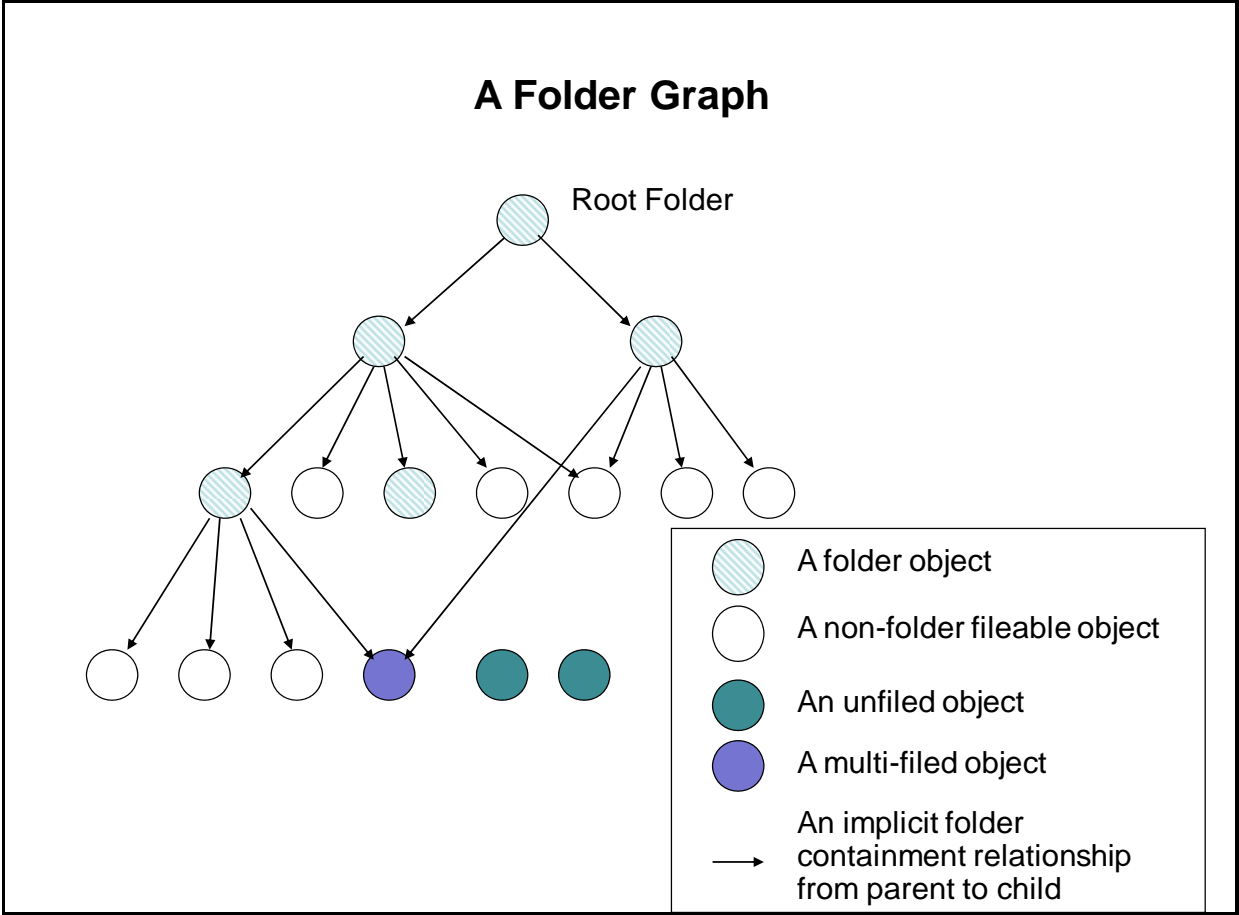
1043 CMIS imposes the following constraints on folder objects:

- 1044 • Every folder object, except for one which is called the **Root Folder**, MUST have one and only
1045 one parent folder. The Root Folder does not have a parent.
- 1046 • A cycle in folder containment relationships is not allowed. That is, a folder object cannot have
1047 itself as one of its descendant objects.
- 1048 • A child object that is a folder object can itself be the parent object of other file-able objects.

1049 With these constraints, the folder objects in a CMIS repository necessarily form a strict hierarchy, with the
1050 Root Folder being the root of the hierarchy.

1051 The child objects of a given folder object, their child objects, and grandchild objects, etc., are called
1052 **Descendant** objects of the given folder object. A folder object together with all its descendant objects are
1053 collectively called a **Tree** rooted at that folder object.

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



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

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

2.1.5.3 Paths

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

- '/' for the root folder
- All paths start with the root folder.
- A set of the folder and object path segments separated by '/' in order of closest to the root.
- Folder and object path segments are specified by pathSegment tokens which can be retrieved by all services that take an `includePathSegments` parameter.
- A pathSegment token MUST not include a '/' character.

1073 ○ It is repository specific how a repository chooses the value for pathSegment.
 1074 Repositories might choose to use cmis:name or content stream filename for
 1075 pathSegment token.

1076 • The pathSegment token for each item MUST uniquely identify the item in the folder.

1077

1078 E.g., if folder A is under the root, and folder B is under A, then the path would be /A/B.

1079 A path for an object may be calculated by taking the item's parent folder cmis:path property and
 1080 appending the "/" character and the object's pathSegment. This constructed path may be given as input
 1081 to the *getObjectByPath* service for object by path retrieval.

1082 The *getObjectParents* service returns *relativePathSegment* tokens. These tokens are the
 1083 pathSegment of the input object relative to the parent folders.

1084 **2.1.5.4 Folder Object-Type Definition**

1085 This section describes the definition of the Folder Object-Type's attribute values and property definitions
 1086 which must be present on Folder instance objects. All attributes and property definitions are listed by
 1087 their ID.

1088 **2.1.5.4.1 Attribute Values**

1089 The Folder Object-Type MUST have the following attribute values.

1090 Notes:

- 1091 • A value of <repository-specific> indicates that the value of the property MAY be set to any valid
 1092 value for the attribute type.
- 1093 • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the
 1094 Object-Type definition.

1095	
1096	id
1097	Value: cmis:folder
1098	
1099	localName
1100	Value: <repository-specific>
1101	
1102	localNamespace
1103	Value: <repository-specific>
1104	
1105	queryName
1106	Value: cmis:folder
1107	
1108	displayName
1109	Value: <repository-specific>
1110	
1111	baseId
1112	Value: cmis:folder
1113	
1114	parentId
1115	Value: Not set

1116
 1117 **description**
 1118 Value: <repository-specific>
 1119
 1120 **creatable**
 1121 Value: <repository-specific>
 1122
 1123 **fileable**
 1124 Value: TRUE
 1125
 1126 **queryable**
 1127 Value: SHOULD be TRUE
 1128
 1129 **controllablePolicy**
 1130 Value: <repository-specific>
 1131
 1132 **includedInSupertypeQuery**
 1133 Value: <repository-specific>
 1134
 1135 **controllableACL**
 1136 Value: <repository-specific>
 1137
 1138 **fulltextIndexed**
 1139 Value: <repository-specific>
 1140

1141 2.1.5.4.2 Property Definitions

1142 The Folder base Object-Type MUST have the following property definitions, and MAY include additional
 1143 property definitions. Any attributes not specified for the Property Definition are repository specific. For all
 1144 property definitions on base types, the query name MUST be the same as the property ID. The
 1145 repository MUST have the following property definitions on the Folder Type:

1146	cmis:name	Name of the object
1147	Inherited:	False
1148	Property Type:	String
1149	Cardinality:	Single
1150	Required:	True
1151		
1152	cmis:objectId	Id of the object
1153	Required:	False
1154	Inherited:	False
1155	Property Type:	ID
1156	Cardinality:	Single
1157	Updatability:	Read Only
1158	Choices:	Not Applicable

1159	Open Choice:	Not Applicable
1160	Repository MUST return this property with non-empty values when an object is requested and the	
1161	property filter does not exclude them	
1162		
1163	cmis:baseTypeId	Id of the base object-type for the object
1164	Required:	False
1165	Inherited:	False
1166	Property Type:	ID
1167	Cardinality:	Single
1168	Updatability:	Read Only
1169	Choices:	Not Applicable
1170	Open Choice:	Not Applicable
1171	Repository MUST return this property with non-empty values when an object is requested and the	
1172	property filter does not exclude them	
1173		
1174	cmis:objectType	Id of the object's type
1175	Required:	False
1176	Inherited:	False
1177	Property Type:	ID
1178	Cardinality:	Single
1179	Updatability:	oncreate
1180	Choices:	Not Applicable
1181	Open Choice:	Not Applicable
1182	Repository MUST return this property with non-empty values when an object is requested and the	
1183	property filter does not exclude them	
1184		
1185	cmis:createdBy	User who created the object.
1186	Required:	False
1187	Inherited:	False
1188	Property Type:	String
1189	Cardinality:	Single
1190	Updatability:	Read Only
1191	Choices:	Not Applicable
1192	Open Choice:	Not Applicable
1193	Queryable:	True
1194	Orderable:	True
1195	Repository MUST return this property with non-empty values when an object is requested and the	
1196	property filter does not exclude them	
1197		
1198	cmis:creationDate	DateTime when the object was created.
1199	Required:	False
1200	Inherited:	False
1201	Property Type:	DateTime

1202	Cardinality:	Single
1203	Updatability:	Read Only
1204	Choices:	Not Applicable
1205	Open Choice:	Not Applicable
1206	Queryable:	True
1207	Orderable:	True
1208	Repository MUST return this property with non-empty values when an object is requested and the	
1209	property filter does not exclude them	
1210		
1211	cmis:lastModifiedBy	User who last modified the object.
1212	Required:	False
1213	Inherited:	False
1214	Property Type:	String
1215	Cardinality:	Single
1216	Updatability:	Read Only
1217	Choices:	Not Applicable
1218	Open Choice:	Not Applicable
1219	Queryable:	True
1220	Orderable:	True
1221	Repository MUST return this property with non-empty values when an object is requested and the	
1222	property filter does not exclude them	
1223		
1224	cmis:lastModificationDate	DateTime when the object was last modified.
1225	Required:	False
1226	Inherited:	False
1227	Property Type:	DateTime
1228	Cardinality:	Single
1229	Updatability:	Read Only
1230	Choices:	Not Applicable
1231	Open Choice:	Not Applicable
1232	Queryable:	True
1233	Orderable:	True
1234	MUST be set on the object	
1235		
1236	cmis:changeToken	Token used for optimistic locking & concurrency checking.
1237	(see section 2.2.1.3 Change Tokens)	
1238	Required:	False
1239	Inherited:	False
1240	Property Type:	String
1241	Cardinality:	Single
1242	Updatability:	Read Only
1243	Choices:	Not Applicable
1244	Open Choice:	Not Applicable

1245	Repository MUST return this property with non-empty values when an object is requested and the	
1246	property filter does not exclude them	
1247		
1248	cmis:parentId	ID of the parent folder of the folder.
1249	Required:	False
1250	Inherited:	False
1251	Property Type:	ID
1252	Cardinality:	Single
1253	Updatability:	Read Only
1254	Choices:	Not Applicable
1255	Open Choice:	Not Applicable
1256	Repository MUST return this property with non-empty values when an object is requested and the	
1257	property filter does not exclude them	
1258		
1259	cmis:path	The fully qualified path to this folder. See section 2.1.5.3
1260		Paths.
1261	Required:	False
1262	Inherited:	False
1263	Property Type:	String
1264	Cardinality:	Single
1265	Updatability:	Read Only
1266	Choices:	Not Applicable
1267	Open Choice:	Not Applicable
1268	Repository MUST return this property with non-empty values when an object is requested and the	
1269	property filter does not exclude them	
1270		
1271	cmis:allowedChildObjectTypes	Id's of the set of Object-types that can be created, moved or
1272		filed into this folder.
1273	Required:	False
1274	Inherited:	False
1275	Property Type:	ID
1276	Cardinality:	Multi
1277	Updatability:	Read Only
1278	Choices:	Not Applicable
1279	Open Choice:	Not Applicable

1280 2.1.6 Relationship Object

1281 A relationship object is semantically a *dependent* object. A relationship object MUST NOT have a
1282 content-stream, and MUST NOT be versionable, MAY be queryable, and MUST NOT be fileable,
1283 although it MAY be controllable.

1284 If a repository does not support relationship objects, the relationship base object-type SHOULD NOT be
1285 returned by a "Get Types" service call.

1286 A **Relationship Object** instantiates an explicit, binary, directional, non-invasive, and typed relationship
1287 between a **Source Object** and a **Target Object**. The source object and the target object MUST both be
1288 independent objects, such as a document object, a folder object, or a policy object. Whether a policy
1289 object is allowed to be the source or target object of a relationship object is repository-specific.

1290 The relationship instantiated by a relationship object is *explicit* since it is explicitly represented by an
1291 object and is explicitly managed by application.

1292 This relationship is *non-invasive* in the sense that creating or removing this relationship SHOULD NOT
1293 modify either the source or the target object. That is, it SHOULD NOT require an update capability (or
1294 permission) on either object; SHOULD NOT affect the versioning state of either object; and SHOULD
1295 NOT change their “Last Modification Date”.

1296 Explicit relationships can be used to create an arbitrary relationship graph among independent objects.
1297 Such a relationship graph is only structural in nature. No inheritance or transitive properties are attached
1298 to a relationship graph.

1299 The notion of a source object and a target object of a relationship is used solely to indicate the direction of
1300 the relationship. No semantics or implementation bias is implied by this terminology.

1301 The binding of a relationship object to a source document object or to a target document object MAY be
1302 either version-specific or version-independent. This version sensitivity is repository-specific, and is largely
1303 transparent to CMIS. An independent object MAY participate in any number of explicit relationships, as
1304 the source object for some and as the target object for others. Multiple relationships MAY exist between
1305 the same pair of source and target objects.

1306 Referential integrity, either between the source object and the target object, or between the relationship
1307 object and the source or target object, is repository-specific. Therefore, creating an explicit relationship
1308 between two objects MAY impose a constraint on any of the three objects, and removing a relationship or
1309 deleting either the source or the target object MAY be restricted by such a constraint. If the source or the
1310 target object of a relationship is deleted, the repository MAY automatically delete the relationship object.

1311 Like all CMIS objects, relationship objects are typed. Typing relationship allows them to be grouped,
1312 identified, and traversed by type id, and for properties to be defined for individual relationship types.

1313 Additionally, a relationship object-type MAY specify that only Objects of a specific Object-Type can
1314 participate as the source object or target object for relationship objects of that type. If no such constraints
1315 are specified, then an independent object of any type MAY be the source or the target of a relationship
1316 object of that type.

1317 When a relationship object is created, the source object ID and the target object ID MUST reference valid
1318 non-relationship CMIS objects.

1319 When a relationship object is retrieved, its source object or target object MAY no longer exist, since
1320 referential integrity MAY not be maintained by a repository.

1321 In addition to object CRUD services, a “Get Relationships” service (*getObjectRelationships*) may be used
1322 to return a set of relationship objects in which a given independent object is identified as the source or the
1323 target object, according to the binding semantics maintained by the repository (i.e., either a version-
1324 specific or a version-independent binding as described above).

1325 2.1.6.1 Relationship Object-Type Definition

1326 This section describes the definition of the Relationship Object-Type’s attribute values and property
1327 definitions which must be present on Relationship instance objects. All attributes and property definitions
1328 are listed by their ID.

2.1.6.1.1 Attributes specific to Relationship Object-Types

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

allowedSourceTypes ID (multi-valued)

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

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

allowedTargetTypes ID (multi-valued)

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

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

2.1.6.1.2 Attribute Values

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

Notes:

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

id

Value: cmis:relationship

localName

Value: <repository-specific>

localNamespace

Value: <repository-specific>

queryName

Value: cmis:relationship

displayName

Value: <repository-specific>

baseId

Value: cmis:relationship

parentId

Value: Not set

description

Value: <repository-specific>

1372
1373 **creatable**
1374 Value: <repository-specific>
1375
1376 **fileable**
1377 Value: FALSE
1378
1379 **queryable**
1380 Value: <repository-specific>
1381
1382 **includedInSupertypeQuery**
1383 Value: <repository-specific>
1384
1385 **controllablePolicy**
1386 Value: <repository-specific>
1387
1388 **allowedSourceTypes**
1389 Value: <repository-specific>
1390
1391 **allowedTargetTypes**
1392 Value: <repository-specific>
1393
1394 **controllableACL**
1395 Value: <repository-specific>
1396
1397 **fulltextIndexed**
1398 Value: <repository-specific>
1399

1400 **2.1.6.1.3 Property Definitions**

1401 The Relationship base Object-Type MUST have the following property definitions, and MAY include
1402 additional property definitions. Any attributes not specified by the Property Definitions are repository
1403 specific. For all property definitions on base types, the query name MUST be the same as the property
1404 ID. The repository MUST have the following property definitions on the Relationship Type:

1405		
1406	cmis:name	Name of the object
1407	Inherited:	False
1408	Property Type:	String
1409	Cardinality:	Single
1410		
1411	cmis:objectId	Id of the object
1412	Required:	False
1413	Inherited:	False
1414	Property Type:	ID

1415	Cardinality:	Single
1416	Updatability:	Read Only
1417	Choices:	Not Applicable
1418	Open Choice:	Not Applicable
1419	Repository MUST return this property with non-empty values when an object is requested and the	
1420	property filter does not exclude them	
1421		
1422	cmis:baseTypeId	Id of the base object-type for the object
1423	Required:	False
1424	Inherited:	False
1425	Property Type:	ID
1426	Cardinality:	Single
1427	Updatability:	Read Only
1428	Choices:	Not Applicable
1429	Open Choice:	Not Applicable
1430	Repository MUST return this property with non-empty values when an object is requested and the	
1431	property filter does not exclude them	
1432		
1433	cmis:objectTypeId	Id of the object's type
1434	Required:	False
1435	Inherited:	False
1436	Property Type:	ID
1437	Cardinality:	Single
1438	Updatability:	oncreate
1439	Choices:	Not Applicable
1440	Open Choice:	Not Applicable
1441	Repository MUST return this property with non-empty values when an object is requested and the	
1442	property filter does not exclude them	
1443		
1444	cmis:createdBy	User who created the object.
1445	Required:	False
1446	Inherited:	False
1447	Property Type:	String
1448	Cardinality:	Single
1449	Updatability:	Read Only
1450	Choices:	Not Applicable
1451	Open Choice:	Not Applicable
1452	Repository MUST return this property with non-empty values when an object is requested and the	
1453	property filter does not exclude them	
1454		
1455	cmis:creationDate	DateTime when the object was created.
1456	Required:	False
1457	Inherited:	False

1458	Property Type:	DateTime
1459	Cardinality:	Single
1460	Updatability:	Read Only
1461	Choices:	Not Applicable
1462	Open Choice:	Not Applicable
1463	Repository MUST return this property with non-empty values when an object is requested and the	
1464	property filter does not exclude them	
1465		
1466	cmis:lastModifiedBy	User who last modified the object.
1467	Required:	False
1468	Inherited:	False
1469	Property Type:	String
1470	Cardinality:	Single
1471	Updatability:	Read Only
1472	Choices:	Not Applicable
1473	Open Choice:	Not Applicable
1474	Repository MUST return this property with non-empty values when an object is requested and the	
1475	property filter does not exclude them	
1476		
1477	cmis:lastModificationDate	DateTime when the object was last modified.
1478	Required:	False
1479	Inherited:	False
1480	Property Type:	DateTime
1481	Cardinality:	Single
1482	Updatability:	Read Only
1483	Choices:	Not Applicable
1484	Open Choice:	Not Applicable
1485	Repository MUST return this property with non-empty values when an object is requested and the	
1486	property filter does not exclude them	
1487		
1488	cmis:changeToken	Opaque token used for optimistic locking & concurrency
1489	checking. (see section 2.2.1.3 Change Tokens)	
1490	Required:	False
1491	Inherited:	False
1492	Property Type:	String
1493	Cardinality:	Single
1494	Updatability:	Read Only
1495	Choices:	Not Applicable
1496	Open Choice:	Not Applicable
1497		
1498	cmis:sourceld	ID of the source object of the relationship.
1499	Required:	True
1500	Inherited:	False

1501	Property Type:	ID
1502	Cardinality:	Single
1503	Choices:	Not Applicable
1504	Open Choice:	Not Applicable
1505		
1506	cmis:targetId	ID of the target object of the relationship.
1507	Required:	True
1508	Inherited:	False
1509	Property Type:	ID
1510	Cardinality:	Single
1511	Choices:	Not Applicable
1512	Open Choice:	Not Applicable

1513 2.1.7 Policy Object

1514 A policy object represents an administrative policy that can be enforced by a repository, such as a
 1515 retention management policy. CMIS 1.0 does not specify what kinds of administrative policies that are
 1516 specifically supported, nor attempts to model administrative policy of any particular kind. Only a base
 1517 object-type is specified for policy objects. Each policy object holds the text of an administrative policy as a
 1518 repository-specific string, which is opaque to CMIS and which may be used to support policies of various
 1519 kinds. A repository may create subtypes of this base type to support different kinds of administrative
 1520 policies more specifically. If a repository does not support policy objects, the policy base object-type
 1521 SHOULD NOT be returned by a “Get Types” service call. This is an extension point for repositories that
 1522 want to expose other capabilities via CMIS that are not supported directly in CMIS 1.0.

1523 Aside from allowing an application to create and maintain policy objects, CMIS allows an application to
 1524 “apply” a policy to an object, and to remove an applied policy from an object. An object to which a policy
 1525 may be applied is called a *controllable* object. A policy MAY be applied to multiple controllable objects.
 1526 Conversely, a repository MAY allow multiple policies applied to a controllable object. (A repository may,
 1527 for example, impose constraints such as only one policy of each kind can be applied to an object.)
 1528 Whether or not an object is controllable is specified by the object’s type definition. Applying a policy to an
 1529 object is to place the object under the control of that policy (while the object may also be under the control
 1530 of other policies at the same time), and removing an applied policy from one of its controlled objects is to
 1531 remove the corresponding control from that object. This control may change the state of the object, may
 1532 impose certain constraints on service calls operating on this object, or may cause certain management
 1533 actions to take place. The effect of this control, when this effect takes place, and how this control interacts
 1534 with other controls, are repository-specific. Only directly/explicitly applied policies are covered by CMIS
 1535 1.0. Indirectly applying policy to an object, e.g. through inheritance, is outside the scope of CMIS 1.0.

1536 A policy object does not have a content-stream and is not versionable. It may be fileable, queryable or
 1537 controllable. Policy objects are handled using the basic CRUD services for objects. If a policy is updated,
 1538 the change may alter the corresponding control on objects that the policy is currently applied to. If a
 1539 controlled object is deleted, all the policies applied to that object, if there are any, are removed from that
 1540 object. A policy object that is currently applied to one or more controllable objects CAN NOT be deleted.
 1541 That is, there is an implicit referential constraint from a controlled object to its controlling policy object(s).
 1542 Besides the basic CRUD services, the “Apply Policy” (*applyPolicy*) and the “Remove Policy”
 1543 (*removePolicy*) services may be used to apply a policy object to a controllable object and respectively to
 1544 remove an applied policy from one of its controlled objects. In addition, the “Get Applied Policies”
 1545 (*getAppliedPolicies*) service may be used to obtain the policy objects that are currently applied to a
 1546 controllable object.

2.1.7.1 Policy Object-Type Definition

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

2.1.7.1.1 Attribute Values

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

Notes:

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

id

Value: cmis:policy

localName

Value: <repository-specific>

localNamespace

Value: <repository-specific>

queryName

Value: cmis:policy

displayName

Value: <repository-specific>

baseId

Value: cmis:policy

parentId

Value: Not set

description

Value: <repository-specific>

creatable

Value: <repository-specific>

fileable

Value: <repository-specific>

queryable

1590 Value: <repository-specific>
 1591
 1592 **includedInSupertypeQuery**
 1593 Value: <repository-specific>
 1594
 1595 **controllablePolicy**
 1596 Value: <repository-specific>
 1597
 1598 **controllableACL**
 1599 Value: <repository-specific>
 1600
 1601 **fulltextIndexed**
 1602 Value: <repository-specific>
 1603

1604 2.1.7.1.2 Property Definitions

1605 The Policy base Object-Type MUST have the following property definitions, and MAY include additional
 1606 property definitions. Any attributes not specified by the Property Definitions are repository specific. For
 1607 all property definitions on base types, the query name MUST be the same as the property ID. The
 1608 repository MUST have the following property definitions on the Policy Type:

1609		
1610	cmis:name	Name of the object
1611	Inherited:	False
1612	Property Type:	String
1613	Cardinality:	Single
1614		
1615	cmis:objectId	Id of the object
1616	Required:	False
1617	Inherited:	False
1618	Property Type:	ID
1619	Cardinality:	Single
1620	Updatability:	Read Only
1621	Choices:	Not Applicable
1622	Open Choice:	Not Applicable
1623		
1624	cmis:baseTypeId	Id of the base object-type for the object
1625	Required:	False
1626	Inherited:	False
1627	Property Type:	ID
1628	Cardinality:	Single
1629	Updatability:	Read Only
1630	Choices:	Not Applicable
1631	Open Choice:	Not Applicable
1632		

1633	cmis:objectTypeId	Id of the object's type
1634	Required:	False
1635	Inherited:	False
1636	Property Type:	ID
1637	Cardinality:	Single
1638	Updatability:	oncreate
1639	Choices:	Not Applicable
1640	Open Choice:	Not Applicable
1641		
1642	cmis:createdBy	User who created the object.
1643	Required:	False
1644	Inherited:	False
1645	Property Type:	String
1646	Cardinality:	Single
1647	Updatability:	Read Only
1648	Choices:	Not Applicable
1649	Open Choice:	Not Applicable
1650		
1651	cmis:creationDate	DateTime when the object was created.
1652	Required:	False
1653	Inherited:	False
1654	Property Type:	DateTime
1655	Cardinality:	Single
1656	Updatability:	Read Only
1657	Choices:	Not Applicable
1658	Open Choice:	Not Applicable
1659		
1660	cmis:lastModifiedBy	User who last modified the object.
1661	Required:	False
1662	Inherited:	False
1663	Property Type:	String
1664	Cardinality:	Single
1665	Updatability:	Read Only
1666	Choices:	Not Applicable
1667	Open Choice:	Not Applicable
1668		
1669	cmis:lastModificationDate	DateTime when the object was last modified.
1670	Required:	False
1671	Inherited:	False
1672	Property Type:	DateTime
1673	Cardinality:	Single
1674	Updatability:	Read Only

1675	Choices:	Not Applicable
1676	Open Choice:	Not Applicable
1677		
1678	cmis:changeToken	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1679		
1680	Required:	False
1681	Inherited:	False
1682	Property Type:	String
1683	Cardinality:	Single
1684	Updatability:	Read Only
1685	Choices:	Not Applicable
1686	Open Choice:	Not Applicable
1687		
1688	cmis:policyText	User-friendly description of the policy
1689	Required:	True
1690	Inherited:	False
1691	Property Type:	String
1692	Cardinality:	Single
1693	Choices:	Not Applicable
1694	Open Choice:	Not Applicable

1695 2.1.8 Access Control

1696 A repository can support either a base set of CMIS-defined permissions and/or its own set of repository
1697 specific permissions.

1698 The getACL service allows the requestor to specify that the result be expressed using only the CMIS
1699 defined permissions. Without this restriction, the response may include, or be solely expressed in
1700 repository specific permissions. The applyACL service permits either CMIS permissions or repository
1701 permissions, or a combination of both, to be used.

1702 2.1.8.1 ACL, ACE, Principal, and Permission

1703 An **ACL** is a list of **Access Control Entries** (ACEs) and MAY hold zero or more ACEs. If an ACL has no
1704 ACEs, the behavior is the same as if the ACL is not set.

1705 An **ACE** holds:

- 1706 • one **Principal**: A principal represents a user management object, e.g. a user, group, or role.
1707 It holds one **String** with the **principalid**.
- 1708 • One or more **Strings** with the names of the **permissions**.
- 1709 • a **Boolean** flag **direct**, which indicates if TRUE the ACE is directly assigned to the object. If
1710 FALSE, that the ACE is somehow derived.

1711 2.1.8.2 CMIS Permissions

1712 There are three basic permissions predefined by CMIS:

- 1713 • **cmis:read**: to be used to express "permission to read". A Repository SHOULD express the
1714 permission for reading properties AND reading content with this permission.
- 1715 • **cmis:write**: to be used to express "permission to write". SHOULD be used to express permission
1716 to write properties and content of an object. MAY include other basic CMIS permissions.

- 1717 • **cmis:all**: SHOULD be used to express all the permissions of a repository. SHOULD include all
1718 other basic CMIS permissions.

1719 How these basic permissions can be mapped to the allowable actions is repository specific. However, the
1720 actual repository semantics for the basic permissions with regard to allowable actions can be discovered
1721 by the *mappings* parameter returned by *getRepositoryInfo* (see below).

1722 Repositories MAY extend this set with repository-specific permissions.

1723 2.1.8.3 ACL Capabilities

1724 Whether a repository supports ACLs at all, may be discovered via *capabilityACL* returned by
1725 *getRepositoryInfo* (see section 2.1.1.1 Optional Capabilities). If *capabilityACL* is *none*, ACLs are not
1726 supported by the repository.

1727 If *capabilityACL* is *discover* or *manage*, additional information about the repositories permission model
1728 and how changes to ACL are handled, can be discovered via the *getRepositoryInfo* service:

- 1729 • **<Array> Enum propagation**: specifies, how non-direct ACEs can be handled by the repository
1730 using the following values (see section 2.2.10.2 applyACL):
- 1731 ○ **objectonly** indicates, that the repository is able to apply ACEs to a document or folder,
1732 without changing the ACLs of other objects.
 - 1733 ○ **propagate**: indicates that the ACEs is to be applied to the given object and all inheriting
1734 objects.
 - 1735 ○ **repositorydetermined** indicates, that the repository has its own mechanism of
1736 computing how changing an ACL for an object influences the non-direct ACEs of other
1737 objects.
- 1738 • **<Array> PermissionDefinition repositoryPermissions**: is a list with names and descriptions of
1739 the supported permissions.
- 1740 • **<Array> PermissionMapping mappings**: contains a list with mappings for the basic CMIS
1741 permissions to allowed actions.

1742 2.1.8.3.1 Supported Permissions

1743 The list of permission definitions returned by *getRepositoryInfo* lists all the *permissions* a repository
1744 supports. This list also includes the CMIS permissions if supported by the repository.

1745 A *PermissionDefinition* holds:

- 1746 • **String permission**: the (technical) name of the permission (unique within the list of permission
1747 definitions).
- 1748 • **(Optional) String description**: an optional description of the permission that should be used as
1749 the permission's name to be presented to the user.

1750 2.1.8.3.2 AllowableActions & Permission Mapping

1751 CMIS provides a mechanism called "AllowableActions" which allows an application to discover the set of
1752 service operations that can currently be performed on a particular object, without having to actually invoke
1753 the service.

1754 The set of allowable actions on an object at a point in time are affected not only by CMIS ACLs, but also
1755 by other factors such as:

- 1756 • Constraints inherent in the CMIS Domain Model based on the object's base type or current
1757 versioning state.
 - 1758 • Policies or other control mechanisms that are opaque to CMIS.
- 1759

1760 CMIS defines several services that applications can use at run-time to discover the AllowableActions for
1761 an object.

1762 If a Repository supports ACLs, then the repository MUST provide a mapping table that defines how the
1763 permissions supported by the repository interact with the CMIS allowable actions, i.e. which permissions
1764 are necessary for a principal to have on one or more objects in order to potentially perform each action,
1765 subject to the other constraints on allowable actions above.

1766 This section defines both the allowable actions as well as how those actions are presented in the
1767 PermissionMapping table.

1768 The Permission Mapping table contains a set of (*key*, *permissions*) pairs:

- 1769 • **String Key:** Because several allowable actions may require permissions on more than one object
1770 – for example, moving a document from one folder to another may require permissions on the
1771 document and each of the folders – the mapping table is defined in terms of permission “keys”,
1772 where each key combines the name of the allowable action as the object for which the principal
1773 needs the required permission.
 - 1774 ○ For example – the canMoveObject.Source key indicates the permissions that the
1775 principal must have on the “source folder” to move an object from that folder into another
1776 folder.
- 1777 • **<Array> String permissions:** The names of one or more permissions that the principal MUST
1778 have. If more than one permission is specified, then the principal MUST be allowed to perform the
1779 operation if they have ANY of the listed permissions.

1780 The list below defines all mapping keys, as well as a permissions mapping that repositories SHOULD
1781 use. Repositories MAY require additional permissions.

1782 For convenience, the list below groups all mapping entries by the underlying Allowable Actions, and
1783 includes descriptive information. For each Allowable Action the following information is given:

1784	Description:	The description and name of the service the AllowableAction enables.
1785	Base Object:	The base object-types for which the allowable action MAY be TRUE.
1786	Operand:	The object the permission applies to.
1787	Key:	The permission mapping key.
1788	Permissions:	The permission values.

1789

1790 **Navigation Services:**

1791 **canGetDescendants**

1792	Description:	Can get the descendants of the folder (<code>getDescendants</code>)
1793	Base Object:	cmis:folder
1794	Operand:	cmis:folder
1795	Key:	canGetDescendants.Folder
1796	Permission:	Read

1797

1798 **canGetFolderTree**

1799	Description:	Can get the sub-folder tree of the folder (<code>getFolderTree</code>)
1800	Base Object:	cmis:folder
1801	Operand:	cmis:folder
1802	Key:	canGetFolderTree.Folder
1803	Permission:	Read

1804

1805 **canGetChildren**

1806	Description:	Can get the children of the folder (<code>getChildren</code>)
1807	Base Object:	<code>cmis:folder</code>
1808	Operand:	<code>cmis:folder</code>
1809	Key:	<code>canGetChildren.Folder</code>
1810	Permission:	Read
1811		
1812	canGetFolderParent	
1813	Description:	Can get the parent/ancestor folder(s) of the folder (<code>getFolderParent</code>)
1814	Base Object:	<code>cmis:folder</code>
1815	Operand:	<code>cmis:folder</code>
1816	Key:	<code>canGetFolderParent.Folder</code>
1817	Permission:	Read
1818		
1819	canGetObjectParents	
1820	Description:	Can get the parent folders of the object. (<code>getObjectParents</code>)
1821	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code> , <code>cmis:policy</code>
1822	Operand	Object
1823	Key:	<code>canGetObjectParents.Object</code>
1824	Permission:	Read
1825		
1826	Object Services:	
1827	canCreateDocument	
1828	Description:	Can create a <code>cmis:document</code> Object in the folder (<code>createDocument</code>)
1829	Base Object:	<code>cmis:folder</code>
1830	Operand:	Folder
1831	Key:	<code>canCreateDocument.Folder</code>
1832	Permission:	Read
1833		
1834	canCreateFolder	
1835	Description:	Can create a <code>cmis:folder</code> Object as a child of the specified folder
1836	(<code>createFolder</code>)	
1837	Base Object:	<code>cmis:folder</code>
1838	Operand:	Folder
1839	Key:	<code>canCreateFolder.Folder</code>
1840	Permission:	Read
1841		
1842	canCreateRelationship	
1843	Description:	Can create a Relationship in which this Object is a source
1844	(<code>createRelationship</code>)	
1845	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code>
1846	Operand:	Object
1847	Key:	<code>canCreateRelationship.Source</code>
1848	Permission:	Read

1849		
1850	canCreateRelationship	
1851	Description:	Can create a Relationship in which this Object is a target
1852		(createRelationship)
1853	Base Object:	cmis:document, cmis:folder
1854	Operand:	Object
1855	Key:	canCreateRelationship.Target
1856	Permission:	Read
1857		
1858	canGetProperties	
1859	Description:	Can read the properties of this object (getProperties)
1860	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1861	Operand:	Object
1862	Key:	canGetProperties.Object
1863	Permission:	Read
1864		
1865	canGetRenditions	
1866	Description:	Can retrieve the renditions of this object (getRenditions)
1867	Base Object:	cmis:document, or cmis:folder
1868	Operand:	Object
1869	Key:	canGetRenditions.Object
1870	Permission:	Read
1871		
1872	canGetContentStream	
1873	Description:	Can get the content stream for the Document object
1874		(getContentStream)
1875	Base Object:	cmis:document
1876	Operand:	Object
1877	Key:	canGetContentStream.Object
1878	Permission:	Read
1879		
1880	canUpdateProperties	
1881	Description:	Can update the properties of this object (updateProperties)
1882	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1883	Operand:	Object
1884	Key:	canUpdateProperties.Object
1885	Permission:	Write
1886		
1887	canMoveObject	
1888	Description:	Can move the object (moveObject)
1889	Base Object:	cmis:document, cmis:folder, cmis:policy
1890	Operand:	Object
1891	Key:	canMoveObject.Object

1892	Permission:	Write
1893		
1894	canMoveObject	
1895	Description:	Can move an object into this folder (moveObject)
1896	Base Object:	cmis:folder
1897	Operand:	Folder
1898	Key:	canMoveObject.Target
1899	Permission:	Read
1900		
1901	canMoveObject	
1902	Description:	Can move an object from this folder (moveObject)
1903	Base Object:	cmis:folder
1904	Operand:	Folder
1905	Key:	canMoveObject.Source
1906	Permission:	Read
1907		
1908	canDeleteObject	
1909	Description:	Can delete this object (deleteObject)
1910	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1911	Operand:	Object
1912	Key:	canDelete.Object
1913	Permission:	Write
1914		
1915	canDeleteObject	
1916	Description:	Can delete an object that is a child of this folder (deleteObject)
1917	Base Object:	cmis:folder
1918	Operand:	Folder
1919	Key:	canDelete.Folder
1920	Permission:	Read
1921		
1922	canSetContentStream	
1923	Description:	Can set the content stream for the Document object
1924		(setContentStream)
1925	Base Object:	cmis:document
1926	Operand:	Object
1927	Key:	canSetContentStream.Document
1928	Permission:	Write
1929		
1930	canDeleteContentStream	
1931	Base Object:	cmis:document
1932	Action:	Can delete the content stream for the Document object
1933		(deleteContentStream)
1934	Operand:	Object

1935	Key:	canDeleteContentStream.Document
1936	Permission:	Write
1937		
1938	canDeleteTree	
1939	Base Object:	cmis:folder
1940	Action:	Can delete the folder and all contained objects (deleteTree)
1941	Operand:	Object
1942	Key:	canDeleteTree.Folder
1943	Permission:	Write
1944		
1945	Filing Services:	
1946	canAddObjectToFolder	
1947	Description:	Can file the document in a folder (addObjectToFolder)
1948	Base Object:	cmis:document, cmis:policy
1949	Operand:	Object
1950	Key:	canAddToFolder.Object
1951	Permission:	Read
1952		
1953	canAddObjectToFolder	
1954	Description:	Can file a document in the specified folder (addObjectToFolder)
1955	Base Object:	cmis:document, cmis:policy
1956	Operand:	Object
1957	Key:	canAddToFolder.Folder
1958	Permission:	Read
1959		
1960	canRemoveObjectFromFolder	
1961	Description:	Can unfile the specified document from a folder
1962	(removeObjectFromFolder)	
1963	Base Object:	cmis:document, cmis:policy
1964	Operand:	Object
1965	Key:	canRemoveObjectFromFolder.Object
1966	Permission:	Read
1967		
1968	canRemoveObjectFromFolder	
1969	Description:	Can unfile a document from the specified folder
1970	(removeObjectFromFolder)	
1971	Base Object:	cmis:document, cmis:policy
1972	Operand:	Object
1973	Key:	canRemoveObjectFromFolder.Folder
1974	Permission:	Read

1975
1976 **Versioning Services:**

1977	canCheckOut	
1978	Description:	Can check out the Document object (<code>checkOut</code>)
1979	Base Object:	cmis:document
1980	Operand:	Object
1981	Key:	canCheckOut.Document
1982	Permission:	Write
1983		
1984	canCancelCheckOut	
1985	Description:	Can cancel the check out the Document object (<code>cancelCheckOut</code>)
1986	Base Object:	cmis:document
1987	Operand:	Object
1988	Key:	canCancelCheckout.Document
1989	Permission:	Write
1990		
1991	canCheckIn	
1992	Description:	Can check in the Document object (<code>checkIn</code>)
1993	Base Object:	cmis:document
1994	Operand:	Object
1995	Key:	canCheckin.Document
1996	Permission:	Write
1997		
1998	canGetAllVersions	
1999	Description:	Can get the version series for the Document object (<code>getAllVersions</code>)
2000	Base Object:	cmis:document
2001	Operand:	Object
2002	Key:	canGetAllVersions.Document
2003	Permission:	Read
2004		
2005	Relationship Services:	
2006	canGetObjectRelationships	
2007	Description:	Can get the relationship in which this object is a source/target (<code>getObjectRelationships</code>)
2008		
2009	Base Object:	cmis:document, cmis:folder, cmis:policy
2010	Operand:	Object
2011	Key:	canGetObjectRelationships.Object
2012	Permission:	Read
2013		
2014	Policy Services:	
2015	canApplyPolicy	
2016	Description:	Can apply a policy to the Object (<code>applyPolicy</code>)

2017	Base Object:	cmis:document, cmis:folder
2018	Operand:	Object
2019	Key:	canAddPolicy.Object
2020	Permission:	Read
2021		
2022	canApplyPolicy	
2023	Description:	Can apply the specified policy to an Object (<code>applyPolicy</code>)
2024	Base Object:	cmis:policy
2025	Operand:	Object
2026	Key:	canAddPolicy.Policy
2027	Permission:	Read
2028		
2029	canRemovePolicy	
2030	Description:	Can remove a policy from the specified Object (<code>removePolicy</code>)
2031	Base Object:	cmis:document, cmis:folder
2032	Operand:	Object
2033	Key:	canRemovePolicy.Object
2034	Permission:	Read
2035		
2036	canRemovePolicy	
2037	Description:	Can remove the specified policy from an Object (<code>removePolicy</code>)
2038	Base Object:	cmis:document, cmis:folder
2039	Operand:	cmis:policy
2040	Key:	canRemovePolicy.Policy
2041	Permission:	Read
2042		
2043	canGetAppliedPolicies	
2044	Description:	Can get the list of Policies applied to the Object
2045		(<code>getAppliedPolicies</code>)
2046	Base Object:	cmis:document, cmis:folder
2047	Operand:	Object
2048	Key:	canGetAppliedPolicies.Object
2049	Permission:	Read
2050		
2051	ACL Services:	
2052	canGetACL	
2053	Description:	Can get ACL for object (<code>getACL</code>)
2054	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2055	Operand:	Object
2056	Key:	canGetACL.Object
2057	Permission:	Read
2058		

2059	canApplyACL	
2060	Description:	Can apply ACL to this object (applyACL)
2061	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2062	Operand:	Object
2063	Key:	canApplyACL.Object
2064	Permission:	Write
2065		

2066

2067 2.1.9 Versioning

2068 CMIS supports versioning of Document objects. Folder objects, relationship objects, and policy objects
2069 cannot be versioned.

2070 Whether or not a Document object is versionable (i.e. whether or not operations performed on the object
2071 via the Versioning Services MUST be allowed) is specified by the “*versionable*” attribute on its Object-
2072 type.

2073 A **version** of a Document object is an explicit/“deep” copy of the object, preserving its state at a certain
2074 point in time. Each version of a Document object is itself a Document object, i.e. has its own *ObjectId*,
2075 property values, MAY be acted upon using all CMIS services that act upon Document objects, etc.

2076 2.1.9.1 Version Series

2077 A **version series** for a Document object is a transitively closed collection of all Document objects that
2078 have been created from an original Document in the Repository. Each version series has a unique,
2079 system-assigned, and immutable **version series ID**.

2080 The version series has transitive closure -- that is, if object B is a version of object A, and object C is a
2081 version of object B, then object C is also a version of object A. The objects in a version series can be
2082 conceptually sequenced by their respective *CreationDate* properties.

2083 Additionally, the repository MAY expose a textual **VersionLabel** that describes to a user the position of
2084 an individual object with respect to the version series. (For example, version 1.0).

2085 *Note:* A Document object that is NOT versionable will always have a single object in its Version Series. A
2086 versionable Document object MAY have one or more objects in its Version Series.

2087 2.1.9.2 Latest Version

2088 The version that has the most recent *LastModificationDate* is called the **Latest Version** of the series, or
2089 equivalently, the latest version of any Document object in the series.

2090 When the latest version of a version series is deleted, a previous version (if there is one) becomes the
2091 latest version.

2092 2.1.9.2.1 Behavioral constraints on non-Latest Versions

2093 Repositories NEED NOT allow the non-latest versions in a Version Series to be updated, queried, or
2094 searched.

2095 2.1.9.3 Major Versions

2096 A Document object in a Version Series MAY be designated as a **Major Version**.

2097 The CMIS specification does not define any semantic/behavioral differences between Major and non-
2098 Major versions in a Version Series. Repositories may enforce/apply additional constraints or semantics for
2099 Major versions, if the effect on CMIS services remains consistent with an allowable behavior of the CMIS
2100 model.

2101 If the Version Series contains one or more Major versions, the one that has the most recent
2102 *LastModificationDate* is the **Latest Major Version** of the version series.

2103 (Note that while a Version Series MUST always have a *Latest Version*, it NEED NOT have a *Latest Major*
2104 *Version*.)

2105 When the latest major version is deleted, a previous major version (if there is one) becomes the latest
2106 major version.

2107 2.1.9.4 Services that modify Version Series

2108 2.1.9.4.1 Checkout

2109 A new version of a versionable Document object is created when the *checkIn* service is invoked on the
2110 Private Working copy (PWC) of this object. A PWC is created by invoking *checkOut* on a versionable
2111 Document object. A repository MAY allow *any* Document object in a version series to be checked out, or
2112 MAY only allow the *Latest Version* to be checked out.

2113 The effects of invoking the *checkout* service MUST be as follows:

- 2114 • A new Document object, referred to herein as the **Private Working Copy (PWC)**, is created.
 - 2115 ○ The PWC NEED NOT be visible to users who have permissions to view other Document
 - 2116 objects in the Version Series.
 - 2117 ○ Until it is checked in (using the *checkIn* service), the PWC MUST NOT be considered the
 - 2118 *LatestMajorVersion* in the Version Series.
 - 2119 ○ The property values for the PWC SHOULD be identical to the properties of the Document
 - 2120 object on which the *checkout* service was invoked. Certain properties such as *cmis:objectId*
 - 2121 may be different. Properties such as *cmis:creationDate* most likely will be different. The
 - 2122 content-stream of the PWC MAY be identical to the content-stream of the Document object
 - 2123 on which the *checkout* service was invoked, or MAY be “not set”.

2124 After a successful *checkout* operation is completed, and until such time when the PWC is deleted (via the
2125 *cancelCheckOut* service) or checked-in (via the *checkIn*) service, the effects on other Documents in the
2126 Version Series MUST be as follows:

- 2127 • The repository MUST throw an exception if the *checkout* service is invoked on any Document in
- 2128 the Version Series. (I.e. there can only be one PWC for a version series at a time.)
- 2129 • The value of the *cmis:isVersionSeriesCheckedOut* property MUST be TRUE.
- 2130 • The value of the *cmis:versionSeriesCheckedOutBy* property MAY be set to a value indicating
- 2131 which user created the PWC. (The Repository MAY still show the “not set” value for this
- 2132 property.)
- 2133 • The value of the *cmis:versionSeriesCheckedOutId* property MAY be set to the *ObjectId* of the
- 2134 PWC. (The Repository MAY still show the “not set” value for this property).
- 2135 • The repository MAY prevent operations that modify or delete the other Documents in the Version
- 2136 Series.

2137 2.1.9.4.2 Updates to the Private Working Copy

2138 If the repository supports the optional “PWCUpdatable” capability, then the repository MUST allow
2139 authorized users to modify the PWC Object using the Object services (e.g. *UpdateProperties*).

2140 If the repository does NOT support the “PWCUpdatable” capability, then the PWC object can only be
2141 modified as part of the *checkIn* service call.

2142 2.1.9.4.3 Discarding Check out

2143 An authorized user MAY discard the check-out using the *cancelCheckOut* service on any Document in
2144 the Version Series or by using the *deleteObject* service on the PWC Object. The effects of discarding a
2145 check-out MUST be as follows:

- 2146 • The PWC Object MUST be deleted.
- 2147 • For all other Documents in the Version Series:
 - 2148 ○ The value of the *cmis:isVersionSeriesCheckedOut* property MUST be FALSE.
 - 2149 ○ The value of the *cmis:versionSeriesCheckedOutBy* property MUST be “not set”.
 - 2150 ○ The value of the *cmis:versionSeriesCheckedOutId* property MUST be “not set”.
 - 2151 ○ The repository MUST allow authorized users to invoke the *checkout* service.

2.1.9.4.4 Checkin

An authorized user/application MAY “check in” the Private Working Copy object via the *checkIn* service.

The *checkIn* service allows users/applications to provide update property values and a content-stream for the PWC object.

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

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

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

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

2.1.9.5 Versioning Properties on Document Objects

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

cmis:isLatestVersion Boolean

TRUE if the Document object is the *Latest Version* in its *Version Series*. FALSE otherwise.

cmis:isMajorVersion Boolean

TRUE if the Document object is a *Major Version* in its *Version Series*. FALSE otherwise.

cmis:isLatestMajorVersion Boolean

TRUE if the Document object is the *Latest Major Version* in its *Version Series*. FALSE otherwise.

cmis:versionLabel String (optional)

Optional textual description the position of an individual object with respect to the version series. (For example, version 1.0).

cmis:versionSeriesId ID

ID of the Version Series for this Object.

cmis:isVersionSeriesCheckedOut Boolean

TRUE if there currently exists a Private Working Copy for this Version Series. FALSE otherwise

2195
2196 **cmis:versionSeriesCheckedOutBy** String
2197 If IsVersionSeriesCheckedOut is TRUE: then an identifier for the user who created the Private
2198 Working Copy. "Not set" otherwise.

2199
2200 **cmis:versionSeriesCheckedOutId** ID
2201 If IsVersionSeriesCheckedOut is TRUE: The Identifier for the Private Working Copy. "Not set"
2202 otherwise.

2203
2204 **cmis:checkinComment** String
2205 Textual comment associated with the given version.

2206 *Note:* Changes made via the Versioning Services that affect the values of these properties MUST NOT
2207 constitute modifications to the Document objects in the Version Series (e.g. MUST NOT affect the
2208 cmis:lastModificationDate, etc.)

2209 **2.1.9.6 Document Creation and Initial Versioning State**

2210 A repository MAY create new Document objects in a "Private Working Copy" state when they are created
2211 via the *createDocument* or *createDocumentFromSource* services. This state is logically equivalent to
2212 having a Version Series that contains exactly one object (the PWC) and 0 other documents.

2213 The repository MAY also create new Document objects in a "Major Version" state. This state is logically
2214 equivalent to having a Version Series that contains exactly one Major Version and 0 other documents.

2215 The repository MAY also create new Document objects in a "Non-Major Version" state. This state is
2216 logically equivalent to having a Version Series that contains exactly one Non-Major Version and 0 other
2217 documents.

2218 If the repository does not support versioning the repository MUST ignore the value of the versioningState
2219 parameter.

2220 **2.1.9.7 Version Specific/Independent membership in Folders**

2221 Repositories MAY treat membership of a Document object in a folder collection as "version-specific" or
2222 "version-independent".

2223 Repositories MUST indicate whether they support version-specific membership in a folder via the
2224 "VersionSpecificFiling" optional capability flag.

2225 If the repository is treating folder collection membership as "version-independent", then:

- 2226 • Moving or Filing a Document Object into a folder MUST result in ALL Documents in the Version
2227 Series being moved/added into the folder.
- 2228 • The Repository MAY return only the latest-version OR latest major-version Document object in a
2229 version series in the response to Navigation service requests (getChildren, getDescendants), and
2230 NEED NOT return other Document Objects filed in the folder that are in the Version Series.

2231 If the repository is treating folder collection membership as "version-specific", then moving or Filing a
2232 Document Object into a folder MUST NOT result in other Documents in the Version Series being
2233 moved/added.

2234 **2.1.9.8 Version Specific/Independent membership in Relationships**

2235 A relationship object MAY have either a version-specific or version-independent binding to its source
2236 and/or target objects. This behavior MAY vary between repositories and between individual relationship
2237 types defined for a Repository.

2238 If a relationship object has a version-independent binding to its source/target object, then:

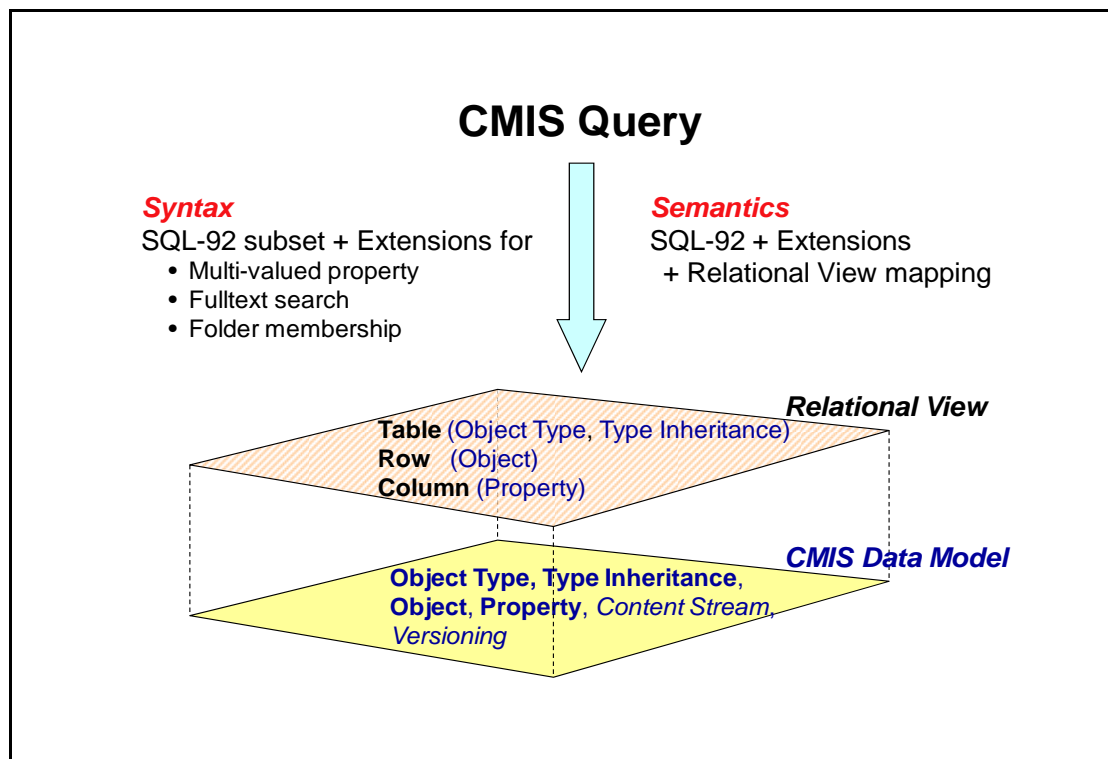
- 2239 • The getObjectRelationships service invoked on a Document Object MUST return the relationship
2240 if Relationship was source/target is set to ANY Document Object in the Version Series.
- 2241 If a relationship object has a version-specific binding to its source/target object, then:
- 2242 • The getObjectRelationships service invoked on a Document Object MUST return the relationship
2243 if Relationship was source/target is set to the ID of the Document Object on which the service was
2244 invoked.

2245 **2.1.9.9 Versioning visibility in Query Services**

- 2246 Repositories MAY include non-latest-versions of Document Objects in results to the Discovery Services
2247 (query).
- 2248 Repositories MUST indicate whether they support querying for non-latest-versions via the
2249 “AllVersionsSearchable” optional capability flag.
- 2250 If “AllVersionsSearchable” is TRUE then the Repository MUST include in the query results ANY
2251 Document Object in the Version Series that matches the query criteria. (subject to other query constraints
2252 such as security.)
- 2253 Additionally, repositories MAY include Private Working Copy objects in results in results to the Discovery
2254 Services (query).
- 2255 Repositories MUST indicate whether they support querying for Private Working Copy objects via the
2256 “PWCSearchable” optional capability flag.
- 2257 If “PWCSearchable” is TRUE then the Repository MUST include in the query results ANY Private Working
2258 Copy Document Objects that matches the query criteria (subject to other query constraints such as
2259 security.)
- 2260 If “PWCSearchable” is FALSE then the Repository MUST NOT include in the query results ANY Private
2261 Working Copy Document Objects that match the query criteria (subject to other query constraints such as
2262 security.)

2263 **2.1.10 Query**

- 2264 CMIS provides a type-based query service for discovering objects that match specified criteria, by
2265 defining a read-only projection of the CMIS data model into a *Relational View*.
- 2266 Through this relational view, queries may be performed via a simplified SQL SELECT statement. This
2267 query language is based on a subset of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database
2268 Language SQL), with a few extensions to enhance its filtering capability for the CMIS data model, such as
2269 existential quantification for multi-valued property, full-text search, and folder membership. Other
2270 statements of the SQL language are not adopted by CMIS. The semantics of this query language is
2271 defined by the SQL-92 standard, plus the extensions, in conjunction with the model mapping defined by
2272 CMIS’s relational view.



2273

2274 2.1.10.1 Relational View Projection of the CMIS Data Model

2275 The relational view of a CMIS repository consists of a collection of virtual tables that are defined on top of
2276 the CMIS data model. This relational view is used for query purposes only.

2277 In this relational view a **Virtual Table** is implicitly defined for each *queryable* Object-Type defined in the
2278 repository. (Non-queryable Object-Types are NOT exposed through this Relational View.)

2279 In each **Virtual Table**, a **Virtual Column** is implicitly defined for each property defined in the Object-Type
2280 Definition AND for all properties defined on ANY ancestor-type of the Object-Type but NOT defined in the
2281 Object-Type definition. Virtual Columns for properties defined on ancestor-types of the Object-type but
2282 NOT defined in the Object-Type definition MUST contain the SQL NULL value. Virtual Columns for
2283 properties whose value is “not set” MUST contain the SQL NULL value.

2284 An object-type’s *queryName* attribute is used as the table name for the corresponding virtual table, and a
2285 property’s *queryName* attribute is used as the column name for the corresponding table column. Please
2286 see the restrictions on *queryName* in the appropriate data model section.

2287 The Virtual Column for a multi-valued property MUST contain a single list value that includes all values of
2288 the property.

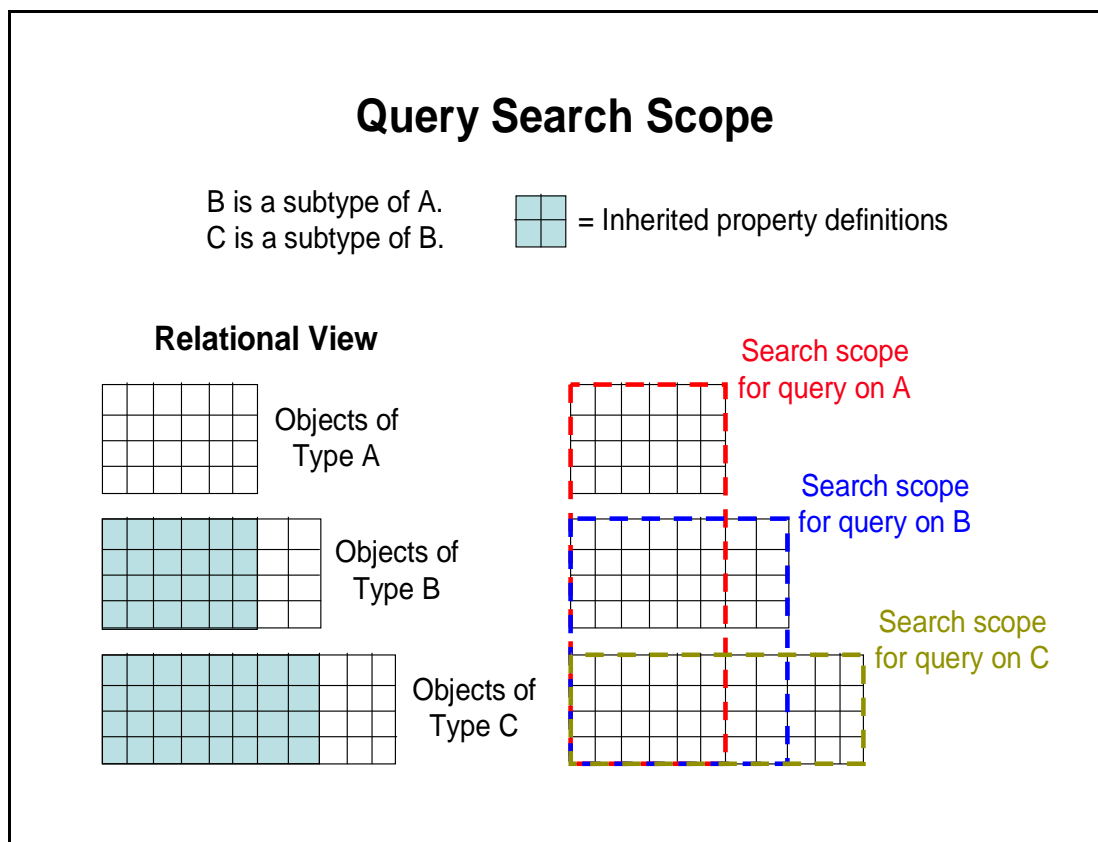
2289 2.1.10.1.1 Object-Type Hierarchy in the Relational View Projection

2290 The Relational View projection of the CMIS Data Model ensures that the Virtual Table for a particular
2291 Object-type is a complete super-set of the Virtual Table for any and all of its ancestor types.

2292 Additionally, an Object-Type definition’s “*includedInSupertypeQuery*” specifies whether objects of that
2293 Object-Type MUST be included in the Virtual Table for any of its ancestor types. If the
2294 “*includedInSupertypeQuery*” attribute of the Object-Type is FALSE, then objects of that Object-Type
2295 MUST NOT be included in the Virtual Table for any of its ancestor types.

2296 Thus the Virtual Table for an Object-type includes a row not only for each Object of that type, but all
 2297 Objects of any of that Object-types' Descendant Types for which the *"includedInSupertypeQuery"*
 2298 attribute is TRUE.

2299 But since the Virtual Table will include only columns for properties defined in the Object-Type underlying
 2300 the Virtual Table, a row that is a query result representing an Object of a Descendant Type can only
 2301 include those columns for properties defined on the Object-Type underlying the Virtual Table.



2302

2303 2.1.10.1.2 Content Streams

2304 Content-streams are NOT exposed through this relational view.

2305 2.1.10.1.3 Result Set

2306 When a query is submitted, a set of pseudo CMIS objects will be returned. These pseudo objects are
 2307 comprised of the properties specified in the select clause of the query statement.

2308 For each property in each object in the result set, the Repository MUST include the property definition ID
 2309 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
 2310 used).

2311 If the select clause of the query statement contains properties from a single type reference then the
 2312 repository MAY represent these pseudo-objects with additional object information.

2313 2.1.10.2 Query Language Definition

2314 This query languages is based on a subset of the SQL-92 grammar. CMIS-specific language extensions
 2315 to SQL-92 are called out explicitly.

2316 The basic structure of a CMIS query is a SQL statement that MUST include the following clauses:

- 2317 • **SELECT [virtual columns]:** This clause identifies the set of virtual columns that will be included
2318 in the query results for each row.
- 2319 • **FROM [Virtual Table Names]:** This clause identifies which Virtual Table(s) the query will run
2320 against.
- 2321 Additionally, a CMIS query MAY include the following clauses:
- 2322 • **WHERE [conditions]:** This clause identifies the constraints that rows MUST satisfy to be
2323 considered a result for the query.
- 2324 • **ORDER BY [sort specification]:** This clause identifies the order in which the result rows MUST
2325 be sorted in the result row set.

2326 2.1.10.2.1 BNF Grammar

2327 This BNF grammar is a “subset” of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language
2328 SQL), except for some production alternatives. Specifically, except for these extensions, the following
2329 production rules are derived from the SQL-92 grammar. The non-terminals used in this grammar are also
2330 borrowed from the SQL-92 grammar without altering their semantics. Accordingly, the non-terminal
2331 <column name> is used for single-valued properties only so that the semantics of SQL can be preserved
2332 and borrowed. This approach not only facilitates comparison of the two query languages, and simplifies
2333 the translation of a CMIS query to a SQL query for a RDBMS-based implementation, but also allows
2334 future expansion of this query language to cover a larger subset of SQL with minimum conflict. The CMIS
2335 extensions are introduced primarily to support multi-valued properties and full-text search, and to test
2336 folder membership. Multi-valued properties are handled separately from single-valued properties, using
2337 separate non-terminals and separate production rules to prevent the extensions from corrupting SQL-92
2338 semantics.
2339

2340
 2341 <CMIS 1.0 query statement> ::= <simple table> [<order by clause>]
 2342 <simple table> ::= SELECT <select list> <from clause> [<where clause>]
 2343 <select list> ::= "*"

 2344 | <select sublist> [{ "," <select sublist> }...]
 2345 <select sublist> ::= <value expression> [[AS] <column name>]

 2346 | <qualifier> "."

 2347 | <multi-valued-column reference>
 2348 <value expression> ::= <column reference> | <numeric value function>
 2349 <column reference> ::= [<qualifier> "."] <column name>
 2350 <multi-valued-column reference> ::= [<qualifier> "."] <multi-valued-column name>
 2351 <numeric value function> ::= SCORE()
 2352 <qualifier> ::= <table name> | <correlation name>
 2353 <from clause> ::= FROM <table reference>
 2354 <table reference> ::= <table name> [[AS] <correlation name>]

 2355 | <joined table>
 2356 <joined table> ::= "(" <joined table> ")"

 2357 | <table reference> [<join type>] JOIN <table reference> <join specification>
 2358 <join type> ::= INNER | LEFT [OUTER]
 2359 <join specification> ::= ON <column reference> "=" <column reference>
 2360 <where clause> ::= WHERE <search condition>
 2361 <search condition> ::= <boolean term> | <search condition> OR <boolean term>
 2362 <boolean term> ::= <boolean factor> | <boolean term> AND <boolean factor>
 2363 <boolean factor> ::= [NOT] <boolean test>
 2364 <boolean test> ::= <predicate> | "(" <search condition> ")"
 2365 <predicate> ::= <comparison predicate> | <in predicate> | <like predicate> | <null predicate>

 2366 | <quantified comparison predicate> | <quantified in predicate>
 2367 | <text search predicate> | <folder predicate>
 2368 <comparison predicate> ::= <value expression> <comp op> <literal>
 2369 <comp op> ::= "=" | "<" | "<" | ">" | "<=" | ">="

 2370 <literal> ::= <signed numeric literal> | <character string literal> | <datetime literal> | <boolean literal>
 2371 <in predicate> ::= <column reference> [NOT] IN "(" <in value list> ")"
 2372 <in value list> ::= <literal> [{ "," <literal> }...]
 2373 <like predicate> ::= <column reference> [NOT] LIKE <character string literal>
 2374 <null predicate> ::= { <column reference> | <multi-valued-column reference> } IS [NOT] NULL
 2375 <quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>
 2376 <quantified in predicate> ::= ANY <multi-valued-column reference> [NOT] IN "(" <in value list> ")"
 2377 <text search predicate> ::= CONTAINS "("

 2378 [<qualifier> " ,"] <quote> <text search expression> <quote> ")"
 2379 <folder predicate> ::= { IN_FOLDER | IN_TREE } "(" [<qualifier> " ,"] <folder id> ")"
 2380 <order by clause> ::= ORDER BY <sort specification> [{ "," <sort specification> }...]
 2381 <sort specification> ::= <column reference> [ASC | DESC]
 2382 <correlation name> ::= <identifier>

2.1.10.2.3.1 Join Support

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

- If the Repository's value for the capabilityJoin attribute is **none**, then no JOIN clauses can be used in queries.
- If the Repository's value for the capabilityJoin attribute is **inneronly**, then only inner JOIN clauses can be used in queries.
- If the Repository's value for the capabilityJoin attribute is **innerandouter**, then inner and/or outer JOIN clauses can be used in queries.

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

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

2.1.10.2.4 WHERE Clause

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

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

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

2.1.10.2.4.1 Comparisons permitted in the WHERE clause.

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

Repositories SHOULD support the comparisons for the property types as described in the list below.

Repositories MAY support additional comparisons and operators. Any additional operators not specified are repository-specific:

<Property Type>

Supported Operators: <List of Operators supported on Type>

Supported Literal: <Supported type of Literal in comparison>

String (Single)

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

Supported Literal: String

String (IN)

Supported Operators: [NOT] IN

Supported Literal: List of Strings

Decimal

Supported Operators: =, <>, <, <=, >, >=

Supported Literal: Decimal

2466 Decimal (IN)
2467 Supported Operators: [NOT] IN
2468 Supported Literal: List of Decimal
2469
2470 Integer
2471 Supported Operators: =, <>, <, <=, >, >=
2472 Supported Literal: Integer
2473
2474 Integer (IN)
2475 Supported Operators: [NOT] IN
2476 Supported Literal: List of Integer
2477
2478 Boolean
2479 Supported Operators: =
2480 Supported Literal: <boolean literal>
2481
2482 DateTime
2483 Supported Operators: =, <>, <*, <=*, >*, >=*
2484 Supported Literal: <datetime literal>
2485 * - comparison is based on chronological before or after date.
2486
2487 DateTime (IN)
2488 Supported Operators: [NOT] IN
2489 Supported Literal: List of <datetime literal>'s
2490
2491 ID
2492 Supported Operators: =, <>
2493 Supported Literal: String
2494
2495 ID (IN)
2496 Supported Operators: [NOT] IN
2497 Supported Literal: List of strings
2498
2499 URI
2500 Supported Operators: =, <>
2501 Supported Literal: String
2502
2503 URI (IN)
2504 Supported Operators: [NOT] IN
2505 Supported Literal: List of strings
2506
2507 URI

2508 Supported Operators: [NOT] LIKE

2509 Supported Literal: String

2510

2511 Operations on the SCORE() output MUST be treated the same as decimal operations.

2512

2513 When using properties in a join statement, comparison MUST be allowed on properties of the same types
2514 as defined by the table above. Repositories MAY extend this behavior.

2515

2516 The ANY operation argument MUST be one of the properties found in the table above which supports
2517 equality operations

2518 **2.1.10.2.4.2 Multi-valued property support (SQL-92 Extension)**

2519 The CMIS query language includes several new non-terminals to expose semantics for querying multi-
2520 valued properties, in a way that does not alter the semantics of existing SQL-92 production rules.

2521 **2.1.10.2.4.2.1 Multi-valued column references**

2522 **BNF grammar structure:** <Multi-valued-column reference>, <multi-valued-column name>

- 2523 • These are non-terminals defined for multi-valued properties whereas SQL-92's <column
2524 reference> and <column name> are retained for single-valued properties only. This is to preserve
2525 the single-value semantics of a regular "column" in the SQL-92 grammar.

2526 **2.1.10.2.4.2.2 <Quantified comparison predicate>**

2527 The SQL-92 production rule for <quantified comparison predicate> is extended to accept a multi-valued
2528 property in place of a <table subquery>. This operation is restricted to equality tests only.

2529

2530 <Table subquery> is not supported in CMIS-SQL.

2531

2532 The SQL-92 <quantifier> is restricted to ANY only.

2533

2534 The SQL-92 <row value constructor> is restricted to a literal only.

2535 **Example:**

```
2536 SELECT      Y.CLAIM_NUM, X.PROPERTY_ADDRESS, Y.DAMAGE_ESTIMATES  
2537 FROM POLICY AS X JOIN CLAIMS AS Y ON ( X.POLICY_NUM = Y.POLICY_NUM )  
2538 WHERE      ( 100000 = ANY Y.DAMAGE_ESTIMATES )
```

2539 *(Note: DAMAGE_ESTIMATES is a multi-valued Integer property.)*

2540 **2.1.10.2.4.2.3 IN/ANY Predicate**

2541 **BNF grammar structure:** <Quantified in predicate>

2542

2543 CMIS-SQL exposes a new IN predicate defined for a multi-valued property. It is modeled after the SQL-
2544 92 IN predicate, but since the entire predicate is different semantically, it has its own production rule in
2545 the BNF grammar below.

2546

2547 The quantifier is restricted to ANY. The predicate MUST be evaluated to TRUE if at least one of the
2548 property's values is (or, is not, if NOT is specified) among the given list of literal values. Otherwise the
2549 predicate is evaluated to FALSE.

2550

2551 The ANY operation argument MUST be one of the properties found in the comparison list above which
2552 supports IN operations.

2553 **Example:**

```
2554         SELECT      *  
2555         FROM CAR_REVIEW  
2556         WHERE        (MAKE = 'buick' ) OR  
2557                     ( ANY FEATURES IN ('NAVIGATION SYSTEM', 'SATELLITE RADIO', 'MP3') )  
2558         (Note: FEATURES is a multi-valued String property.)
```

2559 2.1.10.2.4.3 CONTAINS() predicate function (CMIS-SQL Extension)

2560 **BNF grammar structure::** CONTAINS ([<qualifier> ,] ' <text search expression> ')

2561 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
2562 by a Repository ([See previous section.](#))

2563 **Inputs:**

2564 <Qualifier>

2565 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2566 FROM clause for the query.

- 2567 • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
2568 Table, but a repository MAY ignore the value of the parameter.
- 2569 • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
2570 throw an exception if the qualifier is not specified.

2571 <Text Search Expression>

2572 The <text search expression> parameter MUST be a character string , specifying the full-text
2573 search criteria.

2574

2575 The Text Search Expression may be a set of terms or phrases with an optional '-' to signal
2576 negation. A phrase is defined as a word or group of words. A group of words must be
2577 surrounded by quotes to be considered a single phrase.

2578

2579 Terms separated by whitespace are AND'ed together.

2580 Terms separated by "OR" are OR'ed together

2581 Implicit "AND" has higher precedence than "OR"

2582 Within a word or phrase, each (single-)quote must also be escaped by a preceding backslash "\"

2583 **Return value:**

2584 The predicate returns a Boolean value.

2585 The predicate MUST return TRUE if the object is considered by the repository as "relevant" with
2586 respect to the given <text search expression> parameter.

2587 The predicate MUST return FALSE if the object is considered by the repository as not "relevant"
2588 with respect to the given <text search expression> parameter.

2589 **Constraints:**

2590 At most one CONTAINS() function MUST be included in a single query statement. The repository
2591 MUST throw an exception if more than one CONTAINS() function is found.

2592

2593 The return value of the CONTAINS() function MAY only be included conjunctively (ANDed) with the
2594 aggregate of all other predicates, if there is any, in the WHERE clause.

2595 **2.1.10.2.4.4 SCORE() predicate function**

2596 **BNF grammar structure:** SCORE ()

2597 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided
2598 by a Repository ([See previous section.](#))

2599 **Inputs:** No inputs MUST be provided for this predicate function.

2600 **Return value:**

2601 The SCORE() predicate function returns a decimal value in the interval [0,1] .

2602 A repository MUST return the value 0 if the object is considered by the repository as having
2603 absolutely no relevance with respect to the CONTAINS() function specified in the query.

2604 A repository MUST return the value 1 if the object is considered by the repository as having
2605 absolutely complete relevance with respect to the CONTAINS() function specified in the query.

2606 **Constraints:**

2607 The SCORE() function MUST only be used in queries that also include a CONTAINS() predicate
2608 function

2609 The SCORE() function MUST only be used in the SELECT clause of a query. It MUST NOT be
2610 used in the WHERE clause or in the ORDER BY clauses.

2611 An alias column name defined for the SCORE() function call in the SELECT clause (i.e.,
2612 "SELECT SCORE() AS column_name ...") may be used in the ORDER BY clause.

2613 If SCORE() is included in the SELECT clause and an alias column name is not provided, then a
2614 query name of SEARCH_SCORE is used for the query output, and the property definition ID is
2615 repository-specific.

2616 **2.1.10.2.4.5 IN_FOLDER() predicate function**

2617 **BNF grammar structure:** IN_FOLDER([<qualifier>,] <folder id>)

2618 **Usage:** This is a predicate function that tests whether or not a candidate object is a child-object of the
2619 folder object identified by the given <folder id>.

2620 **Inputs:**

2621 **<qualifier>**

2622 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the
2623 FROM clause for the query.

- 2624 • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual
2625 Table, but a repository MAY ignore the value of the parameter.
2626 • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD
2627 throw an exception if the qualifier is not specified.

2628 **<folder id>**

2629 The value of this parameter MUST be the ID of a folder object in the repository.

2630 **Return value:**

2631 The predicate function MUST return TRUE if the object is a child-object of the folder specified by
2632 <folder id>.

2633 The predicate function MUST return FALSE if the object is a NOT a child-object of the folder
2634 specified by <folder id>.

2.1.10.2.4.6 IN_TREE() predicate function

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

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

Inputs:

<qualifier>

The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the 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.

<folder id>

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

Return value:

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

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

2.1.10.2.5 ORDER BY Clause

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

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

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

Collation rules for the ORDER BY clause are repository specific.

2.1.10.3 Escaping

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

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

2.1.11 Change Log

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

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

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

2.1.11.1 Completeness of the Change Log

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

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

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

2.1.11.2 Change Log Token

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

2.1.11.2.1 “Latest Change Token” repository information

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

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

2.1.11.3 Change Event

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

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

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

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

2.2 Services

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

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

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

2.2.1 Common Service Elements

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

2.2.1.1 Paging

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

Input Parameters:

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

Output Parameters:

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

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

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

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

2.2.1.2 Retrieving additional information on objects in CMIS service calls

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

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

- Properties (retrieves a subset instead of additional information)
- Relationships
- Renditions
- ACLs
- AllowableActions

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

2.2.1.2.1 Properties

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

Optional Input Parameter:

2763 • **String filter:** Value indicating which properties for Objects MUST be returned. Values are:

2764 ○ **Not set:** The set of properties to be returned MUST be determined by the repository.

2765 ○ **A comma-delimited list of property definition Query Names: The properties listed**

2766 **MUST be returned.**

2767 ○ **“*”** : All properties MUST be returned for all objects.

2768 Repositories SHOULD return only the properties specified in the property filter if they exist on the object's

2769 type definition.

2770

2771 If a property filter specifies a property that is 'not set', it MUST be represented as a property element

2772 without a value element.

2773 2.2.1.2.2 Relationships

2774 **Description:** Used to retrieve the relationships in which the object(s) are participating.

2775 **Optional Input Parameter:**

- 2776 • **Enum includeRelationships:** Value indicating what relationships in which the objects returned
- 2777 participate MUST be returned, if any. Values are:
- 2778 none : No relationships MUST be returned. (Default).
- 2779 source : Only relationships in which the objects returned are the source MUST be
- 2780 returned.
- 2781 target : Only relationships in which the objects returned are the target MUST be
- 2782 returned.
- 2783 both : Relationships in which the objects returned are the source or the target MUST be
- 2784 returned.

2785 **Output Parameter for each object:**

- 2786 • **<Array> Relationships:** A collection of the relationship objects.

2787 2.2.1.2.3 Policies

2788 **Description:** Used to retrieve the policies currently applied to the object(s).

2789 **Optional Input Parameter:**

- 2790 • **Boolean includePolicyIds:** If TRUE, then the Repository MUST return the Ids of the policies
- 2791 applied to the object. Defaults to FALSE.

2792 **Output Parameter for each object:**

- 2793 • **<Array> Policies:** A collection of the policy objects.

2794 2.2.1.2.4 Renditions

2795 **Description:** Used to retrieve the renditions of the object(s).

2796 **Optional Input Parameter:**

- 2797 • **String renditionFilter:** The Repository MUST return the set of renditions whose kind matches
- 2798 this filter. See section below for the filter grammar.
- 2799 ○ Defaults to “cmis:none”.

2800 **Output Parameter for each object:**

- 2801 • **<Array> Renditions:** The set of renditions.

2802 2.2.1.2.4.1 Rendition Filter Grammar

2803 The Rendition Filter grammar is defined as follows:

2804 <renditionInclusion> ::= <none> | <wildcard> | <termlist>
 2805 <termlist> ::= <term> | <term> ',' <termlist>
 2806 <term> ::= <kind> | <mimetype>
 2807 <kind> ::= <text>
 2808 <mimetype> ::= <type> '/' <subtype>
 2809 <type> ::= <text>
 2810 <subtype> ::= <text> | <wildcard>
 2811 <text> ::= /* any char except whitespace */
 2812 <wildcard> ::= '*'
 2813 <none> ::= 'cmis:none'

2814 An inclusion pattern allows:

- 2815 • **Wildcard** : include all associated Renditions
- 2816 • **Comma-separated list of Rendition kinds or mimetypes** : include only those Renditions
- 2817 that match one of the specified kinds or mimetypes
- 2818 • **cmis:none**: (Default) exclude all associated Renditions

2819 Examples:

- 2820 • * (include all Renditions)
- 2821 • cmis:thumbnail (include only Thumbnails)
- 2822 • Image/* (include all image Renditions)
- 2823 • application/pdf, application/x-shockwave-flash (include web ready Renditions)
- 2824 • cmis:none (exclude all Renditions)

2825 2.2.1.2.5 ACLs

2826 **Description:** Used to retrieve the ACLs for the object(s) described in the service response.

2827 **Optional Input Parameter:**

- 2828 • **Boolean includeACL:** If TRUE, then the Repository MUST return the ACLs for each object in
- 2829 the result set. Defaults to FALSE.

2830 **Output Parameter for each object:**

- 2831 • **<Array> ACLs:** The list of access control entries of the ACL for the object.

2832 2.2.1.2.6 Allowable Actions

2833 **Description:** Used to retrieve the allowable actions for the object(s) described in the service response.

2834 **Optional Input Parameter:**

- 2835 • **Boolean includeAllowableActions:** If TRUE, then the Repository MUST return the
- 2836 available actions for each object in the result set. Defaults to FALSE.

2837 **Output Parameter for each object:**

- 2838 • **AllowableActions:** See cmisAllowableActionsType in the CMIS schema.

2839 2.2.1.3 Change Tokens

2840 The CMIS base object-type definitions include an opaque string "ChangeToken" property that a
 2841 Repository MAY use for optimistic locking and/or concurrency checking to ensure that user updates do
 2842 not conflict.

2843 If a Repository provides values for the ChangeToken property for an Object, then all invocations of the
 2844 "update" methods on that object (updateProperties, setContentStream, deleteContentStream) MUST

2845 provide the value of the changeToken property as an input parameter, and the Repository MUST throw
2846 an updateConflictException if the value specified for the changeToken does NOT match the
2847 changeToken value for the object being updated.

2848 2.2.1.4 Exceptions

2849 The following sections list the complete set of exceptions that MAY be returned by a repository in
2850 response to a CMIS service method call.

2851 2.2.1.4.1 General Exceptions

2852 The following exceptions MAY be returned by a repository in response to ANY CMIS service method call.

2853 The “Cause” field indicates the circumstances under which a repository SHOULD return a particular
2854 exception.

2855 **invalidArgument**

2856 Cause: One or more of the input parameters to the service method is missing or invalid.

2857

2858 **objectNotFound**

2859 Cause: The service call has specified an object that does not exist in the Repository.

2860

2861 **notSupported**

2862 Cause: The service method invoked requires [an optional capability](#) not supported by the
2863 repository.

2864

2865 **permissionDenied**

2866 Cause: The caller of the service method does not have sufficient permissions to perform the
2867 operation.

2868

2869 **runtime**

2870 Cause: Any other cause not expressible by another CMIS exception.

2871 2.2.1.4.2 Specific Exceptions

2872 The following exceptions MAY be returned by a repository in response to one or more CMIS service
2873 methods calls.

2874 For each exception, the general intent is listed as well as a list of the methods which MAY cause the
2875 exception to be thrown.

2876 **constraint**

2877 Intent: The operation violates a Repository- or Object-level constraint defined in the CMIS
2878 domain model.

2879 Methods:

- 2880 • **Navigation Services:**

- 2881
 - getObjectParents

- 2882 • **Object Services:**

- 2883
 - createDocument

- 2884
 - createDocumentFromSource

- 2885
 - createFolder

- 2886
 - createRelationship

- 2887 ○ createPolicy
- 2888 ○ updateProperties
- 2889 ○ moveObject
- 2890 ○ deleteObject
- 2891 ○ setContentStream
- 2892 ○ deleteContentStream
- 2893 • **Multi-filing Services:**
- 2894 ○ addObjectToFolder
- 2895 • **Versioning Services:**
- 2896 ○ checkOut
- 2897 ○ cancelCheckOut
- 2898 ○ checkIn
- 2899 • **Policy Services:**
- 2900 ○ applyPolicy
- 2901 ○ removePolicy
- 2902 • **Change Log Services:**
- 2903 ○ getContentChanges

2904

2905 **contentAlreadyExists**

2906 Intent: The operation attempts to set the content stream for a Document that already has a
2907 content stream without explicitly specifying the “overwriteFlag” parameter.

2908 Methods:

- 2909 • **Object Services:**
- 2910 ○ setContentStream

2911

2912 **filterNotValid**

2913 Intent: The property filter or rendition filter input to the operation is not valid.

2914 Methods:

- 2915 • **Navigation Services:**
- 2916 ○ getDescendants
- 2917 ○ getChildren
- 2918 ○ getFolderParent
- 2919 ○ getObjectParents
- 2920 ○ getCheckedOutDocs
- 2921 • **Object Services:**
- 2922 ○ getProperties
- 2923 ○ getRenditions
- 2924 ○ getObject
- 2925 ○ getObjectByPath
- 2926 • **Versioning Services:**
- 2927 ○ getPropertiesOfLatestVersion
- 2928 ○ getAllVersions

2929 • **Policy Services:**

2930 ○ getAppliedPolicies

2931

2932 **nameConstraintViolation**

2933 Intent: The repository is not able to store the object that the user is creating/updating due to

2934 a name constraint violation.

2935 Methods:

2936 • **Object Services:**

2937 ○ createDocument

2938 ○ createDocumentFromSource

2939 ○ createFolder

2940 ○ createRelationship

2941 ○ createPolicy

2942 ○ updateProperties

2943 ○ moveObject

2944

2945 **storage**

2946 Intent: The repository is not able to store the object that the user is creating/updating due to

2947 an internal storage problem.

2948 Methods:

2949 • **Object Services:**

2950 ○ createDocument

2951 ○ createDocumentFromSource

2952 ○ createFolder

2953 ○ createRelationship

2954 ○ createPolicy

2955 ○ updateProperties

2956 ○ moveObject

2957 ○ setContentStream

2958 ○ deleteContentStream

2959 • **Versioning Services:**

2960 ○ checkOut

2961 ○ checkIn

2962

2963 **streamNotSupported**

2964 Intent: The operation is attempting to get or set a contentStream for a Document whose

2965 Object-type specifies that a content stream is not allowed for Document's of that type.

2966 Methods:

2967 • **Object Services:**

2968 ○ createDocument

2969 ○ createDocumentFromSource

2970 ○ getContentStream

2971 ○ setContentStream

2972 • **Versioning Services:**
 2973 o checkIn
 2974
 2975 **updateConflict**
 2976 Intent: The operation is attempting to update an object that is no longer current (as
 2977 determined by the repository).
 2978 Methods:
 2979 • **Object Services:**
 2980 o updateProperties
 2981 o moveObject
 2982 o deleteObject
 2983 o deleteTree
 2984 o setContentStream
 2985 o deleteContentStream
 2986 • **Versioning Services:**
 2987 o checkOut
 2988 o cancelCheckOut
 2989 o checkIn

2990
 2991 **versioning**
 2992 Intent: The operation is attempting to perform an action on [a non-current version](#) of a
 2993 Document that cannot be performed on a non-current version.
 2994 Methods:
 2995 • **Object Services:**
 2996 o updateProperties
 2997 o moveObject
 2998 o setContentStream
 2999 o deleteContentStream
 3000 • **Versioning Services:**
 3001 o checkOut
 3002 o cancelCheckOut
 3003 o checkIn

3004 2.2.1.5 ACLs

3005 Those services which allow for the setting of ACLs may take the optional macro cmis:user which allows
 3006 the caller to indicate the operation applies to the current authenticated user.

3007 2.2.2 Repository Services

3008 The Repository Services (getRepositoryInfo, getTypeChildren, getTypeDescendants,
 3009 getTypeDefinition) are used to discover information about the repository, including information about the
 3010 repository and the object-types defined for the repository.

3011 2.2.2.1 getRepositories

3012 **Description:** Returns a list of CMIS repositories available from this CMIS service endpoint.

3013 2.2.2.1.1 Inputs

3014 None.

3015 2.2.2.1.2 Outputs

3016 A list of repository information, with (at least) the following information for each entry:

- 3017 • **ID repositoryId**: The identifier for the Repository.
- 3018 • **String repositoryName**: A display name for the Repository.

3019 2.2.2.1.3 Exceptions Thrown & Conditions

3020 See section 2.2.1.4.1 General Exceptions

3021 2.2.2.2 getRepositoryInfo

3022 **Description**: Returns information about the CMIS repository, the [optional capabilities](#) it supports and its
3023 Access Control information if applicable. .

3024 2.2.2.2.1 Inputs

3025 **Required**:

- 3026 • **ID repositoryId**: The identifier for the Repository.

3027 2.2.2.2.2 Outputs

- 3028 • **ID repositoryId**: The identifier for the Repository.
 - 3029 ○ Note: This MUST be the same identifier as the input to the method.
- 3030 • **String repositoryName**: A display name for the Repository.
- 3031 • **String repositoryDescription**: A display description for the Repository.
- 3032 • **String vendorName**: A display name for the vendor of the Repository's underlying application.
- 3033 • **String productName**: A display name for the Repository's underlying application.
- 3034 • **String productVersion**: A display name for the version number of the Repository's
3035 underlying application.
- 3036 • **ID rootFolderId**: The ID of the Root Folder Object for the Repository.
- 3037 • **<List of capabilities>**: The set of values for the repository-optional capabilities specified in
3038 [section 2.1.1.1 Optional Capabilities](#)
- 3039 • **String latestChangeLogToken**: The change log token corresponding to the most recent
3040 change event for any object in the repository.
- 3041 • **String cmisVersionSupported**: A decimal that indicates what version of the CMIS
3042 specification this repository supports as specified in 2.1.1.2 Implementation Information.
- 3043 • **URI thinClientURI**: A optional repository-specific URI pointing to the repository's web
3044 interface.
- 3045 • **Boolean changesIncomplete**: Indicates whether or not the repository's change log can return
3046 all changes ever made to any object in the repository or only changes made after a particular
3047 point in time. Applicable when the repository's optional capability `capabilityChanges` is not
3048 none.
 - 3049 ○ If FALSE, then the change log can return all changes ever made to every object.
 - 3050 ○ If TRUE, then the change log includes all changes made since a particular point in time,
3051 but not all changes ever made.

- **<List of enum values>** `changesOnType`: Indicates whether changes are available for base types in the repository. Valid values are from `enumBaseObjectTypeId`s. See section 2.1.11 [Change Log](#).
 - `cmis:document`
 - `cmis:folder`
 - `cmis:policy`
 - `cmis:relationship`
- **Enum** `supportedPermissions`: specifies which types of permissions are supported.
 - `basic`: indicates that the CMIS Basic permissions are supported.
 - `repository`: Indicates that repository specific permissions are supported.
 - `both`: indicates that both CMIS basic permissions and repository specific permissions are supported.
- **Enum** `propagation`: The list of allowed values for `applyACL`, which control how non-direct ACEs are handled by the repository:
 - `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.
 - `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.
 - `repositorydetermined`: indicates that the repository uses its own mechanisms to handle non-direct ACEs when applying ACLs.
- **<Array> Permission** `permissions`: The list of repository-specific permissions the repository supports for managing ACEs (see section 2.8 Access Control).
- **<Array> PermissionMapping** `mapping`: The list of mappings for the CMIS Basic permissions to allowable actions (see section 2.8 Access Control).
- **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.
- **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.

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

2.2.2.2.3 Exceptions Thrown & Conditions

See section 2.2.1.4.1 General Exceptions

2.2.2.3 getTypeChildren

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

2.2.2.3.1 Inputs

Required:

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

Optional:

- 3095 • **String typeId:** The typeId of an Object-Type specified in the Repository.
 - 3096 ○ If specified, then the Repository MUST return all of child types of the specified type.
 - 3097 ○ If not specified, then the Repository MUST return all Base Object-Types.
- 3098 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
3099 definitions for each Object-Type returned.
 - 3100 ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.
- 3101 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3102 • **Integer skipCount:** See section 2.2.1.1 Paging.

3103 2.2.2.3.2 Outputs

- 3104 <Array> **Object-Types:** The list of child [Object-Types](#) defined for the given typeId.
- 3105 **Boolean hasMoreItems:** See section 2.2.1.1 Paging.
- 3106 **Optional:**
- 3107 **Integer numItems:** See section 2.2.1.1 Paging.

3108 2.2.2.3.3 Exceptions Thrown & Conditions

3109 See section 2.2.1.4.1 General Exceptions

3110 2.2.2.4 getTypeDescendants

3111 **Description:** Returns the set of descendant [Object-Types](#) defined for the Repository under the specified
3112 Type.

3113 Notes:

- 3114 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3115 • The order in which results are returned is repository-specific.

3116 2.2.2.4.1 Inputs

3117 Required:

- 3118 • **String repositoryId:** The identifier for the Repository.

3119 Optional:

- 3120 • **String typeId:** The typeId of an Object-Type specified in the Repository.
 - 3121 ○ If specified, then the Repository MUST return all descendant types for the specified type.
 - 3122 ○ If not specified, then the Repository MUST return all types and MUST ignore the value of
3123 the depth parameter
- 3124 • **Integer depth:** The number of levels of depth in the type hierarchy from which to return results.
3125 Valid values are:
 - 3126 ○ **1:** Return only types that are children of the type.
 - 3127 ○ **<Integer value greater than 1>:** Return only types that are children of the type and
3128 descendants up to <value> levels deep.
 - 3129 ○ **-1:** Return ALL descendant types at all depth levels in the CMIS hierarchy.
 - 3130 ○ The default value is repository specific and SHOULD be at least 2 or -1.
- 3131 • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
3132 definitions for each Object-Type returned.
 - 3133 ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

3134 2.2.2.4.2 Outputs

3135 <Array> **Object-Types:** The hierarchy of [Object-Types](#) defined for the Repository.

3136 2.2.2.4.3 Exceptions Thrown & Conditions

3137 See section 2.2.1.4.1 General Exceptions

- 3138 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
3139 an invalid depth.

3140 2.2.2.5 getTypeDefinition

3141 **Description:** Gets the definition of the specified Object-Type.Inputs

3142 2.2.2.5.1 Inputs

3143 **Required:**

- 3144 • **String repositoryId:** The identifier for the Repository.
- 3145 • **String typeId:** The typeId of an Object-Type specified in the Repository.

3146 2.2.2.5.2 Outputs

- 3147 • Object-type including all property definitions. See section 2.1.3.3 (Object-Type Property
3148 Definitions) for further details.

3149 2.2.2.5.3 Exceptions Thrown & Conditions

3150 See section 2.2.1.4.1 General Exceptions

3151 2.2.3 Navigation Services

3152 The Navigation Services (getDescendants, getChildren, getFolderParent, getObjectParents,
3153 getCheckedoutDocs), are used to traverse the folder hierarchy in a CMIS **Repository**, and to locate
3154 Documents that are checked out.

3155 2.2.3.1 getChildren

3156 **Description:** Gets the list of child objects contained in the specified folder.

3157 **Notes:**

- 3158 • If the Repository supports the optional “VersionSpecificFiling” capability, then the repository
3159 MUST return the document versions filed in the specified folder.
3160 ○ Otherwise, the latest version of the documents MUST be returned.

3161 2.2.3.1.1 Inputs

3162 **Required:**

- 3163 • **ID repositoryId:** The identifier for the Repository.
- 3164 • **ID folderId:** The identifier for the folder.

3165 **Optional:**

- 3166 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3167 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3168 • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and
3169 the ascending modifier “ASC” or the descending modifier “DESC” for each query name. A
3170 repository's handling of the orderBy input is repository-specific.

- **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the matched object if they exist on the matched object type definition and in the filter.
- **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each child object for use in constructing that object's path.

2.2.3.1.2 Outputs

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

Optional:

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

2.2.3.1.3 Exceptions Thrown & Conditions

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

2.2.3.2 getDescendants

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

Notes:

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

2.2.3.2.1 Inputs

Required:

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

Optional:

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

2.2.3.2.2 Outputs

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

2.2.3.2.3 Exceptions Thrown & Conditions

See section 2.2.1.4.1 General Exceptions

- **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter is not valid.
- **invalidArgument:** The Repository MUST throw this exception if the service is invoked with "depth = 0".
- **invalidArgument:** if the specified folder is not a folder

2.2.3.3 getFolderTree

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

Notes:

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

2.2.3.3.1 Inputs

Required:

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

Optional:

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

2.2.3.3.2 Outputs

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

2.2.3.3.3 Exceptions Thrown & Conditions

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

2.2.3.4 getFolderParent

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

2.2.3.4.1 Inputs

Required:

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

Optional:

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

3288 2.2.3.4.2 Outputs

- 3289 • **Object:** The parent folder object of the specified folder.

3290 2.2.3.4.3 Exceptions Thrown & Conditions

- 3291 • See section 2.2.1.4.1 General Exceptions
- 3292 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
3293 parameter is not valid.
- 3294 • **invalidArgument:** The Repository MUST throw this exception if the folderId input is the root
3295 folder.

3296 2.2.3.5 getObjectParents

3297 **Description:** Gets the parent folder(s) for the specified non-folder, fileable object.

3298 2.2.3.5.1 Inputs

3299 Required:

- 3300 • **ID repositoryId:** The identifier for the Repository.
- 3301 • **ID objectId:** The identifier for the object.

3302 Optional:

- 3303 • **String filter:** See section 2.2.1.2.1 Properties
- 3304 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3305 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3306 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3307 • **Boolean includeRelativePathSegment:** See section 2.1.5.3 Paths.

3308 2.2.3.5.2 Outputs

- 3309 • **<Array> ObjectResults:** A list of the parent folder(s) of the specified objects. Empty for unfilled
3310 objects or for the root folder. Each object result MUST include the following elements if they are
3311 requested:
 - 3312 ○ **<Array> Properties:** The list of properties for the object.
 - 3313 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
 - 3314 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
 - 3315 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
 - 3316 ○ **String relativePathSegment:** If includeRelativePathSegment was TRUE. See section
3317 2.1.5.3 Paths.

3318 2.2.3.5.3 Exceptions Thrown & Conditions

- 3319 • See section 2.2.1.4.1 General Exceptions
- 3320 • **constraint:** The Repository MUST throw this exception if this method is invoked on an object
3321 who Object-Type Definition specifies that it is not fileable.
- 3322 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
3323 parameter is not valid.

2.2.3.6 getCheckedOutDocs

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

2.2.3.6.1 Inputs

Required:

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

Optional:

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

2.2.3.6.2 Outputs

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

Optional:

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

2.2.3.6.3 Exceptions Thrown & Conditions

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

2.2.4 Object Services

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

2.2.4.1 createDocument

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

2.2.4.1.1 Inputs

Required:

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

Optional:

- **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-created Document Object.
 - This parameter MUST be specified if the Repository does NOT support the optional “unfiling” capability.
- **<contentStream> contentStream:** The Content Stream that MUST be stored for the newly-created Document Object. The method of passing the contentStream to the server and the encoding mechanism will be specified by each specific binding. MUST be required if the type requires it.
- **Enum versioningState:** An enumeration specifying what the versioning state of the newly-created object MUST be. If the repository does not support versioning, the repository MUST ignore the versioningState parameter. Valid values are:
 - **none:** The document MUST be created as a non-versionable document.
 - **checkedout:** The document MUST be created in the checked-out state.
 - **major (default):** The document MUST be created as a major version
 - **minor:** The document MUST be created as a minor version.
- **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document object.
- **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document 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 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is specified.

2.2.4.1.2 Outputs

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

2.2.4.1.3 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 “Document”.
 - The cmis:objectTypeId property value is NOT in the list of AllowedChildObjectTypes of the parent-folder specified by folderId.
 - The value of any of the properties violates the min/max/required/length constraints specified in the property definition in the Object-Type.

- 3404 ○ The “contentStreamAllowed” attribute of the Object-Type definition specified by the
3405 cmis:objectTypeId property value is set to “required” and no contentStream input
3406 parameter is provided.
- 3407 ○ The “versionable” attribute of the Object-Type definition specified by the
3408 cmis:objectTypeId property value is set to FALSE and a value for the versioningState
3409 input parameter is provided that is something other than “none”.
- 3410 ○ The “versionable” attribute of the Object-Type definition specified by the
3411 cmis:objectTypeId property value is set to TRUE and the value for the versioningState
3412 input parameter is provided that is “none”.
- 3413 ○ The “controllablePolicy” attribute of the Object-Type definition specified by the
3414 cmis:objectTypeId property value is set to FALSE and at least one policy is provided.
- 3415 ○ The “controllableACL” attribute of the Object-Type definition specified by the
3416 cmis:objectTypeId property value is set to FALSE and at least one ACE is provided.
- 3417 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3418 repository.
- 3419 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository
3420 detects a violation with the given cmis:name property value, the repository MAY throw this
3421 exception or chose a name which does not conflict.
- 3422 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3423 • **streamNotSupported:** The Repository MUST throw this exception if the
3424 “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId
3425 property value is set to “not allowed” and a contentStream input parameter is provided.

3426 2.2.4.2 createDocumentFromSource

3427 **Description:** Creates a document object as a copy of the given source document in the (optionally)
3428 specified location.

3429 2.2.4.2.1 Inputs

3430 Required:

- 3431 • **ID repositoryId:** The identifier for the Repository.
- 3432 • **ID sourceId:** The identifier for the source document.

3433 Optional:

- 3434 • **<Array> properties:** The property values that MUST be applied to the Object. This list of
3435 properties SHOULD only contain properties whose values differ from the source document.
- 3436 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-
3437 created Document Object.
 - 3438 ○ This parameter MUST be specified if the Repository does NOT support the optional
3439 “unfiling” capability.
- 3440 • **Enum versioningState:** An enumeration specifying what the versioing state of the newly-created
3441 object MUST be. Valid values are:
 - 3442 ○ **none:** The document MUST be created as a non-versionable document.
 - 3443 ○ **checkedout:** The document MUST be created in the checked-out state.
 - 3444 ○ **major (default):** The document MUST be created as a major version
 - 3445 ○ **minor:** The document MUST be created as a minor version.
- 3446 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
3447 object.

- 3448 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
3449 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3450 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
3451 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
3452 specified.

3453 2.2.4.2.2 Outputs

3454 **ID objectId:** The ID of the newly-created document.

3455 2.2.4.2.3 Exceptions Thrown & Conditions

- 3456 • See section 2.2.1.4.1 General Exceptions
- 3457 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
3458 met:
 - 3459 ○ The sourceId is not an Object whose baseType is "Document".
 - 3460 ○ The source document's cmis:objectId property value is NOT in the list of
3461 AllowedChildObjectIds of the parent-folder specified by folderId.
 - 3462 ○ The "versionable" attribute of the Object-Type definition specified by the
3463 cmis:objectId property value is set to FALSE and a value for the versioningState
3464 input parameter is provided that is something other than "none".
 - 3465 ○ The "versionable" attribute of the Object-Type definition specified by the
3466 cmis:objectId property value is set to TRUE and the value for the versioningState
3467 input parameter is provided that is "none".
 - 3468 ○ The "controllablePolicy" attribute of the Object-Type definition specified by the
3469 cmis:objectId property value is set to FALSE and at least one policy is provided.
 - 3470 ○ The "controllableACL" attribute of the Object-Type definition specified by the
3471 cmis:objectId property value is set to FALSE and at least one ACE is provided.
 - 3472 ○ At least one of the permissions is used in an ACE provided which is not supported by the
3473 repository.
- 3474 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository
3475 detects a violation with the given cmis:name property value, the repository MAY throw this
3476 exception or chose a name which does not conflict.
- 3477 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 3478 • **streamNotSupported:** The Repository MUST throw this exception if the
3479 "contentStreamAllowed" attribute of the Object-Type definition specified by the cmis:objectId
3480 property value is set to "not allowed" and a contentStream input parameter is provided.

3481 2.2.4.3 createFolder

3482 **Description:** Creates a folder object of the specified type in the specified location.

3483 2.2.4.3.1 Inputs

3484 Required:

- 3485 • **ID repositoryId:** The identifier for the Repository.
- 3486 • **<Array> properties:** The property values that MUST be applied to the newly-created Folder
3487 Object.
- 3488 • **ID folderId:** The identifier for the folder that MUST be the parent folder for the newly-created
3489 Folder Object.

3490 Optional:

- **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Folder object.
- **<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.
- **<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.

2.2.4.3.2 Outputs

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

2.2.4.3.3 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:objectId property value is not an Object-Type whose baseType is "Folder".
 - The value of any of the properties violates the min/max/required/length constraints specified in the property definition in the Object-Type.
 - The cmis:objectId property value is NOT in the list of AllowedChildObjectIds of the parent-folder specified by folderId.
 - The "controllablePolicy" attribute of the Object-Type definition specified by the cmis:objectId property value is set to FALSE and at least one policy is provided.
 - The "controllableACL" attribute of the Object-Type definition specified by the cmis:objectId property value is set to FALSE and at least one ACE is provided.
 - At least one of the permissions is used in an ACE provided which is not supported by the repository.
- **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.
- **storage:** See section 2.2.1.4.2 Specific Exceptions.

2.2.4.4 createRelationship

Description: Creates a relationship object of the specified type

2.2.4.4.1 Inputs

Required:

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

Optional:

- **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Relationship object.
- **<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.
- **<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.

2.2.4.4.2 Outputs

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

2.2.4.4.3 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:objectId property value is not an Object-Type whose baseType is "Relationship".
 - The value of any of the properties violates the min/max/required/length constraints specified in the property definition in the Object-Type.
 - The sourceObjectId's ObjectType is not in the list of "allowedSourceTypes" specified by the Object-Type definition specified by cmis:objectId property value.
 - The targetObjectId's ObjectType is not in the list of "allowedTargetTypes" specified by the Object-Type definition specified by cmis:objectId property value.
 - The "controllablePolicy" attribute of the Object-Type definition specified by the cmis:objectId property value is set to FALSE and at least one policy is provided.
 - The "controllableACL" attribute of the Object-Type definition specified by the cmis:objectId property value is set to FALSE and at least one ACE is provided.
 - At least one of the permissions is used in an ACE provided which is not supported by the repository.
- **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.
- **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:objectId property value is not an Object-Type whose baseType is "Policy".
 - The value of any of the properties violates the min/max/required/length constraints specified in the property definition in the Object-Type.
 - The cmis:objectId property value is NOT in the list of AllowedChildObjectIds of the parent-folder specified by folderId.
 - The "controllablePolicy" attribute of the Object-Type definition specified by the cmis:objectId property value is set to FALSE and at least one policy is provided.
 - The "controllableACL" attribute of the Object-Type definition specified by the cmis:objectId property value is set to FALSE and at least one ACE is provided.
 - At least one of the permissions is used in an ACE provided which is not supported by the repository.
- **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.
- **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

- 3612 **Optional:**
- 3613 • **String filter:** See section 2.2.1.2.1 Properties.
 - 3614 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
 - 3615 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
 - 3616 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
 - 3617 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
 - 3618 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3619 **2.2.4.7.2 Outputs**

- 3620 **<Array> Properties:** The list of properties for the object.
- 3621 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3622 **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 3623 **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3624 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 3625 **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3626 **2.2.4.7.3 Exceptions Thrown & Conditions**

- 3627 See section 2.2.1.4.1 General Exceptions
- 3628 **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter
- 3629 is not valid.

3630 **2.2.4.8 getProperties**

- 3631 **Description:** Gets the list of properties for an Object.

3632 **2.2.4.8.1 Inputs**

- 3633 **Required:**
- 3634 • **ID repositoryId:** The identifier for the Repository.
 - 3635 • **ID objectId:** The identifier for the object

- 3636 **Optional:**
- 3637 • **String filter:** See section 2.2.1.2.1 Properties.

3638 **2.2.4.8.2 Outputs**

- 3639 **<Array> Properties:** The list of properties for the object.

3640 **2.2.4.8.3 Exceptions Thrown & Conditions**

- 3641 See section 2.2.1.4.1 General Exceptions
- 3642 **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter
- 3643 is not valid.

3644 **2.2.4.9 getObjectByPath**

- 3645 **Description:** Gets the specified object.

2.2.4.9.1 Inputs

Required:

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

Optional:

- **String filter:** See section 2.2.1.2.1 Properties.
- **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- **Boolean includePolicyIds:** See section 2.2.1.2.2 Relationships.
- **Boolean includeACL:** See section 2.2.1.2.5 ACLs.

2.2.4.9.2 Outputs

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

AllowableActions: See section 2.2.1.2.6 Allowable Actions.

2.2.4.9.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions

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

2.2.4.10 getContentStream

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

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

2.2.4.10.1 Inputs

Required:

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

Optional:

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

2.2.4.10.2 Outputs

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

2.2.4.10.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions

constraint: The Repository MUST throw this exception if the object specified by objectId does NOT have a content stream or rendition stream.

2.2.4.11 getRenditions

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

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

2.2.4.11.1 Inputs

Required:

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

Optional:

- **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.

2.2.4.11.2 Outputs

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

2.2.4.11.3 Exceptions Thrown & Conditions

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

2.2.4.12 updateProperties

Description: Updates properties of the specified object.

Notes:

- 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.
- 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.

2.2.4.12.1 Inputs

Required:

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

Optional:

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

2.2.4.12.2 Outputs

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

2.2.4.12.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions
- **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.
- **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY throw this exception or chose a name which does not conflict.
- **storage:** See section 2.2.1.4.2 Specific Exceptions.
- **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- **versioning:** The Repository MUST throw this exception if ANY of the following conditions are met:
 - 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.
 - Additionally, the repository MAY throw this exception if the object is a non-current Document Version.

2.2.4.13 moveObject

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

2.2.4.13.1 Inputs

Required:

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

2.2.4.13.2 Outputs

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

2.2.4.13.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions
- **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.).
- **constraint:** The Repository MUST throw this exception if the cmis:objectTypeId property value of the given object is NOT in the list of AllowedChildObjectTypelds of the parent-folder specified by targetFolderId.
- **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. The repository MAY throw this exception or chose a name which does not conflict.
- **storage:** See section 2.2.1.4.2 Specific Exceptions.
- **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- **versioning:** The repository MAY throw this exception if the object is a non-current Document Version.

2.2.4.14 deleteObject

Description: Deletes the specified object.

2.2.4.14.1 Inputs

Required:

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

Optional:

- **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 invoked on a non-document object or non-versionable document object.

2.2.4.14.2 Exceptions Thrown & Conditions

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

2.2.4.15 deleteTree

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

Notes:

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

2.2.4.15.1 Inputs

Required:

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

Optional:

- **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 invoked on a non-document object or non-versionable document object.
- **Enum unfileObjects:** An enumeration specifying how the repository MUST process fileable child- or descendant-objects. Valid values are:
 - **unfile:** Unfile all fileable objects.

- 3800 ○ **deletesinglefiled**: Delete all fileable non-folder objects whose only parent-folders are in
- 3801 the current folder tree. Unfile all other fileable non-folder objects from the current folder tree.
- 3802 ○ **delete (default)**: Delete all fileable objects.
- 3803 • **boolean continueOnFailure**: If TRUE, then the repository SHOULD continue attempting to
- 3804 perform this operation even if deletion of a child- or descendant-object in the specified folder
- 3805 cannot be deleted.
- 3806 ○ If FALSE (**default**), then the repository SHOULD abort this method when it fails to delete a
- 3807 single child- or descendant-object.

3808 2.2.4.15.2 Outputs

- 3809 • **<Array> ID failedToDelete**: A list of identifiers of objects in the folder tree that were not deleted.

3810 2.2.4.15.3 Exceptions Thrown & Conditions

- 3811 • See section 2.2.1.4.1 General Exceptions
- 3812 • **updateConflict**: See section 2.2.1.4.2 Specific Exceptions.

3813 2.2.4.16 setContentStream

3814 **Description**: Sets the content stream for the specified Document object.

3815 **Notes**: A Repository MAY automatically create new Document versions as part of this service method.
 3816 Therefore, the objectId output NEED NOT be identical to the objectId input.

3817 2.2.4.16.1 Inputs

3818 **Required**:

- 3819 • **ID repositoryId**: The identifier for the Repository.
- 3820 • **ID objectId**: The identifier for the Document object.
- 3821 • **<contentStream> contentStream**: The Content Stream

3822 **Optional**:

- 3823 • **Boolean overwriteFlag**: If TRUE (**default**), then the Repository MUST replace the existing
- 3824 content stream for the object (if any) with the input contentStream.
- 3825 ○ If FALSE, then the Repository MUST only set the input contentStream for the object if the
- 3826 object currently does not have a content-stream.
- 3827 • **String changeToken**: See section 2.2.1.3 Change Tokens.

3828 2.2.4.16.2 Outputs

- 3829 • **ID objectId**: The ID of the document.
- 3830 • **String changeToken**: See section 2.2.1.3 Change Tokens.

3831 2.2.4.16.3 Exceptions Thrown & Conditions

- 3832 • See section 2.2.1.4.1 General Exceptions
- 3833 • **contentAlreadyExists**: The Repository MUST throw this exception if the input parameter
- 3834 overwriteFlag is FALSE and the Object already has a content-stream.
- 3835 • **storage**: See section 2.2.1.4.2 Specific Exceptions.
- 3836 • **streamNotSupported**: The Repository MUST throw this exception if the
- 3837 “contentStreamAllowed” attribute of the Object-Type definition specified by the cmis:objectTypeId
- 3838 property value of the given document is set to “notallowed”.

- `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.
- `versioning`: The repository MAY throw this exception if the object is a non-current Document Version.

2.2.4.17 deleteContentStream

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

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

2.2.4.17.1 Inputs

Required:

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

Optional:

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

2.2.4.17.2 Outputs

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

2.2.4.17.3 Exceptions Thrown & Conditions

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

2.2.5 Multi-filing Services

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

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

2.2.5.1 addObjectToFolder

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

2.2.5.1.1 Inputs

Required:

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

Optional:

- 3876 • **Boolean allVersions:** Add all versions of the object to the folder if the repository supports
3877 version-specific filing. Defaults to TRUE.

3878 2.2.5.1.2 Exceptions Thrown & Conditions

- 3879 • See section 2.2.1.4.1 General Exceptions.
- 3880 • **constraint:** The Repository MUST throw this exception if the `cmis:objectTypeId` property value
3881 of the given object is NOT in the list of `AllowedChildObjectTypes` of the parent-folder specified
3882 by `folderId`.

3883 2.2.5.2 removeObjectFromFolder

3884 **Description:** Removes an existing fileable non-folder object from a folder.

3885 2.2.5.2.1 Inputs

3886 **Required:**

- 3887 • **ID repositoryId:** The identifier for the Repository.
- 3888 • **ID objectId:** The identifier of the object.

3889 **Optional:**

- 3890 • **ID folderId:** The folder from which the object is to be removed.
- 3891 ◦ If no value is specified, then the Repository MUST remove the object from all folders in which
3892 it is currently filed.

3893 2.2.5.2.2 Exceptions Thrown & Conditions

- 3894 • See section 2.2.1.4.1 General Exceptions

3895 2.2.6 Discovery Services

3896 The Discovery Services (*query*) are used to search for query-able objects within the Repository.

3897 2.2.6.1 query

3898 **Description:** Executes a CMIS query statement against the contents of the Repository.

3899 2.2.6.1.1 Inputs

3900 **Required:**

- 3901 • **ID repositoryId:** The identifier for the Repository.
- 3902 • **String statement:** CMIS query to be executed. (See section 2.1.10 Query.)

3903 **Optional:**

- 3904 • **Boolean searchAllVersions:**
- 3905 ◦ If TRUE, then the Repository MUST include latest and non-latest versions of document
3906 objects in the query search scope.
- 3907 ◦ If FALSE (**default**), then the Repository MUST only include latest versions of documents
3908 in the query search scope.
- 3909 ◦ If the Repository does not support the optional `capabilityAllVersionsSearchable`
3910 capability, then this parameter value MUST be set to FALSE.
- 3911 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3912 ◦ Note: For query statements where the SELECT clause contains properties from only one
3913 virtual table reference (i.e. referenced object-type), any value for this enum may be used.

3914 If the SELECT clause contains properties from more than one table, then the value of this
3915 parameter MUST be "none" .

3916 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.

3917 ○ If the SELECT clause contains properties from more than one table, then the value of this
3918 parameter MUST not be set.

3919 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3920 ○ Note: For query statements where the SELECT clause contains properties from only one
3921 virtual table reference (i.e. referenced object-type), any value for this parameter may be
3922 used. If the SELECT clause contains properties from more than one table, then the value
3923 of this parameter MUST be "FALSE" .

3924 • **Integer maxItems:** See section 2.2.1.1 Paging.

3925 • **Integer skipCount:** See section 2.2.1.1 Paging.

3926 2.2.6.1.2 Outputs

3927 • **<Array> Object QueryResults:** The set of results for the query. (See section 2.1.10 Query.).
3928 Each object result MUST include the following elements if they are requested:

3929 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.

3930 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.

3931 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3932 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3933 **Optional:**

3934 • **Integer numItems:** See section 2.2.1.1 Paging.

3935

3936 2.2.6.1.3 Exceptions Thrown & Conditions

3937 • See section 2.2.1.4.1 General Exceptions

3938 • If the select clause includes properties from more than a single type reference, then the
3939 repository SHOULD throw an exception if includeRelationships is something other than "none" or
3940 includeAllowableActions is specified as TRUE.

3941 2.2.6.2 getContentChanges

3942 **Description:** Gets a list of content changes. This service is intended to be used by search crawlers or
3943 other applications that need to efficiently understand what has changed in the repository.

3944 **Notes:**

3945 • The content stream is NOT returned for any change event.

3946 • The definition of the authority needed to call this service is repository specific.

3947 • The latest change log token for a repository can be acquired via the getRepositoryInfo service.

3948 2.2.6.2.1 Inputs

3949 **Required:**

3950 • **ID repositoryId:** The identifier for the Repository.

3951 **Optional:**

3952 • **String changeLogToken:**

- 3953 ○ If specified, then the Repository MUST return the change event corresponding to the value of
3954 the specified change log token as the first result in the output.
- 3955 ○ If not specified, then the Repository MUST return the first change event recorded in the
3956 change log.
- 3957 • **Boolean includeProperties:**
- 3958 ○ If TRUE, then the Repository MUST include the updated property values for “updated”
3959 change events if the repository supports returning property values as specified by
3960 capabilityChanges.
- 3961 ○ If FALSE (default), then the Repository MUST NOT include the updated property values for
3962 “updated” change events. The single exception to this is that the objectId MUST always be
3963 included.
- 3964 • **Boolean includePolicyIds:**
- 3965 If TRUE, then the Repository MUST include the IDs of Policies applied to the object referenced in
3966 each change event, if the change event modified the set of policies applied to the object.
- 3967 If FALSE (default), then the Repository will not include policy information.
- 3968 • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the
3969 matched object if they exist on the matched object type definition and in the filter.
- 3970 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 3971 • **Integer maxItems:** See section 2.2.1.1 Paging.

3972 2.2.6.2.2 Outputs

- 3973 • **<Array> changeEvents:** A collection of CMIS objects that MUST include the information [as](#)
3974 [specified in](#) 2.1.11.3. Each result MUST include the following elements if they are requested:
- 3975 ○ **<Array> policyIDs:** The IDs of Policies applied to the object referenced in the change event.
- 3976 ○ **<Array> ACLs:** The ACLs applied to the object reference in the change event.
- 3977 • **String latestChangeLogToken:** The change log token corresponding to the last change event in
3978 changeEvents.
- 3979 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3980 Optional:

- 3981 • **Integer numItems:** See section 2.2.1.1 Paging.

3982 2.2.6.2.3 Exceptions Thrown & Conditions

- 3983 • See section 2.2.1.4.1 General Exceptions
- 3984 • **constraint:** The Repository MUST throw this exception if the event corresponding to the
3985 change log token provided as an input parameter is no longer available in the change log. (E.g.
3986 because the change log was truncated).

3987 2.2.7 Versioning Services

3988 The Versioning services (checkOut, cancelCheckOut, getPropertiesOfLatestVersion, getAllVersions,
3989 deleteAllVersions) are used to navigate or update a Document Version Series.

3990 2.2.7.1 checkOut

3991 **Description:** Create a private working copy of the document.

3992 2.2.7.1.1 Inputs

3993 Required:

- 3994 • **ID repositoryId:** The identifier for the Repository.
- 3995 • **ID objectId:** The identifier of the document version.

3996 2.2.7.1.2 Outputs

- 3997 • **ID objectId:** The identifier for the “Private Working Copy” document.
- 3998 • **Boolean contentCopied:** TRUE if the content-stream of the Private Working Copy is a copy of
3999 the contentStream of the Document that was checked out.
 - 4000 ○ FALSE if the content-stream of the Private Working Copy is “not set”.

4001 2.2.7.1.3 Exceptions Thrown & Conditions

- 4002 • See section 2.2.1.4.1 General Exceptions
- 4003 • **constraint:** The Repository MUST throw this exception if the Document’s Object-Type
4004 definition’s *versionable* attribute is FALSE.
- 4005 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 4006 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 4007 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
4008 Version.

4009 2.2.7.2 cancelCheckOut

4010 **Description:** Reverses the effect of a check-out. Removes the private working copy of the checked-out
4011 document, allowing other documents in the version series to be checked out again.

4012 2.2.7.2.1 Inputs

4013 Required:

- 4014 • **ID repositoryId:** The identifier for the Repository.
- 4015 • **ID objectId:** The identifier of the Private Working Copy.

4016 2.2.7.2.2 Exceptions Thrown & Conditions

- 4017 • See section 2.2.1.4.1 General Exceptions
- 4018 • **constraint:** The Repository MUST throw this exception if the Document’s Object-Type
4019 definition’s *versionable* attribute is FALSE.
- 4020 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 4021 • **versioning:** The repository MAY throw this exception if the object is a non-current Document
4022 Version.

4023 2.2.7.3 checkIn

4024 **Description:** Checks-in the Private Working Copy document.

4025 Notes:

- 4026 • For repositories that do NOT support the optional “*capabilityPWCUpdatable*” *capability*, the
4027 *properties* and *contentStream* input parameters MUST be provided on the checkIn method for
4028 updates to happen as part of checkIn.

- 4029 • Each CMIS protocol bindings MUST specify whether the checkin service MUST always include all
4030 updatable properties, or only those properties whose values are different than the original value
4031 of the object.

4032 2.2.7.3.1 Inputs

4033 Required:

- 4034 • **ID repositoryId:** The identifier for the Repository.
4035 • **ID objectId:** The identifier of the document.

4036 Optional:

- 4037 • **Boolean major:** TRUE (**default**) if the checked-in Document Object MUST be a major version.
4038 ◦ FALSE if the checked-in Document Object MUST NOT be a major version.
4039 • **<Array> properties:** The property values that MUST be applied to the checked-in Document
4040 Object.
4041 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the checked-in
4042 Document Object. The method of passing the contentStream to the server and the encoding
4043 mechanism will be specified by each specific binding.
4044 • **String checkinComment:** See section 2.1.9.5 Versioning Properties on Document Objects.
4045 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document
4046 object.
4047 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document
4048 object.
4049 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created
4050 Document object.

4051 2.2.7.3.2 Outputs

4052 **ID objectId:** The ID of the checked-in document.

4053 2.2.7.3.3 Exceptions Thrown & Conditions

- 4054 • See section 2.2.1.4.1 General Exceptions
4055 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type
4056 definition's *versionable* attribute is FALSE.
4057 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
4058 • **streamNotSupported:** The Repository MUST throw this exception if the
4059 "contentStreamAllowed" attribute of the Object-Type definition specified by the cmis:objectTypeId
4060 property value is set to "not allowed" and a contentStream input parameter is provided.
4061 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

4062 2.2.7.4 getObjectOfLatestVersion

4063 **Description:** Get a the latest Document object in the Version Series.

4064 2.2.7.4.1 Inputs

4065 Required:

- 4066 • **ID repositoryId:** The identifier for the Repository.
4067 • **ID objectId:** The identifier for the Version Series.

4068 Optional:

- 4069 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major
4070 version object in the Version Series.
 - 4071 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
4072 major) version object in the Version Series.
- 4073 • **String filter:** See section 2.2.1.2.1 Properties.
- 4074 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4075 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 4076 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4077 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4078 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4079 2.2.7.4.2 Outputs

- 4080 • **<Array> Properties:** The list of properties for the object.
- 4081 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 4082 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 4083 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 4084 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 4085 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4086 2.2.7.4.3 Exceptions Thrown & Conditions

- 4087 • See section 2.2.1.4.1 General Exceptions
- 4088 • **filterNotValid:** The Repository MUST throw this exception if this property filter input
4089 parameter is not valid.
- 4090 • **objectNotFound:** The Repository MUST throw this exception if the input parameter major is
4091 TRUE and the Version Series contains no major versions.

4092 2.2.7.5 getPropertiesOfLatestVersion

4093 **Description:** Get a subset of the properties for the latest Document Object in the Version Series.

4094 2.2.7.5.1 Inputs

4095 **Required:**

- 4096 • **ID repositoryId:** The identifier for the Repository.
- 4097 • **ID objectId:** The identifier for the Version Series.

4098 **Optional:**

- 4099 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major
4100 version object in the Version Series.
 - 4101 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-
4102 major) version object in the Version Series.
- 4103 • **String filter:** See section 2.2.1.2.1 Properties.

4104 2.2.7.5.2 Outputs

- 4105 • **<Array> Properties:** The list of properties for the object.

2.2.7.5.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions
- `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter is not valid.
- `objectNotFound`: The Repository MUST throw this exception if the input parameter major is TRUE and the Version Series contains no major versions.

2.2.7.6 getAllVersions

Description: Returns the list of all Document Objects in the specified Version Series, sorted by `cmis:creationDate` descending.

Notes:

- The result set for this operation MUST include the Private Working Copy, subject to caller's access privileges.

2.2.7.6.1 Inputs

Required:

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

Optional:

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

2.2.7.6.2 Outputs

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

2.2.7.6.3 Exceptions Thrown & Conditions

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

2.2.8 Relationship Services

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

2.2.8.1 getObjectRelationships

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

2.2.8.1.1 Inputs

Required:

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

4143 • **ID objectId**: The identifier of the object.

4144

4145 **Optional:**

4146 • **Boolean includeSubRelationshipTypes**: If TRUE, then the Repository MUST return all
4147 relationships whose Object-Types are descendant-types of the given object's cmis:objectTypeId
4148 property value as well as relationships of the specified type.

4149 ○ Default is FALSE

4150 ○ If FALSE, then the Repository MUST only return relationships whose Object-Type is
4151 equivalent to the given object's cmis:objectTypeId property value.

4152 • **Enum relationshipDirection**: An enumeration specifying whether the Repository MUST
4153 return relationships where the specified Object is the source of the relationship, the target of the
4154 relationship, or both. Valid values are:

4155 ○ **source: (default)** The Repository MUST return only relationship objects where the specified
4156 object is the source object.

4157 ○ **target**: The Repository MUST return only relationship objects where the specified object is
4158 the target object.

4159 ○ **either**: The Repository MUST return relationship objects where the specified object is
4160 either the source or the target object.

4161 • **ID typeId**: If specified, then the Repository MUST return only relationships whose Object-Type is
4162 of the type specified

4163 ○ If not specified, then the repository MUST return Relationship objects of all types.

4164 • **Integer maxItems**: See section 2.2.1.1 Paging.

4165 • **Integer skipCount**: See section 2.2.1.1 Paging.

4166 • **String filter**: See section 2.2.1.2.1 Properties.

4167 • **Boolean includeAllowableActions**: See section 2.2.1.2.6 Allowable Actions.

4168 2.2.8.1.2 Outputs

4169 • **<Array> Objects**: A list of the relationship objects. Each object result MUST include the following
4170 elements if they are requested:

4171 ○ **<Array> Properties**: The list of properties for the object.

4172 ○ **AllowableActions**: See section 2.2.1.2.6 Allowable Actions.

4173 • **Boolean hasMoreItems**: See section 2.2.1.1 Paging.

4174 **Optional:**

4175 • **Integer numItems**: See section 2.2.1.1 Paging.

4176

4177 2.2.8.1.3 Exceptions Thrown & Conditions

4178 • See section 2.2.1.4.1 General Exceptions

4179 • **filterNotValid**: The Repository MUST throw this exception if this property filter input
4180 parameter is not valid.

4181 2.2.9 Policy Services

4182 The Policy Services (*applyPolicy*, *removePolicy*, *getAppliedPolicies*) are used to apply or remove a policy
4183 object to a controllablePolicy object.

4184 2.2.9.1 applyPolicy

4185 **Description:** Applies a specified policy to an object.

4186 2.2.9.1.1 Inputs

4187 **Required:**

- 4188 • **ID repositoryId:** The identifier for the Repository.
- 4189 • **ID policyId:** The identifier for the Policy to be applied.
- 4190 • **ID objectId:** The identifier of the object.

4191 2.2.9.1.2 Exceptions Thrown & Conditions

4192 See section 2.2.1.4.1 General Exceptions

4193 **constraint :** The Repository MUST throw this exception if the specified object's Object-Type
4194 definition's attribute for *controllablePolicy* is FALSE.

4195 2.2.9.2 removePolicy

4196 **Description:** Removes a specified policy from an object.

4197 2.2.9.2.1 Inputs

4198 **Required:**

- 4199 • **ID repositoryId:** The identifier for the Repository.
- 4200 • **ID policyId:** The identifier for the Policy to be removed.
- 4201 • **ID objectId:** The identifier of the object.

4202 2.2.9.2.2 Exceptions Thrown & Conditions

- 4203 • See section 2.2.1.4.1 General Exceptions
- 4204 • **constraint:** The Repository MUST throw this exception if the specified object's Object-Type
4205 definition's attribute for *controllablePolicy* is FALSE.

4206 2.2.9.3 getAppliedPolicies

4207 **Description:** Gets the list of policies currently applied to the specified object.

4208 2.2.9.3.1 Inputs

4209 **Required:**

- 4210 • **ID repositoryId:** The identifier for the Repository.
- 4211 • **ID objectId:** The identifier of the object.

4212 **Optional:**

4213 **String filter:** See section 2.2.1.2.1 Properties.

4214 2.2.9.3.2 Outputs

4215 **<Array> Objects:** A list of Policy Objects.

4216 2.2.9.3.3 Exceptions Thrown & Conditions

- 4217 • See section 2.2.1.4.1 General Exceptions

- **filterNotValid:** The Repository MUST throw this exception if this property filter input parameter is not valid.

2.2.10 ACL Services

2.2.10.1 getACL

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

2.2.10.1.1 Inputs

Required:

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

Optional:

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

2.2.10.1.2 Outputs

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

Optional:

- **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object – i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.

2.2.10.1.3 Exceptions Thrown & Conditions

- See section 2.2.1.4.1 General Exceptions

2.2.10.1.4 Notes

This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=discover* or *=manage*.

How an ACL for the object is computed is up to the repository. A client MUST NOT assume that the ACEs from the ACL as returned by this service can be applied via *applyACL*.

2.2.10.2 applyACL

Description: Adds or removes the given ACEs to or from the ACL of document or folder object.

2.2.10.2.1 Inputs

Required:

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

Optional:

- **<Array> AccessControlEntryType addACEs:** The ACEs to be added.
- **<Array> AccessControlEntryType removeACEs:** The ACEs to be removed.

- 4255 • **Enum ACLPropagation:** Specifies how ACEs should be handled:
- 4256 ○ **objectonly:** ACEs must be applied without changing the ACLs of other objects.
- 4257 ○ **propagate:** ACEs must be applied by propagate the changes to all “inheriting” objects.
- 4258 ○ **repositorydetermined:** **Default value.** Indicates that the client leaves the behavior to
- 4259 the repository.

4260 **2.2.10.2.2 Outputs**

- 4261 • **<Array> AccessControlEntryType:** The list of access control entries of the resulting ACL for the
- 4262 object

4263 **Optional:**

- 4264 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
- 4265 i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.
- 4266 • **String changeToken:** See section 2.2.1.3 Change Tokens.

4267 **2.2.10.2.3 Exceptions Thrown & Conditions**

- 4268 • See section 2.2.1.4.1 General Exceptions
- 4269 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
- 4270 met:
- 4271 ○ The specified object’s Object-Type definition’s attribute for *controllableACL* is FALSE.
- 4272 ○ The value for *ACLPropagation* does not match the values as returned via
- 4273 *getACLCapabilities*.
- 4274 ○ At least one of the specified values for *permission* in ANY of the ACEs *does not match*
- 4275 ANY of the *permissionNames* as returned by *getACLCapability* and *is not a CMIS Basic*
- 4276 *permission*

4277 **2.2.10.2.4 Notes**

4278 This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=manage*.

4279 How ACEs are added or removed to or from the object is up to the repository – with respect to the

4280 *ACLPropagation* provided by the client. For “shared” ACEs (e.g. via inheritance), the repository MAY

4281 merge the ACEs provided with the ACEs of the ACL already applied to the object (i.e. the ACEs provided

4282 MAY not be completely added or removed from the effective ACL for the object).

4283

4284

3 Restful AtomPub Binding

3.1 Overview

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

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

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

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

3.1.1 Namespaces

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

- CMIS-Core: <http://docs.oasis-open.org/ns/cmis/core/200908/>
 - Prefix: cmis
- CMIS-RestAtom: <http://docs.oasis-open.org/ns/cmis/restatom/200908/>
 - Prefix: cmisra
- Atom : <http://www.w3.org/2005/Atom>
 - Prefix: atom
- AtomPub: <http://www.w3.org/2007/app>
 - Prefix: app

3.1.2 Authentication

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

3.1.3 Response Formats

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

3.1.4 Optional Arguments

The binding supports adding optional parameters to CMIS resources to modify the default behavior. CMIS implementations MUST support arguments being specified as HTTP query string parameters. Names and valid values for HTTP query string parameters are as described in the appropriate CMIS Service descriptions [see CMIS Domain Model]. Valid values of enumeration types are also represented in the CMIS Core XML Schema

3.1.5 Errors and Exceptions

Exceptions MUST be mapped to the appropriate HTTP status code. Repositories SHOULD provide sufficient information in the body of the HTTP response for a user to determine corrective action. See Section 3.2.4 HTTP Status Codes for more information.

3.1.6 Renditions

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

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

The following attributes MAY be included

- title: The Filename (or name property if object) of Rendition
- length: The length of the rendition

3.1.7 Content Streams

The content stream for a document SHOULD be referenced by the content src attribute as well as the edit-media link relation.

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

The following attributes SHOULD be included on the link element:

- href: URI to the content stream
- type: The Media Type of the content stream

3.1.8 Paging of Feeds

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

If the repository can include the number of items (numItems in CMIS Domain Model) in a feed, then the repository SHOULD include the cmisra:numItems extension element in the feed.

3.1.9 Services not Exposed

The following services are not exposed in this binding:

- getRenditions: This is exposed as part of getObject
- getProperties: This is exposed as part of getObject

- 4362 • createDocumentFromSource: This is not exposed in this binding except as the client saving the
4363 resource and resubmitting it without the cmis:objectId.
- 4364 • Setting ACL on Create or Checkin operations
 - 4365 ○ This is currently not possible with the REST binding. The Create or Checkin operation
 - 4366 must be performed first. Then the dependent resource, ACL, must be retrieved and
 - 4367 updated.
- 4368 • setContentStream: This does not return the new object id and change token as specified by the
4369 domain model. This is not possible without introducing a new HTTP header.
- 4370 • deleteContentStream: This does not return the new object id and change token as specified by
4371 the domain model. This is not possible without introducing a new HTTP header.
- 4372 • checkOut: This does not return whether or not content was copied. This is not possible without
4373 introducing a new HTTP header.

4374 3.1.9.1 removePolicy

4375 This service is exposed from the domain model in the RESTful Atom Binding. However, it is not as
4376 straightforward. To remove a policy from an object, one must do:

- 4377 • Get the object.
- 4378 • Fetch the policies collection of the object.
- 4379 • Walk through the feed and find the policy object where cmis:objectId == policy id to remove.
- 4380 • Get the self lin of this policy object.
- 4381 • Perform a DELETE on this URL.

4382

4383 This is also the only case in the RESTful Atom Binding where an URI in a collection (policies) is specific
4384 to that collection.

4385 3.2 HTTP

4386 3.2.1 Entity Tag

4387 CMIS changeTokens are represented as Entity Tags and follow HTTP's use of Entity Tags. CMIS server
4388 implementations SHOULD support Entity Tags. ChangeTokens are also provided as properties and
4389 SHOULD be provided when the object is included inside an atom entry or feed.

4390 3.2.2 HTTP Range

4391 Repositories MAY support HTTP Range requests on Content Streams.

4392 3.2.3 HTTP OPTIONS Method

4393 The repository MAY support the HTTP OPTIONS method on all the resources defined in this
4394 specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP
4395 methods specified for that resource in the Allow header.

4396 3.2.4 HTTP Status Codes

4397 Please see the HTTP specification for more information on the HTTP status codes. These are provided
4398 as guidance from the HTTP specification. If any conflict arises, the HTTP specification is authoritative.

4399 3.2.4.1 General CMIS Exceptions

4400 The following listing defines the HTTP status codes that repositories MUST return for the various common
4401 exceptions defined in CMIS Domain Model.

4402	CMIS Services Exception	HTTP Status Code
4403	invalidArgument	400
4404	objectNotFound	404
4405	permissionDenied	403
4406	notSupported	405
4407	runtime	500
4408	constraint	409
4409	filterNotValid	400
4410	streamNotSupported	403
4411	storage	500
4412	contentAlreadyExists	409
4413	versioning	409
4414	updateConflict	409
4415	nameConstraintViolation	409
4416		

3.2.4.2 Notable HTTP Status Codes

- 415 Unsupported Media Type
 - When a document is POST'ed to a collection that does not support the media type of the document, this status code MUST be returned
- 422 Unprocessable Entity (Defined in RFC4918 Section 11.2)
 - When a request has been POST'ed but cannot be processed, this status code MUST be returned

Please see RFC2616 Section 10 for more information.

3.3 Media Types

CMIS introduces new media types for:

- a CMIS Query document (application/cmismquery+xml)
- a CMIS AllowableActions document (application/cmismallowableactions+xml)
- an Atom Document (Entry or Feed) with any CMIS Markup (application/cmismatom+xml)
- an Atom Feed Document with CMIS Hierarchy extensions (application/cmismtree+xml)
- a CMIS ACL Document (application/cmismacl+xml)

In addition to those media types specified by CMIS, CMIS also leverages these media types:

- AtomPub Service (application/atomsvc+xml)
- Atom Entry (application/atom+xml;type=entry)
- Atom Feed (application/atom+xml;type=feed)

3.3.1 CMIS Atom

Media Type: application/cmismatom+xml

Starting tag: atom:feed or atom:entry

4442 Type Parameters:

- 4443 • type – the semantics of the type parameter MUST be the same as the media type parameter for
4444 atom documents.

4445

4446 This allows clients to differentiate between repositories that require atom media type with CMIS
4447 extensions (application/cmisatom+xml) for creation and repositories that allow generic atom media type
4448 without CMIS extensions (application/atom+xml).

4449

4450 This is only used for CMIS repositories to advertise what media types are accepted for adding to a
4451 collection (e.g., creating resources in a collection). As such CMIS does not require specifying whether an
4452 atom feed has CMIS markup. It is included to be consistent with the Atom media type.

4453

4454 All feeds and entries from a CMIS repository MUST utilize the atom media type for exposing Atom
4455 resources. Please see the individual resources for more information on the media type. This provides
4456 the interoperability with Atom clients.

4457

4458 Example:

4459

```
4460 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4461 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4462 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4463 xmlns:atom="http://www.w3.org/2005/Atom"
4464 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4465 open.org/ns/cmis/restatom/200908/">
4466   <atom:author>
4467     <atom:name>Al Brown</atom:name>
4468   </atom:author>
4469   <atom:id>urn:uuid:efe0542e-8933-4b3e-93f2-4d1caa3fc2d9</atom:id>
4470   <atom:title type="text">CMIS Example Document</atom:title>
4471   <atom:updated>2010-01-25T10:20:58.318-08:00</atom:updated>
4472   <atom:content type="text">some text</atom:content>
4473   <cmisra:object>
4474     <cmis:properties>
4475       <cmis:propertyId localName="rep-cmis:objectTypeId"
4476 propertyDefinitionId="cmis:objectTypeId">
4477         <cmis:value>invoice</cmis:value>
4478       </cmis:propertyId>
4479       <cmis:propertyString localName="rep-cmis:name"
4480 propertyDefinitionId="cmis:name">
4481         <cmis:value>CMIS Example Document</cmis:value>
4482       </cmis:propertyString>
4483     </cmis:properties>
4484   </cmisra:object>
4485 </atom:entry>
4486
```

4487 3.3.2 CMIS Query

4488 Media Type: application/cmisquery+xml

4489 Starting tag: cmis:query

4490

4491 This document contains the representation of a query to be executed in a CMIS repository.

4492

4493 Example:

```

4494 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4495 <cmis:query xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4496 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4497 xmlns:atom="http://www.w3.org/2005/Atom"
4498 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4499 open.org/ns/cmis/restatom/200908/">
4500   <cmis:statement>SELECT * FROM cmis:document</cmis:statement>
4501   <cmis:searchAllVersions>true</cmis:searchAllVersions>
4502   <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
4503   <cmis:includeRelationships>none</cmis:includeRelationships>
4504   <cmis:renditionFilter>*</cmis:renditionFilter>
4505   <cmis:maxItems>50</cmis:maxItems>
4506   <cmis:skipCount>0</cmis:skipCount>
4507 </cmis:query>

```

Please also see the example documents included with the schema.

3.3.3 CMIS Allowable Actions

Media Type: application/cmisallowableactions+xml

Starting tag: cmis:allowableActions

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

Example:

```

4519 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4520 <cmis:allowableActions xmlns:cmis="http://docs.oasis-
4521 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
4522 open.org/ns/cmis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
4523 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4524 open.org/ns/cmis/restatom/200908/">
4525   <cmis:canDeleteObject>true</cmis:canDeleteObject>
4526   <cmis:canUpdateProperties>true</cmis:canUpdateProperties>
4527   <cmis:canGetProperties>true</cmis:canGetProperties>
4528   <cmis:canGetObjectRelationships>true</cmis:canGetObjectRelationships>
4529   <cmis:canGetObjectParents>true</cmis:canGetObjectParents>
4530   <cmis:canMoveObject>true</cmis:canMoveObject>
4531   <cmis:canDeleteContentStream>true</cmis:canDeleteContentStream>
4532   <cmis:canCheckOut>true</cmis:canCheckOut>
4533   <cmis:canCancelCheckOut>true</cmis:canCancelCheckOut>
4534   <cmis:canCheckIn>true</cmis:canCheckIn>
4535   <cmis:canSetContentStream>true</cmis:canSetContentStream>
4536   <cmis:canGetAllVersions>true</cmis:canGetAllVersions>
4537   <cmis:canAddObjectToFolder>true</cmis:canAddObjectToFolder>
4538   <cmis:canRemoveObjectFromFolder>true</cmis:canRemoveObjectFromFolder>
4539   <cmis:canGetContentStream>true</cmis:canGetContentStream>
4540   <cmis:canApplyPolicy>true</cmis:canApplyPolicy>
4541   <cmis:canGetAppliedPolicies>true</cmis:canGetAppliedPolicies>
4542   <cmis:canRemovePolicy>true</cmis:canRemovePolicy>
4543   <cmis:canCreateDocument>true</cmis:canCreateDocument>
4544 </cmis:allowableActions>

```

Please also see the example documents included with the schema.

3.3.4 CMIS Tree

Media Type: application/cmistree+xml

Starting tag: atom:feed

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

Please see Section 3.3.2.1 for more information.

Example:

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/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/cmisis/restatom/200908/">
  <atom:title type="text">Feed for folder1</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>2010-01-25T10:20:58.536-08:00</atom:updated>
  <atom:id>urn:uuid:4a80905c-f774-4a9e-a57d-bf0dae5a796e</atom:id>
  <atom:link type="application/atom+xml;type=feed" rel="self"
href="http://cmisexample.oasis-open.org/repl/cf3c076e-36e9-4ace-8fed-
41e0d92dfc71/3"/>
  <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/cf3c076e-36e9-4ace-8fed-
41e0d92dfc71"/>
  <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
open.org/ns/cmisis/link/200908/foldertree" href="http://cmisexample.oasis-
open.org/repl/cf3c076e-36e9-4ace-8fed-41e0d92dfc71/foldertree"/>
  <atom:link type="application/atom+xml;type=feed" rel="down"
href="http://cmisexample.oasis-open.org/repl/cf3c076e-36e9-4ace-8fed-
41e0d92dfc71/children"/>
  <atom:link type="application/atom+xml;type=entry" rel="up"
href="http://cmisexample.oasis-open.org/repl/bb11830c-7d1e-4b0f-9ff2-
af4857c49200"/>
  <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/63a9c18c-
5e31-4590-8462-86d181e345a4"/>
    <atom:id>urn:uuid:63a9c18c-5e31-4590-8462-86d181e345a4</atom:id>
    <atom:title type="text">CMIS Example Folder as Customer
type</atom:title>
    <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
    <atom:link rel="self" href="http://cmisexample.oasis-
open.org/repl/63a9c18c-5e31-4590-8462-86d181e345a4"/>
    <atom:link rel="edit" href="http://cmisexample.oasis-
open.org/repl/63a9c18c-5e31-4590-8462-86d181e345a4"/>
```



```

4604     <atom:link type="application/cmism+xml;type=allowableActions"
4605 rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
4606 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4607 86d181e345a4/allowableactions"/>
4608     <atom:link type="application/atom+xml;type=entry" rel="describedby"
4609 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4610 86d181e345a4/type"/>
4611     <atom:link type="application/atomsvc+xml" rel="service"
4612 href="http://cmisexample.oasis-open.org/repl//service"/>
4613     <atom:published>2010-01-25T10:20:58.536-08:00</atom:published>
4614     <atom:summary type="html">HTML summary of Entry 63a9c18c-5e31-4590-
4615 8462-86d181e345a4</atom:summary>
4616     <atom:link type="application/atom+xml;type=entry" rel="up"
4617 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4618 86d181e345a4/up"/>
4619     <atom:link type="application/atom+xml;type=feed" rel="down"
4620 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4621 86d181e345a4/children"/>
4622     <atom:link type="application/cmistree+xml" rel="down"
4623 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4624 86d181e345a4/tree"/>
4625     <atom:link type="application/atom+xml;type=feed"
4626 rel="http://docs.oasis-open.org/ns/cmism/link/200908/foldertree"
4627 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4628 86d181e345a4/foldertree"/>
4629     <atom:link type="application/atom+xml;type=feed"
4630 rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
4631 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4632 86d181e345a4/relationships"/>
4633     <atom:link type="application/atom+xml;type=feed"
4634 rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
4635 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4636 86d181e345a4/policies"/>
4637     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
4638 open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
4639 open.org/repl/63a9c18c-5e31-4590-8462-86d181e345a4/acl"/>
4640     <cmisra:object>
4641       <cmis:properties>
4642         <cmis:propertyId localName="rep-cmis:objectId"
4643 propertyDefinitionId="cmis:objectId">
4644           <cmis:value>63a9c18c-5e31-4590-8462-
4645 86d181e345a4</cmis:value>
4646         </cmis:propertyId>
4647       </cmis:properties>
4648     </cmisra:object>
4649     <cmisra:pathSegment>customer</cmisra:pathSegment>
4650     <cmisra:children>
4651       <atom:feed>
4652         <atom:title type="text">CMIS Example Folder as Customer
4653 type</atom:title>
4654         <atom:author>
4655           <atom:name>Al Brown</atom:name>
4656           <atom:uri>http://www.ibm.com/</atom:uri>
4657           <atom:email>albertcbrown@us.ibm.com</atom:email>
4658         </atom:author>
4659         <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4660         <atom:id>urn:uuid:51b5c0cd-e473-4492-82b3-
4661 666fbf913cf0</atom:id>
4662         <atom:link type="application/atom+xml;type=feed" rel="self"
4663 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4664 86d181e345a4/3"/>
4665         <atom:link type="application/atomsvc+xml" rel="service"
4666 href="http://cmisexample.oasis-open.org/repl//service"/>

```

```

4667         <atom:link type="application/atom+xml;type=entry" rel="via"
4668 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4669 86d181e345a4"/>
4670         <atom:link type="application/atom+xml;type=feed"
4671 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
4672 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4673 86d181e345a4/foldertree"/>
4674         <atom:link type="application/atom+xml;type=feed" rel="down"
4675 href="http://cmisexample.oasis-open.org/repl/63a9c18c-5e31-4590-8462-
4676 86d181e345a4/children"/>
4677         <atom:link type="application/atom+xml;type=entry" rel="up"
4678 href="http://cmisexample.oasis-open.org/repl/cf3c076e-36e9-4ace-8fed-
4679 41e0d92dfc71"/>
4680         <atom:entry>
4681             <atom:author>
4682                 <atom:name>Al Brown</atom:name>
4683                 <atom:uri>http://www.ibm.com/</atom:uri>
4684                 <atom:email>albertcbrown@us.ibm.com</atom:email>
4685             </atom:author>
4686             <atom:content src="http://cmisexample.oasis-
4687 open.org/repl/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4688             <atom:id>urn:uuid:20cb7e68-0a7e-46ea-87e0-
4689 09fb8d85286e</atom:id>
4690             <atom:title type="text">CMIS Example Doc as Invoice
4691 type</atom:title>
4692             <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4693             <atom:link rel="self" href="http://cmisexample.oasis-
4694 open.org/repl/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4695             <atom:link rel="edit" href="http://cmisexample.oasis-
4696 open.org/repl/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4697             <atom:link
4698 type="application/cmis+xml;type=allowableActions" rel="http://docs.oasis-
4699 open.org/ns/cmis/link/200908/allowableactions" href="http://cmisexample.oasis-
4700 open.org/repl/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/allowableactions"/>
4701             <atom:link type="application/atom+xml;type=entry"
4702 rel="describedby" href="http://cmisexample.oasis-open.org/repl/20cb7e68-0a7e-
4703 46ea-87e0-09fb8d85286e/type"/>
4704             <atom:link type="application/atomsvc+xml" rel="service"
4705 href="http://cmisexample.oasis-open.org/repl//service"/>
4706             <atom:published>2010-01-25T10:20:58.536-
4707 08:00</atom:published>
4708             <atom:summary type="html">HTML summary of Entry 20cb7e68-
4709 0a7e-46ea-87e0-09fb8d85286e</atom:summary>
4710             <atom:link rel="edit-media"
4711 href="http://cmisexample.oasis-open.org/repl/20cb7e68-0a7e-46ea-87e0-
4712 09fb8d85286e/edit-media"/>
4713             <atom:link rel="alternate" href="http://cmisexample.oasis-
4714 open.org/repl/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/alternate"/>
4715             <atom:link type="application/atom+xml;type=feed" rel="up"
4716 href="http://cmisexample.oasis-open.org/repl/20cb7e68-0a7e-46ea-87e0-
4717 09fb8d85286e/parents"/>
4718             <atom:link type="application/atom+xml;type=feed"
4719 rel="version-history" href="http://cmisexample.oasis-open.org/repl/20cb7e68-
4720 0a7e-46ea-87e0-09fb8d85286e/allversions"/>
4721             <atom:link type="application/atom+xml;type=entry"
4722 rel="current-version" href="http://cmisexample.oasis-open.org/repl/20cb7e68-
4723 0a7e-46ea-87e0-09fb8d85286e/latest"/>
4724             <atom:link type="application/atom+xml;type=feed"
4725 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
4726 href="http://cmisexample.oasis-open.org/repl/20cb7e68-0a7e-46ea-87e0-
4727 09fb8d85286e/relationships"/>

```

```

4728         <atom:link type="application/atom+xml;type=feed"
4729 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
4730 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4731 09fb8d85286e/policies"/>
4732         <atom:link type="application/cmisacl+xml"
4733 rel="http://docs.oasis-open.org/ns/cmis/link/200908/acl"
4734 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4735 09fb8d85286e/acl"/>
4736         <cmisra:object>
4737             <cmis:properties>
4738                 <cmis:propertyId localName="rep-cmis:objectId"
4739 propertyDefinitionId="cmis:objectId">
4740 <cmis:value>20cb7e68-0a7e-46ea-87e0-09fb8d85286e</cmis:value>
4741                 </cmis:propertyId>
4742             </cmis:properties>
4743         </cmisra:object>
4744         <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
4745     </atom:entry>
4746 </atom:feed>
4747 </cmisra:children>
4748 </atom:entry>
4749 <atom:entry>
4750     <atom:author>
4751         <atom:name>Al Brown</atom:name>
4752         <atom:uri>http://www.ibm.com/</atom:uri>
4753         <atom:email>albertcbrown@us.ibm.com</atom:email>
4754     </atom:author>
4755     <atom:content src="http://cmisexample.oasis-open.org/rep1/1deld476-
4756 11fb-47bf-b136-8a8d0b4b030a"/>
4757     <atom:id>urn:uuid:1deld476-11fb-47bf-b136-8a8d0b4b030a</atom:id>
4758     <atom:title type="text">CMIS Example Doc as Invoice type</atom:title>
4759     <atom:updated>2010-01-25T10:20:58.568-08:00</atom:updated>
4760     <atom:link rel="self" href="http://cmisexample.oasis-
4761 open.org/rep1/1deld476-11fb-47bf-b136-8a8d0b4b030a"/>
4762     <atom:link rel="edit" href="http://cmisexample.oasis-
4763 open.org/rep1/1deld476-11fb-47bf-b136-8a8d0b4b030a"/>
4764     <atom:link type="application/cmis+xml;type=allowableActions"
4765 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
4766 href="http://cmisexample.oasis-open.org/rep1/1deld476-11fb-47bf-b136-
4767 8a8d0b4b030a/allowableactions"/>
4768     <atom:link type="application/atom+xml;type=entry" rel="describedby"
4769 href="http://cmisexample.oasis-open.org/rep1/1deld476-11fb-47bf-b136-
4770 8a8d0b4b030a/type"/>
4771     <atom:link type="application/atomsvc+xml" rel="service"
4772 href="http://cmisexample.oasis-open.org/rep1//service"/>
4773     <atom:published>2010-01-25T10:20:58.568-08:00</atom:published>
4774     <atom:summary type="html">HTML summary of Entry 1deld476-11fb-47bf-
4775 b136-8a8d0b4b030a</atom:summary>
4776     <atom:link rel="edit-media" href="http://cmisexample.oasis-
4777 open.org/rep1/1deld476-11fb-47bf-b136-8a8d0b4b030a/edit-media"/>
4778     <atom:link rel="alternate" href="http://cmisexample.oasis-
4779 open.org/rep1/1deld476-11fb-47bf-b136-8a8d0b4b030a/alternate"/>
4780     <atom:link type="application/atom+xml;type=feed" rel="up"
4781 href="http://cmisexample.oasis-open.org/rep1/1deld476-11fb-47bf-b136-
4782 8a8d0b4b030a/parents"/>
4783     <atom:link type="application/atom+xml;type=feed" rel="version-history"
4784 href="http://cmisexample.oasis-open.org/rep1/1deld476-11fb-47bf-b136-
4785 8a8d0b4b030a/allversions"/>
4786     <atom:link type="application/atom+xml;type=entry" rel="current-
4787 version" href="http://cmisexample.oasis-open.org/rep1/1deld476-11fb-47bf-b136-
4788 8a8d0b4b030a/latest"/>

```

```

4789     <atom:link type="application/atom+xml;type=feed"
4790     rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
4791     href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4792     8a8d0b4b030a/relationships"/>
4793     <atom:link type="application/atom+xml;type=feed"
4794     rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
4795     href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4796     8a8d0b4b030a/policies"/>
4797     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
4798     open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
4799     open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/acl"/>
4800     <cmisra:object>
4801       <cmis:properties>
4802         <cmis:propertyId localName="rep-cmis:objectId"
4803         propertyDefinitionId="cmis:objectId">
4804           <cmis:value>1de1d476-11fb-47bf-b136-
4805           8a8d0b4b030a</cmis:value>
4806         </cmis:propertyId>
4807       </cmis:properties>
4808     </cmisra:object>
4809     <cmisra:pathSegment>invoice3.pdf</cmisra:pathSegment>
4810   </atom:entry>
4811 </atom:feed>

```

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

Please also see the example documents included with the schema.

3.3.5 CMIS ACL

Media Type: application/cmisacl+xml

Starting tag: cmis:acl

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

Example:

```

4826 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4827 <cmis:acl xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
4828 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
4829 xmlns:atom="http://www.w3.org/2005/Atom"
4830 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4831 open.org/ns/cmis/restatom/200908/">
4832   <cmis:permission>
4833     <cmis:principal>
4834       <cmis:principalId>Al Brown</cmis:principalId>
4835     </cmis:principal>
4836     <cmis:permission>cmis:read</cmis:permission>
4837     <cmis:permission>cmis:write</cmis:permission>
4838     <cmis:permission>cmis:all</cmis:permission>
4839     <cmis:permission>publish</cmis:permission>
4840     <cmis:direct>true</cmis:direct>
4841   </cmis:permission>
4842 </cmis:acl>

```

4844 Please also see the example documents included with the schema.
4845

4846 **3.4 Atom Extensions for CMIS**

4847 **3.4.1 Atom Element Extensions**

4848 **3.4.1.1 AtomPub Workspace**

4849 **3.4.1.1.1 cmisra:collectionType**

4850 This element is included inside the app:collection element. This specifies the cmis collection type.

4851 **3.4.1.1.2 cmisra:repositoryInfo**

4852 This element is included inside the app:workspace element. This specifies information about the CMIS
4853 repository.

4854 **3.4.1.1.3 cmis:uritemplate**

4855 This element is included inside the app:workspace element. This specifies information about URI
4856 templates

4857 **3.4.1.2 Atom Feed**

4858 **3.4.1.2.1 cmisra:numItems**

4859 This element is included inside the atom:feed element. This specifies the number of items in the feed.

4860 **3.4.1.3 Atom Entry**

4861 **3.4.1.3.1 cmisra:children**

4862 This element is included inside the atom:entry element. This includes the children of the atom entry. This
4863 element MUST include an atom:feed element.

4864

4865 **3.4.1.3.2 cmisra:object**

4866 This element is included inside the atom:entry element for CMIS Document, Folder, Relationship and
4867 Policy objects. This specifies the CMIS object information for the atom entry.

4868

4869 **3.4.1.3.3 cmisra:pathSegment**

4870 This element is included inside the atom:entry element for CMIS Type Definitions that are filable. This
4871 specifies the pathSegment for this object in the folder representing the feed.

4872

4873 **3.4.1.3.4 cmisra:relativePathSegment**

4874 This element is included inside the atom:entry element. This specifies the relative pathSegment for the
4875 object in that particular folder. This MUST be used only inside an object parents feed.

3.4.1.3.5 cmisra:type

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

3.4.1.3.6 cmisra:content

This element specifies the content of the atom:entry element. The content is base64 encoded in the base64 element. The elements of a cmisra:content element are:

- mediaType: This contains the media type of the content as described by RFC4288.
- base64: This contains the base64 content of the file

This element MUST take precedence over atom:content on submission of an atom entry to a repository.

A repository MUST use the atom:content element to return back to the client the content of the document.

Note: This is required when the client has an XML document stored that is might not be well formed and thus would not be able to be included inside atom:content element.

3.4.2 Attributes

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

3.4.2.1 cmisra:id

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

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

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="application/atom+xml;type=feed"
rel="down" href="http://cmisexample.oasis-open.org/repl1/children/e170da7d-
d322-472d-bleb-67bdb1ec18ca/1" cmisra:id="e170da7d-d322-472d-bleb-
67bdb1ec18ca"/>
```

Please also see the example documents included with the schema.

3.4.2.2 cmisra:renditionKind

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

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/repl/rendition/e170da7d-d322-472d-
bleb-67bdbllec18ca/1" cmisra:renditionKind="cmis:thumbnail"/>
```

Please also see the example documents included with the schema.

3.4.3 CMIS Link Relations

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

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

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

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

These are the link relation types specified by CMIS:

3.4.3.1 Existing Link Relations

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

- self
 - This link relation provides the URI to retrieve this resource again.
 - Service: The appropriate service that generated the atom entry or feed.
 - Resources: All except AllowableActions, ACL and Content Streams
- service
 - The service link relation when provided on a CMIS resource **MUST** point to an AtomPub service document with only one workspace element. This workspace element **MUST** represent the repository containing that resource.
 - Media Type: application/atomsvc+xml
 - Resources: All except AllowableActions, ACL and Content Streams
- describedby
 - When used on a CMIS resource, this link relation **MUST** point to an atom entry that describes the type of that resource.
 - Service: getTypeDefinition on specified object
 - Media Type: application/atom+xml;type=entry
 - Resources: CMIS Document, CMIS Folder, CMIS Relationship, CMIS Policy objects and CMIS Types

- 4965
 - via
 - 4966
 - When used on an Atom Feed document, this link relation MUST point to the atom entry
 - 4967
 - representing the CMIS resource from whom this feed is derived.
 - 4968
 - Media Type: application/atom+xml;type=entry
 - 4969
 - Resources: All CMIS Feeds and Collections
- 4970
 - edit-media
 - 4971
 - When used on a CMIS document resource, this link relation MUST point to the URI for
 - 4972
 - content stream of the CMIS document. This URI MUST be used to set or delete the
 - 4973
 - content stream. This URI MAY be used to retrieve the content stream for the document.
 - 4974
 - Service: setContentStream (PUT) , deleteContentStream (DELETE)
 - 4975
 - Media Type: Specific to resource
 - 4976
 - Resources: CMIS Document
- 4977
 - edit
 - 4978
 - When used on a CMIS resource, this link relation MUST provide an URI that can be used
 - 4979
 - with the HTTP PUT method to modify the atom:entry for the CMIS resource
 - 4980
 - Service: getObject (GET), updateProperties (PUT)
 - 4981
 - Media Type: application/atom+xml;type=entry
 - 4982
 - Resources: CMIS Documents, CMIS Folders, CMIS Relationships and CMIS Policies
- 4983
 - alternate
 - 4984
 - This is used to express Renditions on a CMIS resource. See section 3.2.7 Renditions.
 - 4985
 - Service: getContentStream for specified rendition
 - 4986
 - Resources: CMIS Document, CMIS Folder and CMIS Policies
- 4987
 - first
 - 4988
 - This is used for Paging. Please see the AtomPub specification.
 - 4989
 - Media Type: application/atom+xml;type=feed
 - 4990
 - Resources: All Feeds
- 4991
 - previous
 - 4992
 - This is used for Paging. Please see the AtomPub specification.
 - 4993
 - Media Type: application/atom+xml;type=feed
 - 4994
 - Resources: All Feeds
- 4995
 - next
 - 4996
 - This is used for Paging. Please see the AtomPub specification.
 - 4997
 - Media Type: application/atom+xml;type=feed
 - 4998
 - Resources: All Feeds
- 4999
 - last
 - 5000
 - This is used for Paging. Please see the AtomPub specification.
 - 5001
 - Media Type: application/atom+xml;type=feed
 - 5002
 - Resources: All Feeds

5005 Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on
5006 these link relations.

3.4.3.2 Hierarchy Navigation Internet Draft Link Relations

CMIS leverages the following link relations:

- up
 - Service: getFolderParent, getObjectParents, getTypeDefinition, getObject
 - Media Type: application/atom+xml;type=feed, application/atom+xml;type=entry
 - Resources: CMIS Document, CMIS Folder, CMIS Type Definitions, CMIS Folder Children, CMIS Folder Descendants, CMIS FolderTree, CMIS Type Children, CMIS Type Descendants
 - This link relation is not included on CMIS Base Type Definitions or the CMIS Root Folder
- down
 - Service: getChildren, getDescendants, getTypeChildren, getTypeDescendants
 - Media Type:
 - For children: application/atom+xml;type=feed
 - For descendants: application/cmistree+xml
 - The descendants feed resource when retrieved from the CMIS repository will use the Atom Feed Media Type (application/atom+xml;type=feed)
 - Resources: CMIS Folder, Type

3.4.3.3 Versioning Internet Draft Link Relations

CMIS leverages the following link relations from the Internet Draft:

- version-history
 - Service: getAllVersions
 - Media Type: application/atom+xml;type=feed
 - Resources: CMIS Document
- current-version
 - Service: getObjectForLatestVersion
 - Media Type: application/atom+xml;type=entry
 - Resources: CMIS Document
- working-copy
 - Service: getObject for private-working-copy specified by cmis:versionSeriesCheckedOutId property
 - Media Type: application/atom+xml;type=entry
 - Resources: CMIS Document

3.4.3.4 CMIS Specific Link Relations

CMIS defines the following link relations:

- <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>
 - This link relation MUST point to a resource containing a CMIS AllowableActions document for the CMIS resource containing this link relation.
 - Service: getAllowableActions
 - Media Type: application/cmisallowableactions+xml
 - Resources: CMIS Documents, CMIS Folders, CMIS Policies, and CMIS Relationships

- 5049
- <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>
 - This link relation MUST point to a resource containing an Atom Feed of CMIS relationship resources for the CMIS resource containing this link relation.
 - Service: getObjectRelationships
 - Media Type: application/atom+xml;type=feed
 - Resources: CMIS Documents, CMIS Folders, and CMIS Policies
 - <http://docs.oasis-open.org/ns/cmis/link/200908/source>
 - 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.
 - Source Link on Relationship
 - Media Type: application/atom+xml;type=entry
 - Resources: CMIS Relationships
 - <http://docs.oasis-open.org/ns/cmis/link/200908/target>
 - 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.
 - Target Link on Relationship
 - Media Type: application/atom+xml;type=entry
 - Resources: CMIS Relationships
 - <http://docs.oasis-open.org/ns/cmis/link/200908/policies>
 - This link relation MUST point to a resource containing an Atom Feed of CMIS Policy resources for the CMIS resource containing this link relation.
 - Service: getAppliedPolicies
 - Media Type: application/atom+xml;type=feed
 - Resources: CMIS Documents and CMIS Folders
 - <http://docs.oasis-open.org/ns/cmis/link/200908/acl>
 - This link relation MUST point to a resource containing a CMIS ACL document for the CMIS resource containing this link relation.
 - Service: getACL
 - Media Type: application/cmisacl+xml
 - Resources: CMIS Documents, CMIS Folders, CMIS Relationships, and CMIS Policies that are securable
 - <http://docs.oasis-open.org/ns/cmis/link/200908/changes>
 - This link relation MUST point to an Atom Feed containing the set of changes
 - Service: getContentChanges
 - Media Type: application/atom+xml;type=feed
 - Resources: AtomPub Workspace Element in Service Document
 - <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>
 - Used in AtomPub Service Document to identify the folder tree for a specified folder
 - Service: getFolderTree
 - Media Type: application/atom+xml;type=feed
 - Resources: CMIS Folder, also used in AtomPub Service Document for root folder

- 5093 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>
- 5094 ▪ Used in AtomPub Service Document to identify the base types descendants
- 5095 ▪ Service: getTypeDescendants
- 5096 ▪ Media Type: application/atom+xml;type=feed
- 5097 ▪ Resources: AtomPub Workspace Element in Service Document
- 5098 ○ <http://docs.oasis-open.org/ns/cmis/link/200908/rootdescendants>
- 5099 ▪ Used in AtomPub Service Document to identify the root folder descendants
- 5100 ▪ Service: getDescendants for root folder
- 5101 ▪ Media Type: application/atom+xml;type=feed
- 5102 ▪ Resources: AtomPub Workspace Element in Service Document
- 5103

5104 **3.5 Atom Resources**

5105 For all Atom Resources used in this specification, the following MUST be followed:

5106 **3.5.1 Feeds**

5107 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis objects:

- 5108 • atom:updated SHOULD be the latest time the folder or its contents was updated. If unknown by
- 5109 the underlying repository, it MUST be the current time.
- 5110 • atom:author/atom:name MUST be the CMIS property cmis:createdBy
- 5111 • atom:title MUST be the CMIS property cmis:name
- 5112 • The atom:link with relation self MUST be generated to return the URI of the feed. If paging or any
- 5113 other mechanism is used to filter, sort, or change the representation of the feed, the URI MUST
- 5114 point back a resource with the same representation.
- 5115 • A feed SHOULD contain the element app:collection, describing the appropriate media types
- 5116 supported for creation of new entries in the feed
- 5117 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's
- 5118 specification and be a valid URI.
- 5119 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
- 5120 contained in the feed, then a link with the relation next MUST be included in the feed.

5121

5122 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis types:

- 5123 • atom:updated SHOULD be the latest time type definition was updated. If unknown by the
- 5124 underlying repository, it MUST be the current time.
- 5125 • atom:author/atom:name is repository specific
- 5126 • atom:title MUST be the displayName attribute of the CMIS Type Definition.
- 5127 • The atom:link with relation self MUST be generated to return the URI of the feed
- 5128 • atom:id SHOULD be derived from the id attribute of the CMIS Type Definition. This id MUST be
- 5129 compliant with atom's specification and be a valid URI.
- 5130 • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than
- 5131 contained in the feed, then a link with the relation next MUST be included in the feed.

5132

5133 If on the root type, all fields are repository specific.

5134

5135 Ordering of entries in a feed is repository-specific if orderBy argument is not specified. If orderBy
5136 argument is specified, the order of the entries in the feed SHOULD conform to the ordering specified by
5137 the orderBy argument.

5138

5139 Note: Please see feedvalidator.org to validate Atom compliance.

5140 3.5.2 Entries

5141 At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry
5142 document and conform to the guidelines below for a cmis object:

- 5143 • atom:title MUST be the cmis:name property
- 5144 • app:edited MUST be cmis:lastModifiedDate
- 5145 • atom:updated MUST be cmis:lastModifiedDate
- 5146 • atom:published MUST be cmis:createdDate
- 5147 • atom:author/atom:name MUST be cmis:createdBy
- 5148 • All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are
5149 duplicated in an atom element
- 5150 • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom's
5151 specification and be a valid URI.
- 5152 • The repository SHOULD populate the atom:summary tag with text that best represents a
5153 summary of the object. For example, an HTML table containing the properties and their values or
5154 the description of the document if available.

5155

5156 For Documents that support Content Streams:

5157 The repository SHOULD use the atom:content/src attribute to point to the content stream.
5158 The client SHOULD use cmisra:content if the content is not well-formed or would have
5159 trouble fitting inside an atom:content element. The repository MUST use the
5160 cmisra:content element if provided by the client over the atom:content element.

5161

5162 Other Objects (Folders, Relationships, and other Document Types that do not support Content
5163 Streams, etc):

5164 The repository MUST comply with the atom specification and have an atom:content
5165 element. This is repository specific. Any value in the content field MUST be ignored if the
5166 atom entry represents a non-document object by the CMIS repository when the atom
5167 entry is POST'ed to a collection or sent to the repository via a PUT.

5168

5169 When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding
5170 writable CMIS property. For example, atom:title will overwrite cmis:name.

5171

5172 At any point where an Atom document of CMIS Type is sent or returned, it must be a valid Atom Entry
5173 document and conform to the guidelines below for a cmis type definition:

- 5174 • atom:title MUST be the cmis:displayName
- 5175 • The repository SHOULD populate the atom:summary tag with text that best represents a
5176 summary of the object. For example, the type description if available.
- 5177 • The repository MUST comply with the atom specification and have an atom:content element. This
5178 is repository specific. Any value in the content field MUST be ignored if the atom entry represents
5179 a non-document object by the CMIS repository when the atom entry is POST'ed to a collection or
5180 sent to the repository via a PUT.

5181
5182
5183 Any atom element that is not specified is repository-specific.

5184 3.5.2.1 Hierarchical Atom Entries

5185 The repository SHOULD NOT provide any links to hierarchical objects if those capabilities are not
5186 supported with the exception of getTypeDescendants which is required

5187
5188 For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate
5189 a cmisra:children element in the atom:entry with the enclosing feed of its direct children. This pattern
5190 continues until the depth is satisfied.

5191
5192 The cmisra:children element MUST include an atom:feed element that contains the children entries of this
5193 resource.

5194
5195 If an entry does not contain cmisra:children element, then the entry MAY have children even though it is
5196 not represented in the atom entry.

5197
5198 For Example, here is a minimal Atom Entry with CMIS Children Extension Element:

```
5199 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5200 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
5201 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
5202 xmlns:atom="http://www.w3.org/2005/Atom"
5203 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5204 open.org/ns/cmis/restatom/200908/">
5205   <atom:author>
5206     <atom:name>Al Brown</atom:name>
5207   </atom:author>
5208   <atom:content src="http://cmisexample.oasis-open.org/repl/afl8c7f-b554-
5209 4dfb-bfel-1f41e4b34fef"/>
5210   <atom:id>urn:uuid:afl8c7f-b554-4dfb-bfel-1f41e4b34fef</atom:id>
5211   <atom:title type="text">CMIS Example Folder as Customer type</atom:title>
5212   <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5213   <cmisra:object>
5214     <cmis:properties>
5215       <cmis:propertyId localName="rep-cmis:objectId"
5216 propertyDefinitionId="cmis:objectId">
5217         <cmis:value>afl8c7f-b554-4dfb-bfel-1f41e4b34fef</cmis:value>
5218       </cmis:propertyId>
5219     </cmis:properties>
5220   </cmisra:object>
5221   <cmisra:pathSegment>customer</cmisra:pathSegment>
5222   <cmisra:children>
5223     <atom:feed>
5224       <atom:title type="text">CMIS Example Folder as Customer
5225 type</atom:title>
5226       <atom:author>
5227         <atom:name>Al Brown</atom:name>
5228         <atom:uri>http://www.ibm.com/</atom:uri>
5229         <atom:email>albertcbrown@us.ibm.com</atom:email>
5230       </atom:author>
5231       <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5232       <atom:id>urn:uuid:ce2d65af-b246-454b-90ff-0986d9b05178</atom:id>
5233       <atom:link type="application/atom+xml;type=feed" rel="self"
5234 href="http://cmisexample.oasis-open.org/repl/afl8c7f-b554-4dfb-bfel-
5235 1f41e4b34fef/3"/>
```

```

5236         <atom:link type="application/atomsvc+xml" rel="service"
5237 href="http://cmisexample.oasis-open.org/repl//service"/>
5238         <atom:link type="application/atom+xml;type=entry" rel="via"
5239 href="http://cmisexample.oasis-open.org/repl/af1d8c7f-b554-4dfb-bfe1-
5240 1f41e4b34fef"/>
5241         <atom:link type="application/atom+xml;type=feed"
5242 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
5243 href="http://cmisexample.oasis-open.org/repl/af1d8c7f-b554-4dfb-bfe1-
5244 1f41e4b34fef/foldertree"/>
5245         <atom:link type="application/atom+xml;type=feed" rel="down"
5246 href="http://cmisexample.oasis-open.org/repl/af1d8c7f-b554-4dfb-bfe1-
5247 1f41e4b34fef/children"/>
5248         <atom:link type="application/atom+xml;type=entry" rel="up"
5249 href="http://cmisexample.oasis-open.org/repl/2eb09309-58f7-4627-b735-
5250 4d5cf4ba6554"/>
5251         <atom:entry>
5252             <atom:author>
5253                 <atom:name>Al Brown</atom:name>
5254             </atom:author>
5255             <atom:content src="http://cmisexample.oasis-
5256 open.org/repl/af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef"/>
5257             <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-
5258 1f41e4b34fef</atom:id>
5259             <atom:title type="text">CMIS Example Child of
5260 Folder</atom:title>
5261             <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5262             <cmisra:object>
5263                 <cmis:properties>
5264                     <cmis:propertyId localName="rep-cmis:objectId"
5265 propertyDefinitionId="cmis:objectId">
5266                         <cmis:value>af1d8c7f-b554-4dfb-bfe1-
5267 1f41e4b34fef</cmis:value>
5268                     </cmis:propertyId>
5269                 </cmis:properties>
5270             </cmisra:object>
5271             <cmisra:pathSegment>document</cmisra:pathSegment>
5272         </atom:entry>
5273     </atom:feed>
5274 </cmisra:children>
5275 </atom:entry>
5276

```

Please also see the example documents included with the schema.

3.6 AtomPub Service Document (Repository)

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

CMIS Services exposed:

GET: getRepositories, getRepositoryInfo

Media Type: application/atomsvc+xml

How the client will get the initial AtomPub (APP) service document or the URI for the service document is repository specific. Examples are via URI, or loading the service document from disk.

The service document will be available from Atom Entry and Atom Feed documents via a link relationship, service. That AtomPub service document MUST contain only one workspace element which MUST be the workspace representing the repository containing the Atom Entry or Atom Feed document.

A workspace element for a CMIS repository MUST have a collection element for each of following collections: Each collection MUST also contain a `cmisra:collectionType` element with the given value:

- Root Folder Children Collection: Root folder of the Repository
 - 'root' for the children collection of the root folder
 - `cmisra:collectiontype='root'`
- Types Children Collection: Collection containing the base types in the repository
 - 'types' for the children collection
 - `cmisra:collectiontype='types'`

The workspace element SHOULD contain these collections if the repository supports this functionality:

- 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'`

The repository MUST include the URI templates in the workspace elements.

The workspace element MUST also contain the following link element with the relation:

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

The workspace element MUST contain the following link elements of the following relations for those services which are supported by the repository:

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

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

3.6.1 URI Templates

CMIS defines the following URI Templates:

- objectbyid
- objectbypath
- query
- typebyid

Repositories MUST provide the following URI Templates:

- objectbyid
- objectbypath
- typebyid

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

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

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

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

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

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

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

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

All variables MUST be in the template.

Structure of URI Template:

```
<xs:complexType name="cmisUriTemplateType">
  <xs:sequence>
    <xs:element name="template" type="xs:string" />
    <xs:element name="type" type="xs:string" />
    <xs:element name="mediatype" type="xs:string" />
    <xs:any processContents="lax" namespace="##other"
minOccurs="0"
maxOccurs="unbounded" />
  </xs:sequence>
</xs:complexType>
```


Example of URI Template element in an AtomPub Workspace Element:

```
<?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/repl/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
  <cmisra:type>objectbyid</cmisra:type>
  <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
</cmisra:uritemplate>
```

Please also see the example documents included with the schema.

3.6.1.1 Object By Id

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

Type: objectbyid

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

Service: getObjectById

Variables that are supported by the template:

- {id}: Id of object
- {filter}: Property Filter
- {includeAllowableActions}
 - Valid values: true, false
- {includePolicyIds}: Include Policy Ids:
 - Valid values: true, false
- {includeRelationships}: Include relationships
 - Valid values: See enumIncludeRelationships
- {includeACL}: Include ACLs
 - Valid values: true, false
- {renditionFilter}
 - Valid values: Please see renditionFilter in CMIS Domain Model

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/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
includeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
    <cmisra:type>objectbyid</cmisra:type>
    <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
</cmisra:uritemplate>

```

Please also see the example documents included with the schema.

3.6.1.2 Object By Path

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

Type: objectbypath

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

Service: getObjectByPath

Variables that are supported by the template:

- {path}: Path of Object
- {filter}: Property Filter
- {includeAllowableActions}: Boolean for include Allowable Actions
 - Valid values: true, false
- {includePolicyIds}: Include Policy Ids:
 - Valid values: true, false
- {includeRelationships}: Include relationships
 - Valid values: See enumIncludeRelationships
- {includeACL}: Include ACLs
 - Valid values: true, false
- {renditionFilter}
 - Valid values: Please see renditionFilter in CMIS Domain Model

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/objectbypath?p={path}&filter={filter}&includeAllowableAc
tions={includeAllowableActions}&includePolicyIds={includePolicyIds}&in
cludeRelationships={includeRelationships}&includeACL={includeACL}</cmisra:
template>
  <cmisra:type>objectbypath</cmisra:type>
  <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>

```

```
</cmisra:uritemplate>
```

Please also see the example documents included with the schema.

3.6.1.3 Query

Type: query

Media Type: application/atom+xml;type=feed

Service: query

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

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/repl/query?q={q}&searchAllVersions={searchAllVersions}&maxIte
ms={maxItems}&skipCount={skipCount}&includeAllowableActions={includeAl
lowableActions}&includeRelationships={includeRelationships}</cmisra:templ
ate>
  <cmisra:type>query</cmisra:type>
  <cmisra:mediatype>application/atom+xml;type=feed</cmisra:mediatype>
</cmisra:uritemplate>
```

Please also see the example documents included with the schema.

3.6.1.4 Type By Id

Type: typebyid

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

Service: getTypeDefinition

Variables that are supported by the template:

- {id}: CMIS Type Id

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/repl/type?id={id}</cmisra:template>
  <cmisra:type>query</cmisra:type>
  <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
</cmisra:uritemplate>
```

Please also see the example documents included with the schema.

3.6.2 HTTP Methods

3.6.2.1 GET

This retrieves the AtomPub Service document for a specified repository. This exposes the capabilities defined in getRepositories and getRepositoryInfo in the Domain Model.

The optional argument MAY be specified:

- repositoryId:
 - This query parameter allows a client to specify a different repository than the one that is referenced by the URI.
 - If specified, the repository MUST return the AtomPub services document for the specified repository if that repository exists.
 - If not specified, the repository MUST return the service document for the repository that is referenced by URI.

3.7 Service Collections

These are the collections that are included on an AtomPub Service document in the workspace element. For any HTTP verb not specified on a resource, each implementation MAY choose to implement that HTTP verb in a repository-specific manner.

3.7.1 Root Folder Collection

This is a collection described in the service document. Please see [Folder Children](#).

3.7.2 Query Collection

This is a collection for processing queries. If the implementation supports GET on this collection, then the implementation SHOULD at least return a feed consisting of zero or more atom entries. These atom entries should represent persisted objects related to query such as persisted queries, long running queries or search templates.

5565
5566 CMIS Services exposed via HTTP verbs:
5567 POST: Query
5568
5569 Media Type: application/atom+xml;type=feed
5570 Accept:
5571

- MUST support CMIS Query document,
- MAY support other media type

5572
5573
5574 Link Relations on resulting feed from Query Collection:
5575

- service: Points to service document containing the CMIS repository. The service document
- 5576 MUST contain only one workspace element.
 - 5577 ○ Media Type: application/atomsvc+xml
- 5578 • paging link relations as appropriate: first, next, previous, last

5579
5580 The following CMIS Atom extension element MAY be included inside the atom feed:
5581

- cmisra:numItems

5582
5583 The following CMIS Atom extension element MUST be included inside the atom entries:
5584

- cmisra:object inside atom:entry

5585

5586 **3.7.2.1 POST**
5587 This collection MUST accept CMIS Query documents (application/cmisquery+xml).
5588
5589 Upon submission (creation) of a query document, a response must be returned with a Location header
5590 representing the feed for that query. If the query cannot be performed and an atom feed returned, the
5591 repository MUST return the appropriate HTTP status code. In addition, the server SHOULD return the
5592 feed directly. If the server does so, the server should also return the Content-Location header.
5593
5594 The feed returned MUST contain a set of atom entries representing the result set from the query.
5595
5596 The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The
5597 atom entries MUST contain the CMIS extension element (cmisra:object) containing the properties
5598 specified by the query in the select clause of the query statement.
5599
5600 If all the selected properties can be mapped to the same type reference, then the repository MAY include
5601 additional information in the atom entry.
5602
5603 Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.
5604
5605 Status Codes:
5606

- 201 Success

5607
5608 Headers returned:

- 5609 • Location Header
- 5610 • Content-Location Header
- 5611

5612 Link Relations on resulting feed from POST to Query Collection:

- 5613 • service: Points to service document containing the CMIS repository. The service document
- 5614 MUST contain only one workspace element.
 - 5615 ○ Media Type: application/atomsvc+xml
- 5616 • paging link relations as appropriate: first, next, previous, last
- 5617

5618 Example client request:

```
5619 POST /Query HTTP/1.1
5620 Host: example.org
5621 Content-Length: 756
5622 Content-Type: application/cmisquery+xml
5623
5624
5625 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5626 <cmis:query xmlns:app="http://www.w3.org/2007/app"
5627 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5628 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5629 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5630 open.org/ns/cmis/restatom/200908/">
5631   <cmis:statement>SELECT cmis:objectId FROM cmis:document</cmis:statement>
5632   <cmis:searchAllVersions>true</cmis:searchAllVersions>
5633   <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
5634   <cmis:includeRelationships>none</cmis:includeRelationships>
5635   <cmis:renditionFilter>*</cmis:renditionFilter>
5636   <cmis:maxItems>50</cmis:maxItems>
5637   <cmis:skipCount>0</cmis:skipCount>
5638 </cmis:query>
5639
5640
```

5641

5642 Example server response:

```
5643 HTTP/1.1 201 Created
5644 Date: Mon, 25 Jan 2010 10:21:00 -0800
5645 Content-Length: 1830
5646 Content-Type: application/atom+xml;type=feed
5647 Content-Location: http://cmisexample.oasis-open.org/repl/queryresult/44ce5b47-
5648 ebc3-4513-86e0-d3f46c77d0a8
5649 Location: http://cmisexample.oasis-open.org/repl/queryresult/44ce5b47-ebc3-
5650 4513-86e0-d3f46c77d0a8
5651
5652
5653 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5654 <atom:feed xmlns:app="http://www.w3.org/2007/app"
5655 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5656 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
5657 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5658 open.org/ns/cmis/restatom/200908/">
5659   <atom:title type="text">CMIS Query Result for SELECT cmis:objectId FROM
5660 cmis:document</atom:title>
5661   <atom:author>
5662     <atom:name>Al Brown</atom:name>
5663     <atom:uri>http://www.ibm.com/</atom:uri>
5664     <atom:email>albertcbrown@us.ibm.com</atom:email>
5665   </atom:author>
5666   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
5667   <atom:id>urn:uuid:811b1b9b-80f5-4788-b46c-aa77564e294b</atom:id>
```

```

5668     <atom:link type="application/atom+xml;type=feed" rel="self"
5669 href="http://cmisexample.oasis-open.org/rep1/11355977-434b-4e71-b83a-
5670 77dea9878e04/3"/>
5671     <atom:link type="application/atomsvc+xml" rel="service"
5672 href="http://cmisexample.oasis-open.org/rep1//service"/>
5673     <atom:entry>
5674       <atom:author>
5675         <atom:name>Al Brown</atom:name>
5676       </atom:author>
5677       <atom:content src="http://cmisexample.oasis-open.org/rep1/a3386ea0-
5678 0477-4a74-96bd-70d3dalc483a"/>
5679       <atom:id>urn:uuid:a3386ea0-0477-4a74-96bd-70d3dalc483a</atom:id>
5680       <atom:title type="text">Resulting Document</atom:title>
5681       <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
5682       <cmisra:object>
5683         <cmis:properties>
5684           <cmis:propertyId queryName="cmis:objectId" localName="rep-
5685 cmis:objectId" propertyDefinitionId="cmis:objectId">
5686             <cmis:value>a3386ea0-0477-4a74-96bd-
5687 70d3dalc483a</cmis:value>
5688           </cmis:propertyId>
5689         </cmis:properties>
5690       </cmisra:object>
5691     </atom:entry>
5692 </atom:feed>
5693

```

Please also see the example documents included with the schema.

3.7.3 Checked Out Collection

This is a collection described in the service document that contains the private working copies (PWCs) of the checkedout documents in the repository.

CMIS Services:

GET: getCheckedOutDocs

POST: checkOut

Media Type: application/atom+xml;type=feed

Accept:

- MUST support Atom Entry Documents with CMIS extensions
 - application/atom+xml;type=entry or
 - application/cmisaatom+xml
- MAY support other media type

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
- paging link relations as appropriate: first, next, previous, last

The following CMIS Atom extension element MAY be included inside the atom feed:

- cmisra:numItems

The following CMIS Atom extension element **MUST** be included inside the atom entries:

- cmisra:object inside atom:entry

3.7.3.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- folderId
- maxItems
- skipCount
- renditionFilter
- includeAllowableActions
- includeRelationships

3.7.3.2 POST

When an atom entry is POST'ed to this collection, the atom entry will be checked out. A Content-Location header **MUST** be returned containing the location of the private working copy.

Example client request:

```
POST /CheckedOut HTTP/1.1
Host: example.org
Content-Length: 1044
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:content src="http://cmisexample.oasis-open.org/rep1/8d32d716-701b-
4491-84e8-ad57c8230940"/>
  <atom:id>urn:uuid:8d32d716-701b-4491-84e8-ad57c8230940</atom:id>
  <atom:title type="text">CMIS Example Document to checkout</atom:title>
  <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
  <cmisra:object>
    <cmis:properties>
      <cmis:propertyId localName="rep-cmis:objectId"
propertyDefinitionId="cmis:objectId">
        <cmis:value>8d32d716-701b-4491-84e8-ad57c8230940</cmis:value>
      </cmis:propertyId>
    </cmis:properties>
  </cmisra:object>
</atom:entry>
```

Example server response:

```
HTTP/1.1 201 Created
Date: Mon, 25 Jan 2010 10:21:00 -0800
Content-Length: 7846
```


5771 Content-Type: application/atom+xml;type=entry
 5772 Content-Location: http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-
 5773 8fab-4aa6e6797dbe
 5774 Location: http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5775 4aa6e6797dbe
 5776
 5777
 5778 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
 5779 <atom:entry xmlns:app="http://www.w3.org/2007/app"
 5780 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
 5781 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
 5782 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
 5783 open.org/ns/cmisis/restatom/200908/">
 5784 <atom:author>
 5785 <atom:name>Al Brown</atom:name>
 5786 <atom:uri>http://www.ibm.com/</atom:uri>
 5787 <atom:email>albertcbrown@us.ibm.com</atom:email>
 5788 </atom:author>
 5789 <atom:content src="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-
 5790 491c-8fab-4aa6e6797dbe"/>
 5791 <atom:id>urn:uuid:6cce57fc-4e31-491c-8fab-4aa6e6797dbe</atom:id>
 5792 <atom:title type="text">CMIS Example Child of Folder</atom:title>
 5793 <atom:updated>2010-01-25T10:21:00.396-08:00</atom:updated>
 5794 <atom:link rel="self" href="http://cmisexample.oasis-
 5795 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
 5796 <atom:link rel="edit" href="http://cmisexample.oasis-
 5797 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
 5798 <atom:link type="application/cmisis+xml;type=allowableActions"
 5799 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
 5800 href="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5801 4aa6e6797dbe/allowableactions"/>
 5802 <atom:link type="application/atom+xml;type=entry" rel="describedby"
 5803 href="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5804 4aa6e6797dbe/type"/>
 5805 <atom:link type="application/atomsvc+xml" rel="service"
 5806 href="http://cmisexample.oasis-open.org/repl//service"/>
 5807 <atom:published>2010-01-25T10:21:00.396-08:00</atom:published>
 5808 <atom:summary type="html">HTML summary of Entry 6cce57fc-4e31-491c-8fab-
 5809 4aa6e6797dbe</atom:summary>
 5810 <atom:link rel="edit-media" href="http://cmisexample.oasis-
 5811 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/edit-media"/>
 5812 <atom:link rel="alternate" href="http://cmisexample.oasis-
 5813 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/alternate"/>
 5814 <atom:link type="application/atom+xml;type=feed" rel="up"
 5815 href="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5816 4aa6e6797dbe/parents"/>
 5817 <atom:link type="application/atom+xml;type=feed" rel="version-history"
 5818 href="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5819 4aa6e6797dbe/allversions"/>
 5820 <atom:link type="application/atom+xml;type=entry" rel="current-version"
 5821 href="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5822 4aa6e6797dbe/latest"/>
 5823 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
 5824 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
 5825 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/relationships"/>
 5826 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
 5827 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
 5828 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/policies"/>
 5829 <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
 5830 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
 5831 open.org/repl/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/acl"/>
 5832 <atom:link type="application/atom+xml;type=feed" rel="working-copy"
 5833 href="http://cmisexample.oasis-open.org/repl/6cce57fc-4e31-491c-8fab-
 5834 4aa6e6797dbe/pwc"/>

```

5835     <cmisra:object>
5836         <cmis:properties>
5837             <cmis:propertyId localName="rep-cmis:objectId"
5838 propertyDefinitionId="cmis:objectId">
5839                 <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
5840             </cmis:propertyId>
5841             <cmis:propertyId localName="rep-cmis:objectTypeId"
5842 propertyDefinitionId="cmis:objectTypeId">
5843                 <cmis:value>customer</cmis:value>
5844             </cmis:propertyId>
5845             <cmis:propertyString localName="rep-cmis:name"
5846 propertyDefinitionId="cmis:name">
5847                 <cmis:value>CMIS Example Child of Folder</cmis:value>
5848             </cmis:propertyString>
5849             <cmis:propertyDateTime localName="rep-cmis:creationDate"
5850 propertyDefinitionId="cmis:creationDate">
5851                 <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
5852             </cmis:propertyDateTime>
5853             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
5854 propertyDefinitionId="cmis:lastModificationDate">
5855                 <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
5856             </cmis:propertyDateTime>
5857             <cmis:propertyId localName="rep-cmis:baseTypeId"
5858 propertyDefinitionId="cmis:baseTypeId">
5859                 <cmis:value>cmis:document</cmis:value>
5860             </cmis:propertyId>
5861             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
5862 propertyDefinitionId="cmis:lastModifiedBy">
5863                 <cmis:value>Al Brown</cmis:value>
5864             </cmis:propertyString>
5865             <cmis:propertyString localName="rep-cmis:createdBy"
5866 propertyDefinitionId="cmis:createdBy">
5867                 <cmis:value>Al Brown</cmis:value>
5868             </cmis:propertyString>
5869             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
5870 propertyDefinitionId="cmis:isLatestVersion">
5871                 <cmis:value>true</cmis:value>
5872             </cmis:propertyBoolean>
5873             <cmis:propertyBoolean localName="rep-
5874 cmis:isVersionSeriesCheckedOut"
5875 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
5876                 <cmis:value>true</cmis:value>
5877             </cmis:propertyBoolean>
5878             <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
5879 propertyDefinitionId="cmis:isMajorVersion">
5880                 <cmis:value>false</cmis:value>
5881             </cmis:propertyBoolean>
5882             <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
5883 propertyDefinitionId="cmis:isLatestMajorVersion">
5884                 <cmis:value>false</cmis:value>
5885             </cmis:propertyBoolean>
5886             <cmis:propertyBoolean localName="rep-cmis:isImmutable"
5887 propertyDefinitionId="cmis:isImmutable">
5888                 <cmis:value>false</cmis:value>
5889             </cmis:propertyBoolean>
5890             <cmis:propertyString localName="rep-cmis:checkinComment"
5891 propertyDefinitionId="cmis:checkinComment">
5892                 <cmis:value>Checkin comment</cmis:value>
5893             </cmis:propertyString>
5894             <cmis:propertyString localName="rep-cmis:versionLabel"
5895 propertyDefinitionId="cmis:versionLabel">
5896                 <cmis:value>0.1</cmis:value>
5897             </cmis:propertyString>

```

```

5898         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
5899 propertyDefinitionId="cmis:contentStreamMimeType">
5900         <cmis:value>text/plain</cmis:value>
5901       </cmis:propertyString>
5902       <cmis:propertyString localName="rep-cmis:contentStreamFileName"
5903 propertyDefinitionId="cmis:contentStreamFileName">
5904       <cmis:value>text.txt</cmis:value>
5905     </cmis:propertyString>
5906     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
5907 propertyDefinitionId="cmis:contentStreamLength">
5908     <cmis:value>4234</cmis:value>
5909   </cmis:propertyInteger>
5910   <cmis:propertyString displayName="Keywords for Document"
5911 localName="keywords" propertyDefinitionId="keywords">
5912     <cmis:value>document</cmis:value>
5913     <cmis:value>example</cmis:value>
5914     <cmis:value>sample</cmis:value>
5915     <cmis:value>cmis</cmis:value>
5916   </cmis:propertyString>
5917   <cmis:propertyId localName="rep-cmis:versionSeriesCheckedOutId"
5918 propertyDefinitionId="cmis:versionSeriesCheckedOutId">
5919   <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
5920 </cmis:propertyId>
5921   <cmis:propertyString localName="rep-
5922 cmis:versionSeriesCheckedOutBy"
5923 propertyDefinitionId="cmis:versionSeriesCheckedOutBy">
5924   <cmis:value>Al Brown</cmis:value>
5925 </cmis:propertyString>
5926 </cmis:properties>
5927 </cmisra:object>
5928 </atom:entry>
5929

```

Please also see the example documents included with the schema.

3.7.4 Unfiled Collection

This is a collection described in the service document that contains all the unfiled documents in the repository. This collection **MUST** be available if un-filing or multi-filing is supported by the repository.

A repository that supports un-filing **MAY** provide read access (GET). If read access is not provided, the repository **SHOULD** respond to a read attempt with the HTTP status code 405 (notSupported).

CMIS Services:

POST: removeObjectFromFolder

Media Type: application/atom+xml;type=feed

Accept:

- **MUST** support Atom Entry Documents with CMIS extensions
 - application/atom+xml;type=entry or
 - application/cmisatom+xml
- **MAY** support other media type

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

- paging link relations as appropriate: first, next, previous, last

The following CMIS Atom extension element MAY be included inside the atom feed:

- cmisra:numItems

The following CMIS Atom extension element MUST be included inside the atom entries:

- cmisra:object inside atom:entry

3.7.4.1 POST

This removes the object from all folders in the repository by default. If the optional argument removeFrom is specified, the object will only be removed from that folder only.

If the Atom Entry POST'ed, does not have the CMIS extensions with a valid cmis:objectId property, the document does not exist, or the document is not in that folder, the appropriate HTTP status code MUST be returned.

This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- HTTP Success: 201
- Location Header

The following arguments may be supplied. Please see the domain model for more information:

- removeFrom: For repositories which support multi-filing, this parameter identifies which folder to remove this object from. If specified, it indicates the folder from which the object shall be moved. If not specified, the object will be removed from all folders.

Example client request:

```
POST /Unfiled HTTP/1.1
Host: example.org
Content-Length: 1043
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:content src="http://cmisexample.oasis-open.org/repl/12aa2bec-6f43-
47d1-99ef-21797867173c"/>
  <atom:id>urn:uuid:12aa2bec-6f43-47d1-99ef-21797867173c</atom:id>
  <atom:title type="text">CMIS Example Document to unfiled</atom:title>
  <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
  <cmisra:object>
    <cmis:properties>
      <cmis:propertyId localName="rep-cmis:objectId"
propertyDefinitionId="cmis:objectId">
        <cmis:value>12aa2bec-6f43-47d1-99ef-21797867173c</cmis:value>
      </cmis:propertyId>
```

```
6003         </cmis:properties>
6004     </cmisra:object>
6005 </atom:entry>
6006
```

6007

6008 Example server response:

```
6009 HTTP/1.1 201 Created
6010 Date: Mon, 25 Jan 2010 10:21:00 -0800
6011 Content-Length: 7234
6012 Content-Type: application/atom+xml;type=entry
6013 Content-Location: http://cmisexample.oasis-open.org/repl/queryresult/15118373-
6014 8911-442b-9774-da3b102f224c
6015 Location: http://cmisexample.oasis-open.org/repl/queryresult/15118373-8911-
6016 442b-9774-da3b102f224c
6017
6018
6019 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6020 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6021 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6022 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6023 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6024 open.org/ns/cmis/restatom/200908/">
6025     <atom:author>
6026         <atom:name>Al Brown</atom:name>
6027         <atom:uri>http://www.ibm.com</atom:uri>
6028         <atom:email>albertcbrown@us.ibm.com</atom:email>
6029     </atom:author>
6030     <atom:content src="http://cmisexample.oasis-open.org/repl/15118373-8911-
6031 442b-9774-da3b102f224c"/>
6032     <atom:id>urn:uuid:15118373-8911-442b-9774-da3b102f224c</atom:id>
6033     <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6034     <atom:updated>2010-01-25T10:21:00.443-08:00</atom:updated>
6035     <atom:link rel="self" href="http://cmisexample.oasis-
6036 open.org/repl/15118373-8911-442b-9774-da3b102f224c"/>
6037     <atom:link rel="edit" href="http://cmisexample.oasis-
6038 open.org/repl/15118373-8911-442b-9774-da3b102f224c"/>
6039     <atom:link type="application/cmis+xml;type=allowableActions"
6040 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6041 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6042 da3b102f224c/allowableactions"/>
6043     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6044 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6045 da3b102f224c/type"/>
6046     <atom:link type="application/atomsvc+xml" rel="service"
6047 href="http://cmisexample.oasis-open.org/repl//service"/>
6048     <atom:published>2010-01-25T10:21:00.443-08:00</atom:published>
6049     <atom:summary type="html">HTML summary of Entry 15118373-8911-442b-9774-
6050 da3b102f224c</atom:summary>
6051     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6052 open.org/repl/15118373-8911-442b-9774-da3b102f224c/edit-media"/>
6053     <atom:link rel="alternate" href="http://cmisexample.oasis-
6054 open.org/repl/15118373-8911-442b-9774-da3b102f224c/alternate"/>
6055     <atom:link type="application/atom+xml;type=feed" rel="up"
6056 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6057 da3b102f224c/parents"/>
6058     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6059 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6060 da3b102f224c/allversions"/>
6061     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6062 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6063 da3b102f224c/latest"/>
```

```

6064     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6065 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
6066 open.org/rep1/15118373-8911-442b-9774-da3b102f224c/relationships"/>
6067     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6068 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
6069 open.org/rep1/15118373-8911-442b-9774-da3b102f224c/policies"/>
6070     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
6071 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
6072 open.org/rep1/15118373-8911-442b-9774-da3b102f224c/acl"/>
6073     <cmisra:object>
6074         <cmis:properties>
6075             <cmis:propertyId localName="rep-cmis:objectId"
6076 propertyDefinitionId="cmis:objectId">
6077                 <cmis:value>15118373-8911-442b-9774-da3b102f224c</cmis:value>
6078             </cmis:propertyId>
6079             <cmis:propertyId localName="rep-cmis:objectTypeId"
6080 propertyDefinitionId="cmis:objectTypeId">
6081                 <cmis:value>customer</cmis:value>
6082             </cmis:propertyId>
6083             <cmis:propertyString localName="rep-cmis:name"
6084 propertyDefinitionId="cmis:name">
6085                 <cmis:value>CMIS Example Document to unfiled</cmis:value>
6086             </cmis:propertyString>
6087             <cmis:propertyDateTime localName="rep-cmis:creationDate"
6088 propertyDefinitionId="cmis:creationDate">
6089                 <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
6090             </cmis:propertyDateTime>
6091             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6092 propertyDefinitionId="cmis:lastModificationDate">
6093                 <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
6094             </cmis:propertyDateTime>
6095             <cmis:propertyId localName="rep-cmis:baseTypeId"
6096 propertyDefinitionId="cmis:baseTypeId">
6097                 <cmis:value>cmis:document</cmis:value>
6098             </cmis:propertyId>
6099             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6100 propertyDefinitionId="cmis:lastModifiedBy">
6101                 <cmis:value>Al Brown</cmis:value>
6102             </cmis:propertyString>
6103             <cmis:propertyString localName="rep-cmis:createdBy"
6104 propertyDefinitionId="cmis:createdBy">
6105                 <cmis:value>Al Brown</cmis:value>
6106             </cmis:propertyString>
6107             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6108 propertyDefinitionId="cmis:isLatestVersion">
6109                 <cmis:value>true</cmis:value>
6110             </cmis:propertyBoolean>
6111             <cmis:propertyBoolean localName="rep-
6112 cmis:isVersionSeriesCheckedOut"
6113 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6114                 <cmis:value>false</cmis:value>
6115             </cmis:propertyBoolean>
6116             <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6117 propertyDefinitionId="cmis:isMajorVersion">
6118                 <cmis:value>false</cmis:value>
6119             </cmis:propertyBoolean>
6120             <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6121 propertyDefinitionId="cmis:isLatestMajorVersion">
6122                 <cmis:value>false</cmis:value>
6123             </cmis:propertyBoolean>
6124             <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6125 propertyDefinitionId="cmis:isImmutable">
6126                 <cmis:value>false</cmis:value>
6127             </cmis:propertyBoolean>

```

```

6128         <cmis:propertyString localName="rep-cmis:checkinComment"
6129 propertyDefinitionId="cmis:checkinComment">
6130         <cmis:value>Checkin comment</cmis:value>
6131       </cmis:propertyString>
6132       <cmis:propertyString localName="rep-cmis:versionLabel"
6133 propertyDefinitionId="cmis:versionLabel">
6134       <cmis:value>0.1</cmis:value>
6135     </cmis:propertyString>
6136     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6137 propertyDefinitionId="cmis:contentStreamMimeType">
6138     <cmis:value>text/plain</cmis:value>
6139   </cmis:propertyString>
6140   <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6141 propertyDefinitionId="cmis:contentStreamFileName">
6142   <cmis:value>text.txt</cmis:value>
6143 </cmis:propertyString>
6144 <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6145 propertyDefinitionId="cmis:contentStreamLength">
6146 <cmis:value>4234</cmis:value>
6147 </cmis:propertyInteger>
6148 <cmis:propertyString displayName="Keywords for Document"
6149 localName="keywords" propertyDefinitionId="keywords">
6150   <cmis:value>document</cmis:value>
6151   <cmis:value>example</cmis:value>
6152   <cmis:value>sample</cmis:value>
6153   <cmis:value>cmis</cmis:value>
6154 </cmis:propertyString>
6155 </cmis:properties>
6156 </cmisra:object>
6157 </atom:entry>
6158

```

Please also see the example documents included with the schema.

3.7.5 Types Children Collection

This is a collection described in the service document that contains the types in the repository under the specified parent type. If no parent type is specified, then the base types are returned in the feed. This feed does not include any nesting and is a flat feed.

CMIS Services:

GET: getTypeChildren

Media Type: application/atom+xml;type=feed

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 type definition entry whose children represent this feed
- down: points to the atom feed document representing the descendents collection for this same 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.

6181 This feed contains a set of atom entries for each child type definition.

6182

6183 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6184 • cmisra:numItems

6185

6186 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6187 • cmisra:type inside atom:entry

6188

6189

6190 3.7.5.1 GET

6191 The following arguments may be supplied. Please see the domain model for more information:

- 6192 • includePropertyDefinitions
- 6193 • maxItems
- 6194 • skipCount
- 6195 • typeId

6196 3.8 Collections

6197 For any HTTP verb not specified on a resource, each implementation MAY choose to implement that HTTP
6198 verb in a repository-specific manner.

6199

6200 3.8.1 Relationships Collection

6201 This is the set of relationships available (either source or target or both) from a specific item such as a
6202 document, folder or policy.

6203 CMIS Services:

6204 GET: getObjectRelationships

6205 POST: createRelationship

6206 Media Type: application/atom+xml;type=feed

6207 Accept:

- 6208 • MUST support Atom Entry Documents with CMIS extensions
 - 6209 ○ application/atom+xml;type=entry or
 - 6210 ○ application/cmisatom+xml
- 6211 • MAY support other media type

6212

6213 Link Relations:

- 6214 • service: Points to service document containing the CMIS repository. The service document
6215 MUST contain only one workspace element.
 - 6216 ○ Media Type: application/atomsvc+xml
- 6217 • paging link relations as appropriate: first, next, previous, last

6218

6219 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6220 • cmisra:numItems

6221

6222 The following CMIS Atom extension element **MUST** be included inside the atom entries:

- 6223 • cmisra:object inside atom:entry

6224

6225 3.8.1.1 GET

6226 The following arguments may be supplied. Please see the domain model for more information:

- 6227 • typeId
- 6228 • includeSubRelationshipTypes
- 6229 • relationshipDirection
- 6230 • maxItems
- 6231 • skipCount
- 6232 • filter
- 6233 • includeAllowableActions

6234 3.8.1.2 POST

6235 When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new
6236 CMIS relationship, then that relationship will be created.

6237 The server **MUST** return the appropriate HTTP status code if the source is different than the sourceId or
6238 target different than the targetId for the source and targets specified in this collection.

6239 The server **MUST** return the appropriate status code if the cmis:objectTypeId is not specified.

6240

6241 Example client request:

```
6242 POST /relationships/source/dbf0316c-47b5-47c9-a2fa-f005eb93f0a4 HTTP/1.1
6243 Host: example.org
6244 Content-Length: 1432
6245 Content-Type: application/atom+xml;type=entry
6246
6247
6248 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6249 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6250 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6251 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6252 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6253 open.org/ns/cmis/restatom/200908/">
6254   <atom:author>
6255     <atom:name>Al Brown</atom:name>
6256   </atom:author>
6257   <atom:content src="http://cmisexample.oasis-open.org/repl/dab97641-8c94-
6258 4a12-a604-7532980f05cb"/>
6259   <atom:id>urn:uuid:dab97641-8c94-4a12-a604-7532980f05cb</atom:id>
6260   <atom:title type="text">New Relationship</atom:title>
6261   <atom:updated>2010-01-25T10:20:58.864-08:00</atom:updated>
6262   <cmisra:object>
6263     <cmis:properties>
6264       <cmis:propertyId localName="rep-cmis:objectTypeId"
6265 propertyDefinitionId="cmis:objectTypeId">
6266         <cmis:value>customerRelationships</cmis:value>
6267       </cmis:propertyId>
6268       <cmis:propertyId localName="rep-cmis:sourceId"
6269 propertyDefinitionId="cmis:sourceId">
6270         <cmis:value>dbf0316c-47b5-47c9-a2fa-f005eb93f0a4</cmis:value>
6271       </cmis:propertyId>
```

```

        <cmis:propertyId localName="rep-cmis:targetId"
propertyDefinitionId="cmis:targetId">
        <cmis:value>b9baac7d-7584-445e-bcd1-29af9b25bf2f</cmis:value>
    </cmis:propertyId>
</cmis:properties>
</cmisra:object>
</atom:entry>

```

Example server response:

```

HTTP/1.1 201 Created
Date: Mon, 25 Jan 2010 10:20:58 -0800
Content-Length: 4684
Content-Type: application/atom+xml;type=entry
Content-Location: http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495
Location: http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495

<?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/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
    <atom:id>urn:uuid:b3006a8f-345b-4c27-86df-3f4b157bb495</atom:id>
    <atom:title type="text">New Relationship</atom:title>
    <atom:updated>2010-01-25T10:20:58.880-08:00</atom:updated>
    <atom:link rel="self" href="http://cmisexample.oasis-
open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
    <atom:link rel="edit" href="http://cmisexample.oasis-
open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
    <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/b3006a8f-345b-4c27-86df-3f4b157bb495/allowableactions"/>
    <atom:link type="application/atom+xml;type=entry" rel="describedby"
href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/type"/>
    <atom:link type="application/atomsvc+xml" rel="service"
href="http://cmisexample.oasis-open.org/rep1//service"/>
    <atom:published>2010-01-25T10:20:58.880-08:00</atom:published>
    <atom:summary type="html">HTML summary of Entry b3006a8f-345b-4c27-86df-3f4b157bb495</atom:summary>
    <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
open.org/ns/cmis/link/200908/source" href="http://cmisexample.oasis-
open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/source"/>
    <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
open.org/ns/cmis/link/200908/target" href="http://cmisexample.oasis-
open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/target"/>
    <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/policies"/>

```

```

6333     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6334 open.org/ns/cmisacl/link/200908/acl" href="http://cmisexample.oasis-
6335 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/acl"/>
6336     <cmisra:object>
6337         <cmis:properties>
6338             <cmis:propertyId localName="rep-cmis:objectId"
6339 propertyDefinitionId="cmis:objectId">
6340                 <cmis:value>b3006a8f-345b-4c27-86df-3f4b157bb495</cmis:value>
6341             </cmis:propertyId>
6342             <cmis:propertyId localName="rep-cmis:objectTypeId"
6343 propertyDefinitionId="cmis:objectTypeId">
6344                 <cmis:value>customerRelationships</cmis:value>
6345             </cmis:propertyId>
6346             <cmis:propertyString localName="rep-cmis:name"
6347 propertyDefinitionId="cmis:name">
6348                 <cmis:value>New Relationship</cmis:value>
6349             </cmis:propertyString>
6350             <cmis:propertyDateTime localName="rep-cmis:creationDate"
6351 propertyDefinitionId="cmis:creationDate">
6352                 <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
6353             </cmis:propertyDateTime>
6354             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6355 propertyDefinitionId="cmis:lastModificationDate">
6356                 <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
6357             </cmis:propertyDateTime>
6358             <cmis:propertyId localName="rep-cmis:baseTypeId"
6359 propertyDefinitionId="cmis:baseTypeId">
6360                 <cmis:value>cmis:relationship</cmis:value>
6361             </cmis:propertyId>
6362             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6363 propertyDefinitionId="cmis:lastModifiedBy">
6364                 <cmis:value>Al Brown</cmis:value>
6365             </cmis:propertyString>
6366             <cmis:propertyString localName="rep-cmis:createdBy"
6367 propertyDefinitionId="cmis:createdBy">
6368                 <cmis:value>Al Brown</cmis:value>
6369             </cmis:propertyString>
6370             <cmis:propertyId localName="rep-cmis:sourceId"
6371 propertyDefinitionId="cmis:sourceId">
6372                 <cmis:value>d4551c6d-30bd-4fc2-9c84-a55f11559e89</cmis:value>
6373             </cmis:propertyId>
6374             <cmis:propertyId localName="rep-cmis:targetId"
6375 propertyDefinitionId="cmis:targetId">
6376                 <cmis:value>fe7e056f-c4bf-42f1-a03e-3ababcf2491d</cmis:value>
6377             </cmis:propertyId>
6378         </cmis:properties>
6379     </cmisra:object>
6380 </atom:entry>
6381

```

Please also see the example documents included with the schema.

3.8.2 Folder Children Collection

This is a collection comprised of all the direct children of a particular folder represented as a feed.

CMIS Services:

GET: getChildren

POST:

createDocument

6391 or createFolder
6392 or createPolicy
6393 or moveObject
6394 or addObjectToFolder
6395

6396 Media Type: application/atom+xml;type=feed
6397

6398 Accept:

- 6399 • MUST support Atom Entry Documents with CMIS extensions
 - 6400 • MAY support other media type
- 6401

6402 Link Relations:

- 6403 • service: Points to service document containing the CMIS repository. The service document
6404 MUST contain only one workspace element.
 - 6405 ○ Media Type: application/atomsvc+xml
 - 6406 • via: points to the atom entry of the folder generating this collection
 - 6407 • up: points to the atom entry document for this folder's parent
 - 6408 ○ If the root folder, this link relation MUST NOT be included.
 - 6409 ○ Media Type: application/atom+xml;type=entry
 - 6410 • down: points to the atom feed document representing the descendants feed with a media type of
6411 application/cmistree+xml
 - 6412 ○ If a repository does not support capabilityGetDescendants, then this link SHOULD NOT
6413 be included.
 - 6414 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder.
6415 This is represented as a feed with CMIS hierarchy extensions.
 - 6416 ○ Media Type: application/atom+xml;type=feed
 - 6417 • paging link relations as appropriate: first, next, previous, last
- 6418

6419 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6420 • cmisra:numItems
- 6421

6422 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6423 • cmisra:object inside atom:entry
 - 6424 • cmisra:pathSegment inside atom:entry if pathSegment is not false
- 6425

6426 3.8.2.1 GET

6427 HTTP Code:

- 6428 • 200 OK (Success)
- 6429

6430 The following arguments may be supplied. Please see the domain model for more information:

- 6431 • maxItems
- 6432 • skipCount
- 6433 • filter

- 6434 • includeAllowableActions
- 6435 • includeRelationships
- 6436 • renditionFilter
 - 6437 ○ If specified, renditions will be returned as links with relation alternate.
- 6438 • orderBy
- 6439 • includePathSegment

6440 3.8.2.2 POST

6441 CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please
6442 see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 6443 • HTTP Success: 201
- 6444 • Location Header

6445

6446 The following arguments MAY be supplied.

- 6447 • sourceFolderId: This parameter indicates the folder from which the object shall be moved from to
6448 the current specified folder. This parameter is not allowed for create operations.
 - 6449 ○ If specified moveObject will be performed.
 - 6450 ○ If not specified, addObjectToFolder will be performed.
- 6451 • versioningState: The optional argument versioningState MAY specify additional versioning
6452 behavior such as checkIn as major or minor. Please see CMIS Domain Model for more
6453 information on this parameter.

6454

6455 POSTing an Atom Entry document with CMIS markup:

6456 Adding a document to a folder:

6457 If the atom entry has a cmis property cmis:objectId that is valid for the repository, the object will
6458 be added to the folder.

6459

6460 When an object is added to the folder, in repositories that do not support multi-filing it will be
6461 removed from the previous folder and the operation treated as move. If the repository supports
6462 multiple folders, it will be added to the new folder.

6463 If the optional argument sourceFolderId is specified, then the object will be removed from the
6464 folder specified.

6465

6466 If atom:content is missing from the request, the repository MUST treat the missing atom:content
6467 element as an empty atom:content element.

6468 Example client request:

```
6469 POST /obj/1cd0d82f-d579-4897-9b0a-ad0917595445?sourceFolderId=313fd58d-2eab-
6470 41af-9517-06dadb010d49 HTTP/1.1
6471 Host: example.org
6472 Content-Length: 1227
6473 Content-Type: application/atom+xml;type=entry
6474
6475 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6476 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6477 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6478 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6479 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6480 open.org/ns/cmis/restatom/200908/">
6481 <atom:author>
6482 <atom:name>Al Brown</atom:name>
```

```

6484     </atom:author>
6485     <atom:id>urn:uuid:1cd0d82f-d579-4897-9b0a-ad0917595445</atom:id>
6486     <atom:title type="text">Document - To Be Moved</atom:title>
6487     <atom:updated>2010-01-25T10:20:58.708-08:00</atom:updated>
6488     <atom:content src="http://cmisexample.oasis-
6489 open.org/rep1//content/1cd0d82f-d579-4897-9b0a-ad0917595445"/>
6490     <cmisra:object>
6491         <cmis:properties>
6492             <cmis:propertyId localName="rep-cmis:objectId"
6493 propertyDefinitionId="cmis:objectId">
6494                 <cmis:value>1cd0d82f-d579-4897-9b0a-ad0917595445</cmis:value>
6495             </cmis:propertyId>
6496             <cmis:propertyId localName="rep-cmis:objectTypeId"
6497 propertyDefinitionId="cmis:objectTypeId">
6498                 <cmis:value>invoice</cmis:value>
6499             </cmis:propertyId>
6500         </cmis:properties>
6501     </cmisra:object>
6502 </atom:entry>
6503

```

Example server response:

```

6506 HTTP/1.1 201 Created
6507 Date: Mon, 25 Jan 2010 10:20:58 -0800
6508 Content-Length: 7213
6509 Content-Type: application/atom+xml;type=entry
6510 Content-Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-
6511 8141-4aed91d28b5b
6512 Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
6513 4aed91d28b5b
6514
6515
6516 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6517 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6518 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6519 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6520 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6521 open.org/ns/cmis/restatom/200908/">
6522     <atom:author>
6523         <atom:name>Al Brown</atom:name>
6524         <atom:uri>http://www.ibm.com/</atom:uri>
6525         <atom:email>albertcbrown@us.ibm.com</atom:email>
6526     </atom:author>
6527     <atom:content src="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-
6528 49fb-8141-4aed91d28b5b"/>
6529     <atom:id>urn:uuid:b4423b8a-e46e-49fb-8141-4aed91d28b5b</atom:id>
6530     <atom:title type="text">Document - To Be Moved</atom:title>
6531     <atom:updated>2010-01-25T10:20:58.786-08:00</atom:updated>
6532     <atom:link rel="self" href="http://cmisexample.oasis-
6533 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b"/>
6534     <atom:link rel="edit" href="http://cmisexample.oasis-
6535 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b"/>
6536     <atom:link type="application/cmis+xml;type=allowableActions"
6537 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6538 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
6539 4aed91d28b5b/allowableactions"/>
6540     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6541 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-
6542 4aed91d28b5b/type"/>
6543     <atom:link type="application/atomsvc+xml" rel="service"
6544 href="http://cmisexample.oasis-open.org/rep1//service"/>
6545     <atom:published>2010-01-25T10:20:58.786-08:00</atom:published>

```

```

6546     <atom:summary type="html">HTML summary of Entry  b4423b8a-e46e-49fb-8141-
6547 4aed91d28b5b</atom:summary>
6548     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6549 open.org/repl/b4423b8a-e46e-49fb-8141-4aed91d28b5b/edit-media"/>
6550     <atom:link rel="alternate" href="http://cmisexample.oasis-
6551 open.org/repl/b4423b8a-e46e-49fb-8141-4aed91d28b5b/alternate"/>
6552     <atom:link type="application/atom+xml;type=feed" rel="up"
6553 href="http://cmisexample.oasis-open.org/repl/b4423b8a-e46e-49fb-8141-
6554 4aed91d28b5b/parents"/>
6555     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6556 href="http://cmisexample.oasis-open.org/repl/b4423b8a-e46e-49fb-8141-
6557 4aed91d28b5b/allversions"/>
6558     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6559 href="http://cmisexample.oasis-open.org/repl/b4423b8a-e46e-49fb-8141-
6560 4aed91d28b5b/latest"/>
6561     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6562 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
6563 open.org/repl/b4423b8a-e46e-49fb-8141-4aed91d28b5b/relationships"/>
6564     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6565 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
6566 open.org/repl/b4423b8a-e46e-49fb-8141-4aed91d28b5b/policies"/>
6567     <atom:link type="application/cmisac+xml" rel="http://docs.oasis-
6568 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
6569 open.org/repl/b4423b8a-e46e-49fb-8141-4aed91d28b5b/acl"/>
6570     <cmisra:object>
6571       <cmis:properties>
6572         <cmis:propertyId localName="rep-cmis:objectId"
6573 propertyDefinitionId="cmis:objectId">
6574           <cmis:value>b4423b8a-e46e-49fb-8141-4aed91d28b5b</cmis:value>
6575         </cmis:propertyId>
6576         <cmis:propertyId localName="rep-cmis:objectTypeId"
6577 propertyDefinitionId="cmis:objectTypeId">
6578           <cmis:value>invoice</cmis:value>
6579         </cmis:propertyId>
6580         <cmis:propertyString localName="rep-cmis:name"
6581 propertyDefinitionId="cmis:name">
6582           <cmis:value>Document - To Be Moved</cmis:value>
6583         </cmis:propertyString>
6584         <cmis:propertyDateTime localName="rep-cmis:creationDate"
6585 propertyDefinitionId="cmis:creationDate">
6586           <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
6587         </cmis:propertyDateTime>
6588         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6589 propertyDefinitionId="cmis:lastModificationDate">
6590           <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
6591         </cmis:propertyDateTime>
6592         <cmis:propertyId localName="rep-cmis:baseTypeId"
6593 propertyDefinitionId="cmis:baseTypeId">
6594           <cmis:value>cmis:document</cmis:value>
6595         </cmis:propertyId>
6596         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6597 propertyDefinitionId="cmis:lastModifiedBy">
6598           <cmis:value>Al Brown</cmis:value>
6599         </cmis:propertyString>
6600         <cmis:propertyString localName="rep-cmis:createdBy"
6601 propertyDefinitionId="cmis:createdBy">
6602           <cmis:value>Al Brown</cmis:value>
6603         </cmis:propertyString>
6604         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6605 propertyDefinitionId="cmis:isLatestVersion">
6606           <cmis:value>true</cmis:value>
6607         </cmis:propertyBoolean>

```

```

6608         <cmis:propertyBoolean localName="rep-
6609 cmis:isVersionSeriesCheckedOut"
6610 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6611         <cmis:value>false</cmis:value>
6612     </cmis:propertyBoolean>
6613     <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6614 propertyDefinitionId="cmis:isMajorVersion">
6615         <cmis:value>false</cmis:value>
6616     </cmis:propertyBoolean>
6617     <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6618 propertyDefinitionId="cmis:isLatestMajorVersion">
6619         <cmis:value>false</cmis:value>
6620     </cmis:propertyBoolean>
6621     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6622 propertyDefinitionId="cmis:isImmutable">
6623         <cmis:value>false</cmis:value>
6624     </cmis:propertyBoolean>
6625     <cmis:propertyString localName="rep-cmis:checkinComment"
6626 propertyDefinitionId="cmis:checkinComment">
6627         <cmis:value>Checkin comment</cmis:value>
6628     </cmis:propertyString>
6629     <cmis:propertyString localName="rep-cmis:versionLabel"
6630 propertyDefinitionId="cmis:versionLabel">
6631         <cmis:value>0.1</cmis:value>
6632     </cmis:propertyString>
6633     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6634 propertyDefinitionId="cmis:contentStreamMimeType">
6635         <cmis:value>text/plain</cmis:value>
6636     </cmis:propertyString>
6637     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6638 propertyDefinitionId="cmis:contentStreamFileName">
6639         <cmis:value>text.txt</cmis:value>
6640     </cmis:propertyString>
6641     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6642 propertyDefinitionId="cmis:contentStreamLength">
6643         <cmis:value>4234</cmis:value>
6644     </cmis:propertyInteger>
6645     <cmis:propertyString displayName="Keywords for Document"
6646 localName="keywords" propertyDefinitionId="keywords">
6647         <cmis:value>document</cmis:value>
6648         <cmis:value>example</cmis:value>
6649         <cmis:value>sample</cmis:value>
6650         <cmis:value>cmis</cmis:value>
6651     </cmis:propertyString>
6652 </cmis:properties>
6653 </cmisra:object>
6654 </atom:entry>
6655

```

Please also see the example documents included with the schema.

Creating a CMIS Object (in that folder):

If the cmis:objectId property is missing, the object will be created and then added to the folder. If the cmis:objectId property is present but not a valid object Id, the repository MUST return the appropriate HTTP status code.

For Documents:

If Content Stream is not provided and it is required by the type definition, the repository MUST return the appropriate HTTP status code.

Content Streams MAY be provided by any of the following mechanisms:

- As part of the atom entry via the src attribute on the content element (AtomPub)
 - src attribute: Implementers MAY support external references to content
 - If the URI in the src attribute is not reachable, then an appropriate http status code should be returned.
- As part of the atom entry inlining via the content element (AtomPub)
 - Please see the AtomPub specification RFC5023 for the processing model of the content element.
- If the cmisra:content is provided by the client inside the atom:entry, the cmisra:content element MUST take precedence over the atom:content element. (CMIS)
 - This element cmisra:content is base64 encoded
- At a later time (AtomPub)
 - At a later time by replacing the edit-media link with a new content

The optional argument versioningState MAY specify additional versioning behavior such as checkin.

Example client request:

```
POST /obj/bb2b208b-3acd-4abe-9788-8078a239f228 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:bb2b208b-3acd-4abe-9788-8078a239f228</atom:id>
  <atom:title type="text">New Invoice</atom:title>
  <atom:updated>2010-01-25T10:20:58.818-08:00</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>bb2b208b-3acd-4abe-9788-8078a239f228</cmis:value>
      </cmis:propertyId>
      <cmis:propertyId localName="rep-cmis:objectTypeId"
propertyDefinitionId="cmis:objectTypeId">
        <cmis:value>invoice</cmis:value>
      </cmis:propertyId>
    </cmis:properties>
  </cmisra:object>
</atom:entry>
```

6722 Example server response:

```
6723 HTTP/1.1 201 Created
6724 Date: Mon, 25 Jan 2010 10:20:58 -0800
6725 Content-Length: 7191
6726 Content-Type: application/atom+xml;type=entry
6727 Content-Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-
6728 ad0b-10ea94c4b93d
6729 Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6730 10ea94c4b93d
6731
6732
6733 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6734 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6735 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6736 open.org/ns/cmismessaging/200908/" xmlns:cmism="http://docs.oasis-
6737 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
6738 open.org/ns/cmismessaging/200908/">
6739   <atom:author>
6740     <atom:name>Al Brown</atom:name>
6741     <atom:uri>http://www.ibm.com</atom:uri>
6742     <atom:email>albertcbrown@us.ibm.com</atom:email>
6743   </atom:author>
6744   <atom:content src="http://cmisexample.oasis-open.org/rep1/13475008-6a20-
6745 4454-ad0b-10ea94c4b93d"/>
6746   <atom:id>urn:uuid:13475008-6a20-4454-ad0b-10ea94c4b93d</atom:id>
6747   <atom:title type="text">New Invoice</atom:title>
6748   <atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
6749   <atom:link rel="self" href="http://cmisexample.oasis-
6750 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>
6751   <atom:link rel="edit" href="http://cmisexample.oasis-
6752 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>
6753   <atom:link type="application/cmismessaging+xml;type=allowableActions"
6754 rel="http://docs.oasis-open.org/ns/cmismessaging/200908/allowableactions"
6755 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6756 10ea94c4b93d/allowableactions"/>
6757   <atom:link type="application/atom+xml;type=entry" rel="describedby"
6758 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6759 10ea94c4b93d/type"/>
6760   <atom:link type="application/atomsvc+xml" rel="service"
6761 href="http://cmisexample.oasis-open.org/rep1/service"/>
6762   <atom:published>2010-01-25T10:20:58.833-08:00</atom:published>
6763   <atom:summary type="html">HTML summary of Entry 13475008-6a20-4454-ad0b-
6764 10ea94c4b93d</atom:summary>
6765   <atom:link rel="edit-media" href="http://cmisexample.oasis-
6766 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/edit-media"/>
6767   <atom:link rel="alternate" href="http://cmisexample.oasis-
6768 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/alternate"/>
6769   <atom:link type="application/atom+xml;type=feed" rel="up"
6770 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6771 10ea94c4b93d/parents"/>
6772   <atom:link type="application/atom+xml;type=feed" rel="version-history"
6773 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6774 10ea94c4b93d/allversions"/>
6775   <atom:link type="application/atom+xml;type=entry" rel="current-version"
6776 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6777 10ea94c4b93d/latest"/>
6778   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6779 open.org/ns/cmismessaging/200908/relationships" href="http://cmisexample.oasis-
6780 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/relationships"/>
6781   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6782 open.org/ns/cmismessaging/200908/policies" href="http://cmisexample.oasis-
6783 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/policies"/>
```

```

6784     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6785 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
6786 open.org/repl/13475008-6a20-4454-ad0b-10ea94c4b93d/acl"/>
6787     <cmisra:object>
6788         <cmis:properties>
6789             <cmis:propertyId localName="rep-cmis:objectId"
6790 propertyDefinitionId="cmis:objectId">
6791                 <cmis:value>13475008-6a20-4454-ad0b-10ea94c4b93d</cmis:value>
6792             </cmis:propertyId>
6793             <cmis:propertyId localName="rep-cmis:objectTypeId"
6794 propertyDefinitionId="cmis:objectTypeId">
6795                 <cmis:value>invoice</cmis:value>
6796             </cmis:propertyId>
6797             <cmis:propertyString localName="rep-cmis:name"
6798 propertyDefinitionId="cmis:name">
6799                 <cmis:value>New Invoice</cmis:value>
6800             </cmis:propertyString>
6801             <cmis:propertyDateTime localName="rep-cmis:creationDate"
6802 propertyDefinitionId="cmis:creationDate">
6803                 <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
6804             </cmis:propertyDateTime>
6805             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6806 propertyDefinitionId="cmis:lastModificationDate">
6807                 <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
6808             </cmis:propertyDateTime>
6809             <cmis:propertyId localName="rep-cmis:baseTypeId"
6810 propertyDefinitionId="cmis:baseTypeId">
6811                 <cmis:value>cmis:document</cmis:value>
6812             </cmis:propertyId>
6813             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6814 propertyDefinitionId="cmis:lastModifiedBy">
6815                 <cmis:value>Al Brown</cmis:value>
6816             </cmis:propertyString>
6817             <cmis:propertyString localName="rep-cmis:createdBy"
6818 propertyDefinitionId="cmis:createdBy">
6819                 <cmis:value>Al Brown</cmis:value>
6820             </cmis:propertyString>
6821             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6822 propertyDefinitionId="cmis:isLatestVersion">
6823                 <cmis:value>true</cmis:value>
6824             </cmis:propertyBoolean>
6825             <cmis:propertyBoolean localName="rep-
6826 cmis:isVersionSeriesCheckedOut"
6827 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6828                 <cmis:value>>false</cmis:value>
6829             </cmis:propertyBoolean>
6830             <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6831 propertyDefinitionId="cmis:isMajorVersion">
6832                 <cmis:value>>false</cmis:value>
6833             </cmis:propertyBoolean>
6834             <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6835 propertyDefinitionId="cmis:isLatestMajorVersion">
6836                 <cmis:value>>false</cmis:value>
6837             </cmis:propertyBoolean>
6838             <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6839 propertyDefinitionId="cmis:isImmutable">
6840                 <cmis:value>>false</cmis:value>
6841             </cmis:propertyBoolean>
6842             <cmis:propertyString localName="rep-cmis:checkinComment"
6843 propertyDefinitionId="cmis:checkinComment">
6844                 <cmis:value>Checkin comment</cmis:value>
6845             </cmis:propertyString>
6846             <cmis:propertyString localName="rep-cmis:versionLabel"
6847 propertyDefinitionId="cmis:versionLabel">

```

```

6848         <cmis:value>0.1</cmis:value>
6849     </cmis:propertyString>
6850     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6851 propertyDefinitionId="cmis:contentStreamMimeType">
6852         <cmis:value>text/plain</cmis:value>
6853     </cmis:propertyString>
6854     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6855 propertyDefinitionId="cmis:contentStreamFileName">
6856         <cmis:value>text.txt</cmis:value>
6857     </cmis:propertyString>
6858     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6859 propertyDefinitionId="cmis:contentStreamLength">
6860         <cmis:value>4234</cmis:value>
6861     </cmis:propertyInteger>
6862     <cmis:propertyString displayName="Keywords for Document "
6863 localName="keywords" propertyDefinitionId="keywords">
6864         <cmis:value>document</cmis:value>
6865         <cmis:value>example</cmis:value>
6866         <cmis:value>sample</cmis:value>
6867         <cmis:value>cmis</cmis:value>
6868     </cmis:propertyString>
6869 </cmis:properties>
6870 </cmisra:object>
6871 </atom:entry>
6872

```

Please also see the example documents included with the schema.

POSTing other document formats: (AtomPub)

The behavior is repository specific when a non Atom entry or an atom document without the CMIS elements is posted to a folder collection.

For example, the repository MAY auto-create a document with a specific type (document) the client could edit.

If the repository does not support this scenario or another exception occurs, then the repository MUST return the appropriate HTTP status code.

Optional arguments:

- versioningState (for createDocument)
- sourceFolderId (for moveObject)

3.8.3 Policies Collection

This is an atom feed of all the policy objects currently applied to a specific object. This is the only collection where the URI's of the objects in the collection MUST be specific to that collection. A DELETE on the policy object in the collection is a removal of the policy from the object NOT a deletion of the policy object itself.

CMIS Services:

GET: getAppliedPolicies

POST: applyPolicy (to object representing this collection of policies)

DELETE: removePolicy

Media Type: application/atom+xml;type=feed

6899 Accept:

- 6900 • MUST support Atom Entry Documents with CMIS extensions
 - 6901 ○ application/atom+xml;type=entry or
 - 6902 ○ application/cmisatom+xml
- 6903 • MAY support other media type

6904

6905 Link Relations:

- 6906 • service: Points to service document containing the CMIS repository. The service document
 - 6907 MUST contain only one workspace element.
 - 6908 ○ Media Type: application/atomsvc+xml
- 6909 • via: points to the atom entry of the resource generating this collection
- 6910 • paging link relations as appropriate: first, next, previous, last

6911

6912 The policy entries displayed here are specific to the object generating this collection. A DELETE method
6913 on those URIs will invoke removePolicy().

6914

6915 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6916 • cmisra:numItems

6917

6918 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6919 • cmisra:object inside atom:entry

6920

6921 3.8.3.1 GET

6922 The following arguments may be supplied. Please see the domain model for more information:

- 6923 • filter

6924 3.8.3.2 POST

6925 When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the
6926 object.

6927

6928 Example client request:

```
6929 POST /policies/f3670f66-62ee-487f-b733-999a69237024 HTTP/1.1
6930 Host: example.org
6931 Content-Length: 1039
6932 Content-Type: application/atom+xml;type=entry
6933
6934
6935 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6936 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6937 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6938 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6939 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6940 open.org/ns/cmis/restatom/200908/">
6941   <atom:author>
6942     <atom:name>Al Brown</atom:name>
6943   </atom:author>
6944   <atom:content src="http://cmisexample.oasis-open.org/repl/f3670f66-62ee-
6945 487f-b733-999a69237024"/>
```

```

6946     <atom:id>urn:uuid:f3670f66-62ee-487f-b733-999a69237024</atom:id>
6947     <atom:title type="text">Security Policy for Invoices</atom:title>
6948     <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
6949     <cmisra:object>
6950         <cmis:properties>
6951             <cmis:propertyId localName="rep-cmis:objectId"
6952 propertyDefinitionId="cmis:objectId">
6953                 <cmis:value>f3670f66-62ee-487f-b733-999a69237024</cmis:value>
6954             </cmis:propertyId>
6955         </cmis:properties>
6956     </cmisra:object>
6957 </atom:entry>
6958

```

Example server response:

```

6961 HTTP/1.1 201 Created
6962 Date: Mon, 25 Jan 2010 10:20:58 -0800
6963 Content-Length: 4043
6964 Content-Type: application/atom+xml;type=entry
6965 Content-Location: http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-
6966 bdfe-690761576116
6967 Location: http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
6968 690761576116
6969
6970
6971 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6972 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6973 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6974 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6975 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6976 open.org/ns/cmis/restatom/200908/">
6977     <atom:author>
6978         <atom:name>Al Brown</atom:name>
6979         <atom:uri>http://www.ibm.com/</atom:uri>
6980         <atom:email>albertcbrown@us.ibm.com</atom:email>
6981     </atom:author>
6982     <atom:content src="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-
6983 4354-bdfe-690761576116"/>
6984     <atom:id>urn:uuid:55cca51b-6cfa-4354-bdfe-690761576116</atom:id>
6985     <atom:title type="text">Security Policy for Invoices</atom:title>
6986     <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
6987     <atom:link rel="self" href="http://cmisexample.oasis-
6988 open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116"/>
6989     <atom:link rel="edit" href="http://cmisexample.oasis-
6990 open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116"/>
6991     <atom:link type="application/cmis+xml;type=allowableActions"
6992 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
6993 href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
6994 690761576116/allowableactions"/>
6995     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6996 href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
6997 690761576116/type"/>
6998     <atom:link type="application/atomsvc+xml" rel="service"
6999 href="http://cmisexample.oasis-open.org/repl//service"/>
7000     <atom:published>2010-01-25T10:20:58.849-08:00</atom:published>
7001     <atom:summary type="html">HTML summary of Entry 55cca51b-6cfa-4354-bdfe-
7002 690761576116</atom:summary>
7003     <atom:link type="application/atom+xml;type=feed" rel="up"
7004 href="http://cmisexample.oasis-open.org/repl/55cca51b-6cfa-4354-bdfe-
7005 690761576116/parents"/>
7006     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7007 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
7008 open.org/repl/55cca51b-6cfa-4354-bdfe-690761576116/relationships"/>

```

```

7009     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7010 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7011 open.org/rep1/55cca51b-6cfa-4354-bdfe-690761576116/acl"/>
7012     <cmisra:object>
7013         <cmis:properties>
7014             <cmis:propertyId localName="rep-cmis:objectId"
7015 propertyDefinitionId="cmis:objectId">
7016                 <cmis:value>55cca51b-6cfa-4354-bdfe-690761576116</cmis:value>
7017             </cmis:propertyId>
7018             <cmis:propertyId localName="rep-cmis:objectTypeId"
7019 propertyDefinitionId="cmis:objectTypeId">
7020                 <cmis:value>generalSecurityPolicy</cmis:value>
7021             </cmis:propertyId>
7022             <cmis:propertyString localName="rep-cmis:name"
7023 propertyDefinitionId="cmis:name">
7024                 <cmis:value>Security Policy for Invoices</cmis:value>
7025             </cmis:propertyString>
7026             <cmis:propertyDateTime localName="rep-cmis:creationDate"
7027 propertyDefinitionId="cmis:creationDate">
7028                 <cmis:value>2010-01-25T10:20:58.849-08:00</cmis:value>
7029             </cmis:propertyDateTime>
7030             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
7031 propertyDefinitionId="cmis:lastModificationDate">
7032                 <cmis:value>2010-01-25T10:20:58.864-08:00</cmis:value>
7033             </cmis:propertyDateTime>
7034             <cmis:propertyId localName="rep-cmis:baseTypeId"
7035 propertyDefinitionId="cmis:baseTypeId">
7036                 <cmis:value>cmis:policy</cmis:value>
7037             </cmis:propertyId>
7038             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7039 propertyDefinitionId="cmis:lastModifiedBy">
7040                 <cmis:value>Al Brown</cmis:value>
7041             </cmis:propertyString>
7042             <cmis:propertyString localName="rep-cmis:createdBy"
7043 propertyDefinitionId="cmis:createdBy">
7044                 <cmis:value>Al Brown</cmis:value>
7045             </cmis:propertyString>
7046         </cmis:properties>
7047     </cmisra:object>
7048 </atom:entry>
7049

```

Please also see the example documents included with the schema.

3.8.3.3 DELETE

This is the only collection where the URI's of the objects in the collection MUST be specific to that collection. A DELETE on the policy object in the collection is a removal of the policy from the object NOT a deletion of the policy object itself.

3.9 Feeds

For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP verb in a repository-specific manner.

3.9.1 Object Parents Feed

This is the set of parents for a specific object.

CMIS Services:

7063 GET: getObjectParents
 7064 Media Type: application/atom+xml;type=feed
 7065
 7066 Link Relations:

- 7067 • service: Points to service document containing the CMIS repository. The service document
 7068 MUST contain only one workspace element.
 - 7069 ○ Media Type: application/atomsvc+xml
- 7070 • via: points to the atom entry of object who's parents are represented by this collection
 7071

7072 This feed contains a set of atom entries for each parent of the object that MUST contain:

- 7073 • cmisra:object inside atom:entry
- 7074 • cmisra:relativePathSegment inside atom:entry for the name of the object inside the folder
 7075

7076 Example:

```

7077 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7078 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7079 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7080 xmlns:atom="http://www.w3.org/2005/Atom"
7081 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7082 open.org/ns/cmisis/restatom/200908/">
7083   <atom:title type="text">Parent Feed for 268d30b5-91a0-47f0-b985-
7084 6765e178f0bb</atom:title>
7085   <atom:author>
7086     <atom:name>Al Brown</atom:name>
7087     <atom:uri>http://www.ibm.com/</atom:uri>
7088     <atom:email>albertcbrown@us.ibm.com</atom:email>
7089   </atom:author>
7090   <atom:updated>2010-01-25T10:20:59.818-08:00</atom:updated>
7091   <atom:id>urn:uuid:6f541940-4abf-471b-99f0-8e6f66d53789</atom:id>
7092   <atom:link type="application/atom+xml;type=feed" rel="self"
7093 href="http://cmisexample.oasis-open.org/repl/268d30b5-91a0-47f0-b985-
7094 6765e178f0bb/3"/>
7095   <atom:link type="application/atomsvc+xml" rel="service"
7096 href="http://cmisexample.oasis-open.org/repl//service"/>
7097   <atom:link type="application/atom+xml;type=entry" rel="via"
7098 href="http://cmisexample.oasis-open.org/repl/268d30b5-91a0-47f0-b985-
7099 6765e178f0bb"/>
7100   <atom:link type="application/atom+xml;type=feed" rel="first"
7101 href="http://cmisexample.oasis-open.org/repl/268d30b5-91a0-47f0-b985-
7102 6765e178f0bb/first"/>
7103   <atom:link type="application/atom+xml;type=feed" rel="next"
7104 href="http://cmisexample.oasis-open.org/repl/268d30b5-91a0-47f0-b985-
7105 6765e178f0bb/4"/>
7106   <atom:link type="application/atom+xml;type=feed" rel="previous"
7107 href="http://cmisexample.oasis-open.org/repl/268d30b5-91a0-47f0-b985-
7108 6765e178f0bb/2"/>
7109   <atom:link type="application/atom+xml;type=feed" rel="last"
7110 href="http://cmisexample.oasis-open.org/repl/268d30b5-91a0-47f0-b985-
7111 6765e178f0bb/last"/>
7112   <cmisra:numItems>1</cmisra:numItems>
7113   <atom:entry>
7114     <atom:author>
7115       <atom:name>Al Brown</atom:name>
7116       <atom:uri>http://www.ibm.com/</atom:uri>
7117       <atom:email>albertcbrown@us.ibm.com</atom:email>
7118     </atom:author>

```



```

7119     <atom:content src="http://cmisexample.oasis-open.org/rep1/661d6945-
7120 8f75-4dea-8799-7ba07b0e510e"/>
7121     <atom:id>urn:uuid:661d6945-8f75-4dea-8799-7ba07b0e510e</atom:id>
7122     <atom:title type="text">Customer Folder</atom:title>
7123     <atom:updated>2010-01-25T10:20:59.833-08:00</atom:updated>
7124     <atom:link rel="self" href="http://cmisexample.oasis-
7125 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
7126     <atom:link rel="edit" href="http://cmisexample.oasis-
7127 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
7128     <atom:link type="application/cmismime+xml;type=allowableActions"
7129 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7130 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7131 7ba07b0e510e/allowableactions"/>
7132     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7133 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7134 7ba07b0e510e/type"/>
7135     <atom:link type="application/atomsvc+xml" rel="service"
7136 href="http://cmisexample.oasis-open.org/rep1/service"/>
7137     <atom:published>2010-01-25T10:20:59.833-08:00</atom:published>
7138     <atom:summary type="html">HTML summary of Entry 661d6945-8f75-4dea-
7139 8799-7ba07b0e510e</atom:summary>
7140     <atom:link type="application/atom+xml;type=entry" rel="up"
7141 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7142 7ba07b0e510e/up"/>
7143     <atom:link type="application/atom+xml;type=feed" rel="down"
7144 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7145 7ba07b0e510e/children"/>
7146     <atom:link type="application/cmistree+xml" rel="down"
7147 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7148 7ba07b0e510e/tree"/>
7149     <atom:link type="application/atom+xml;type=feed"
7150 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7151 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7152 7ba07b0e510e/foldertree"/>
7153     <atom:link type="application/atom+xml;type=feed"
7154 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7155 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7156 7ba07b0e510e/relationships"/>
7157     <atom:link type="application/atom+xml;type=feed"
7158 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7159 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7160 7ba07b0e510e/policies"/>
7161     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7162 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7163 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e/acl"/>
7164     <cmisra:object>
7165       <cmis:properties>
7166         <cmis:propertyId localName="rep-cmis:objectId"
7167 propertyDefinitionId="cmis:objectId">
7168           <cmis:value>661d6945-8f75-4dea-8799-
7169 7ba07b0e510e</cmis:value>
7170         </cmis:propertyId>
7171         <cmis:propertyId localName="rep-cmis:objectTypeId"
7172 propertyDefinitionId="cmis:objectTypeId">
7173           <cmis:value>customer</cmis:value>
7174         </cmis:propertyId>
7175         <cmis:propertyString localName="rep-cmis:name"
7176 propertyDefinitionId="cmis:name">
7177           <cmis:value>Customer Folder</cmis:value>
7178         </cmis:propertyString>
7179         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7180 propertyDefinitionId="cmis:creationDate">
7181           <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
7182         </cmis:propertyDateTime>

```

```

7183         <cmis:propertyDateTime localName="rep-
7184 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7185         <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
7186         </cmis:propertyDateTime>
7187         <cmis:propertyId localName="rep-cmis:baseTypeId"
7188 propertyDefinitionId="cmis:baseTypeId">
7189         <cmis:value>cmis:folder</cmis:value>
7190         </cmis:propertyId>
7191         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7192 propertyDefinitionId="cmis:lastModifiedBy">
7193         <cmis:value>Al Brown</cmis:value>
7194         </cmis:propertyString>
7195         <cmis:propertyString localName="rep-cmis:createdBy"
7196 propertyDefinitionId="cmis:createdBy">
7197         <cmis:value>Al Brown</cmis:value>
7198         </cmis:propertyString>
7199         <cmis:propertyId localName="rep-cmis:parentId"
7200 propertyDefinitionId="cmis:parentId">
7201         <cmis:value>661d6945-8f75-4dea-8799-
7202 7ba07b0e510eup</cmis:value>
7203         </cmis:propertyId>
7204         </cmis:properties>
7205     </cmisra:object>
7206     <cmisra:relativePathSegment>customer1</cmisra:relativePathSegment>
7207 </atom:entry>
7208 </atom:feed>

```

Please also see the example documents included with the schema.

3.9.1.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions
- includeRelationships
- renditionFilter
- includeRelativePathSegment
 - If true, then the cmisra:relativePathSegment element MUST be included in the response.

3.9.2 Changes

This is a link relationship described in the service document that contains the changes in the repository in the workspace element. The link relation pointing to this feed is <http://docs.oasis-open.org/ns/cmis/link/200908/changes>.

The ChangeLog Token is specified in the URI specified by the paging link notations. Through this binding it is not possible to retrieve the ChangeLog Token from the URIs.

CMIS Services:

GET: getContentChanges()

Media Type: application/atom+xml;type=feed

Link Relations:

- service: Points to service document containing the CMIS repository. The service document MUST contain only one workspace element.

- 7233 ○ Media Type: application/atomsvc+xml
- 7234 • paging link relations as appropriate: first, next, previous, last
- 7235 ○ ChangeLogToken is incorporated into the URI specified by the next link relation
- 7236
- 7237 This feed MUST be ordered from oldest first to newest.
- 7238
- 7239 If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is
- 7240 not available, the client should revisit the feed in the future and look for new items and the next link
- 7241 relation.
- 7242
- 7243 The following CMIS Atom extension element MAY be included inside the atom feed:
- 7244 • cmisra:numItems
- 7245
- 7246 The following CMIS Atom extension element MUST be included inside the atom entries:
- 7247 • cmisra:object inside atom:entry
- 7248

7249 Example:

```

7250 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7251 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7252 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7253 xmlns:atom="http://www.w3.org/2005/Atom"
7254 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7255 open.org/ns/cmis/restatom/200908/">
7256   <atom:title type="text">changelog feed</atom:title>
7257   <atom:author>
7258     <atom:name>Al Brown</atom:name>
7259     <atom:uri>http://www.ibm.com/</atom:uri>
7260     <atom:email>albertcbrown@us.ibm.com</atom:email>
7261   </atom:author>
7262   <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7263   <atom:id>urn:uuid:0bfc5306-fc76-4cd8-a0c0-7653dd43c0ff</atom:id>
7264   <atom:link type="application/atom+xml;type=feed" rel="self"
7265 href="http://cmisexample.oasis-open.org/repl/oId/3"/>
7266   <atom:link type="application/atomsvc+xml" rel="service"
7267 href="http://cmisexample.oasis-open.org/repl//service"/>
7268   <atom:link type="application/atom+xml;type=feed" rel="first"
7269 href="http://cmisexample.oasis-open.org/repl/oId/first"/>
7270   <atom:link type="application/atom+xml;type=feed" rel="next"
7271 href="http://cmisexample.oasis-open.org/repl/oId/4"/>
7272   <atom:link type="application/atom+xml;type=feed" rel="previous"
7273 href="http://cmisexample.oasis-open.org/repl/oId/2"/>
7274   <atom:link type="application/atom+xml;type=feed" rel="last"
7275 href="http://cmisexample.oasis-open.org/repl/oId/last"/>
7276   <cmisra:numItems>2</cmisra:numItems>
7277   <atom:entry>
7278     <atom:author>
7279       <atom:name>Al Brown</atom:name>
7280       <atom:uri>http://www.ibm.com/</atom:uri>
7281       <atom:email>albertcbrown@us.ibm.com</atom:email>
7282     </atom:author>
7283     <atom:content src="http://cmisexample.oasis-open.org/repl/3f724c1d-
7284 12c8-43f2-919f-674df52b6ebd"/>
7285     <atom:id>urn:uuid:3f724c1d-12c8-43f2-919f-674df52b6ebd</atom:id>
7286     <atom:title type="text">CMIS Example Folder as Customer Policy
7287 type</atom:title>
7288     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>

```

```

7289         <atom:link rel="self" href="http://cmisexample.oasis-
7290 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
7291         <atom:link rel="edit" href="http://cmisexample.oasis-
7292 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
7293         <atom:link type="application/cmismedia+xml;type=allowableActions"
7294 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7295 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7296 674df52b6ebd/allowableactions"/>
7297         <atom:link type="application/atom+xml;type=entry" rel="describedby"
7298 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7299 674df52b6ebd/type"/>
7300         <atom:link type="application/atomsvc+xml" rel="service"
7301 href="http://cmisexample.oasis-open.org/rep1//service"/>
7302         <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
7303         <atom:summary type="html">HTML summary of Entry 3f724c1d-12c8-43f2-
7304 919f-674df52b6ebd</atom:summary>
7305         <atom:link type="application/atom+xml;type=entry" rel="up"
7306 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7307 674df52b6ebd/up"/>
7308         <atom:link type="application/atom+xml;type=feed" rel="down"
7309 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7310 674df52b6ebd/children"/>
7311         <atom:link type="application/cmistree+xml" rel="down"
7312 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7313 674df52b6ebd/tree"/>
7314         <atom:link type="application/atom+xml;type=feed"
7315 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7316 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7317 674df52b6ebd/foldertree"/>
7318         <atom:link type="application/atom+xml;type=feed"
7319 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7320 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7321 674df52b6ebd/relationships"/>
7322         <atom:link type="application/atom+xml;type=feed"
7323 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7324 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7325 674df52b6ebd/policies"/>
7326         <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7327 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7328 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd/acl"/>
7329         <cmisra:object>
7330             <cmis:properties>
7331                 <cmis:propertyId localName="rep-cmis:objectId">
7332 propertyDefinitionId="cmis:objectId">
7333                     <cmis:value>3f724c1d-12c8-43f2-919f-
7334 674df52b6ebd</cmis:value>
7335                 </cmis:propertyId>
7336                 <cmis:propertyId localName="rep-cmis:objectTypeId">
7337 propertyDefinitionId="cmis:objectTypeId">
7338                     <cmis:value>customerpolicy</cmis:value>
7339                 </cmis:propertyId>
7340                 <cmis:propertyString localName="rep-cmis:name">
7341 propertyDefinitionId="cmis:name">
7342                     <cmis:value>CMIS Example Folder as Customer Policy
7343 type</cmis:value>
7344                 </cmis:propertyString>
7345                 <cmis:propertyDateTime localName="rep-cmis:creationDate">
7346 propertyDefinitionId="cmis:creationDate">
7347                     <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
7348                 </cmis:propertyDateTime>
7349                 <cmis:propertyDateTime localName="rep-
7350 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7351                     <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
7352                 </cmis:propertyDateTime>

```

```

7353         <cmis:propertyId localName="rep-cmis:baseTypeId"
7354 propertyDefinitionId="cmis:baseTypeId">
7355         <cmis:value>cmis:folder</cmis:value>
7356     </cmis:propertyId>
7357     <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7358 propertyDefinitionId="cmis:lastModifiedBy">
7359         <cmis:value>Al Brown</cmis:value>
7360     </cmis:propertyString>
7361     <cmis:propertyString localName="rep-cmis:createdBy"
7362 propertyDefinitionId="cmis:createdBy">
7363         <cmis:value>Al Brown</cmis:value>
7364     </cmis:propertyString>
7365     <cmis:propertyId localName="rep-cmis:parentId"
7366 propertyDefinitionId="cmis:parentId">
7367         <cmis:value>3f724c1d-12c8-43f2-919f-
7368 674df52b6ebdup</cmis:value>
7369     </cmis:propertyId>
7370 </cmis:properties>
7371 <cmis:changeEventInfo>
7372     <cmis:changeType>updated</cmis:changeType>
7373     <cmis:changeTime>2010-01-25T10:20:59.255-
7374 08:00</cmis:changeTime>
7375 </cmis:changeEventInfo>
7376 </cmisra:object>
7377 <cmisra:pathSegment>policy</cmisra:pathSegment>
7378 </atom:entry>
7379 <atom:entry>
7380     <atom:author>
7381         <atom:name>Al Brown</atom:name>
7382         <atom:uri>http://www.ibm.com/</atom:uri>
7383         <atom:email>albertcbrown@us.ibm.com</atom:email>
7384     </atom:author>
7385     <atom:content src="http://cmisexample.oasis-open.org/rep1/6e27bada-
7386 b5a2-4a39-be2c-269806eb0d42"/>
7387     <atom:id>urn:uuid:6e27bada-b5a2-4a39-be2c-269806eb0d42</atom:id>
7388     <atom:title type="text">CMIS Example Document</atom:title>
7389     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7390     <atom:link rel="self" href="http://cmisexample.oasis-
7391 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"/>
7392     <atom:link rel="edit" href="http://cmisexample.oasis-
7393 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"/>
7394     <atom:link type="application/cmismedia+xml;type=allowableActions"
7395 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/allowableactions"
7396 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7397 269806eb0d42/allowableactions"/>
7398     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7399 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7400 269806eb0d42/type"/>
7401     <atom:link type="application/atomsvc+xml" rel="service"
7402 href="http://cmisexample.oasis-open.org/rep1//service"/>
7403     <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
7404     <atom:summary type="html">HTML summary of Entry 6e27bada-b5a2-4a39-
7405 be2c-269806eb0d42</atom:summary>
7406     <atom:link rel="edit-media" href="http://cmisexample.oasis-
7407 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/edit-media"/>
7408     <atom:link rel="alternate" href="http://cmisexample.oasis-
7409 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/alternate"/>
7410     <atom:link type="application/atom+xml;type=feed" rel="up"
7411 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7412 269806eb0d42/parents"/>
7413     <atom:link type="application/atom+xml;type=feed" rel="version-history"
7414 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7415 269806eb0d42/allversions"/>

```

```

7416     <atom:link type="application/atom+xml;type=entry" rel="current-
7417 version" href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7418 269806eb0d42/latest"/>
7419     <atom:link type="application/atom+xml;type=feed"
7420 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7421 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7422 269806eb0d42/relationships"/>
7423     <atom:link type="application/atom+xml;type=feed"
7424 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7425 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7426 269806eb0d42/policies"/>
7427     <atom:link type="application/cmisac+xml" rel="http://docs.oasis-
7428 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7429 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/acl"/>
7430     <cmisra:object>
7431       <cmis:properties>
7432         <cmis:propertyId localName="rep-cmis:objectId">
7433 propertyDefinitionId="cmis:objectId">
7434           <cmis:value>6e27bada-b5a2-4a39-be2c-
7435 269806eb0d42</cmis:value>
7436         </cmis:propertyId>
7437         <cmis:propertyId localName="rep-cmis:objectTypeId">
7438 propertyDefinitionId="cmis:objectTypeId">
7439           <cmis:value>document</cmis:value>
7440         </cmis:propertyId>
7441         <cmis:propertyString localName="rep-cmis:name">
7442 propertyDefinitionId="cmis:name">
7443           <cmis:value>CMIS Example Document</cmis:value>
7444         </cmis:propertyString>
7445         <cmis:propertyDateTime localName="rep-cmis:creationDate">
7446 propertyDefinitionId="cmis:creationDate">
7447           <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
7448         </cmis:propertyDateTime>
7449         <cmis:propertyDateTime localName="rep-
7450 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7451           <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
7452         </cmis:propertyDateTime>
7453         <cmis:propertyId localName="rep-cmis:baseTypeId">
7454 propertyDefinitionId="cmis:baseTypeId">
7455           <cmis:value>cmis:document</cmis:value>
7456         </cmis:propertyId>
7457         <cmis:propertyString localName="rep-cmis:lastModifiedBy">
7458 propertyDefinitionId="cmis:lastModifiedBy">
7459           <cmis:value>Al Brown</cmis:value>
7460         </cmis:propertyString>
7461         <cmis:propertyString localName="rep-cmis:createdBy">
7462 propertyDefinitionId="cmis:createdBy">
7463           <cmis:value>Al Brown</cmis:value>
7464         </cmis:propertyString>
7465         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion">
7466 propertyDefinitionId="cmis:isLatestVersion">
7467           <cmis:value>true</cmis:value>
7468         </cmis:propertyBoolean>
7469         <cmis:propertyBoolean localName="rep-
7470 cmis:isVersionSeriesCheckedOut"
7471 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7472           <cmis:value>false</cmis:value>
7473         </cmis:propertyBoolean>
7474         <cmis:propertyBoolean localName="rep-cmis:isMajorVersion">
7475 propertyDefinitionId="cmis:isMajorVersion">
7476           <cmis:value>false</cmis:value>
7477         </cmis:propertyBoolean>
7478         <cmis:propertyBoolean localName="rep-
7479 cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">

```

```

7480         <cmis:value>false</cmis:value>
7481     </cmis:propertyBoolean>
7482     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7483 propertyDefinitionId="cmis:isImmutable">
7484         <cmis:value>false</cmis:value>
7485     </cmis:propertyBoolean>
7486     <cmis:propertyString localName="rep-cmis:checkinComment"
7487 propertyDefinitionId="cmis:checkinComment">
7488         <cmis:value>Checkin comment</cmis:value>
7489     </cmis:propertyString>
7490     <cmis:propertyString localName="rep-cmis:versionLabel"
7491 propertyDefinitionId="cmis:versionLabel">
7492         <cmis:value>0.1</cmis:value>
7493     </cmis:propertyString>
7494     <cmis:propertyString localName="rep-
7495 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
7496         <cmis:value>text/plain</cmis:value>
7497     </cmis:propertyString>
7498     <cmis:propertyString localName="rep-
7499 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
7500         <cmis:value>text.txt</cmis:value>
7501     </cmis:propertyString>
7502     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7503 propertyDefinitionId="cmis:contentStreamLength">
7504         <cmis:value>4234</cmis:value>
7505     </cmis:propertyInteger>
7506     <cmis:propertyString displayName="Keywords for Document"
7507 localName="keywords" propertyDefinitionId="keywords">
7508         <cmis:value>document</cmis:value>
7509         <cmis:value>example</cmis:value>
7510         <cmis:value>sample</cmis:value>
7511         <cmis:value>cmis</cmis:value>
7512     </cmis:propertyString>
7513 </cmis:properties>
7514 <cmis:changeEventInfo>
7515     <cmis:changeType>updated</cmis:changeType>
7516     <cmis:changeTime>2010-01-25T10:20:59.271-
7517 08:00</cmis:changeTime>
7518 </cmis:changeEventInfo>
7519 </cmisra:object>
7520 <cmisra:pathSegment>invoice.pdf</cmisra:pathSegment>
7521 </atom:entry>
7522 </atom:feed>

```

Please also see the example documents included with the schema.

3.9.2.1 GET

The following optional parameters may be supplied:

- filter
- maxItems
- includeACL
- includePolicyIds
- includeProperties
- changeLogToken: If this parameter is specified, start the changes from the specified token. The changeLogToken is embedded in the paging link relations for normal iteration through the change list.

3.9.3 Folder Descendants

This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available via the link relation down with the application/cmistree+xml media type. Please see the Hierarchical Atom Entries for more information on format.

If a repository does not support capabilityGetDescendants, then these resources SHOULD NOT be exposed.

CMIS Services:

GET: getDescendants

DELETE: deleteTree

Media Type: application/atom+xml;type=feed

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

The following CMIS Atom extension element MAY be included inside the atom feed:

- cmisra:numItems

The following CMIS Atom extension element MUST be included inside the atom entries:

- cmisra:object inside atom:entry
- cmisra:pathSegment inside atom:entry
- cmisra:children inside atom:entry

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/">
```



```

7580     <atom:title type="text">Feed for folder1</atom:title>
7581     <atom:author>
7582         <atom:name>Al Brown</atom:name>
7583         <atom:uri>http://www.ibm.com/</atom:uri>
7584         <atom:email>albertcbrown@us.ibm.com</atom:email>
7585     </atom:author>
7586     <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
7587     <atom:id>urn:uuid:cb0a47d4-8d09-46f9-9b09-584acad684af</atom:id>
7588     <atom:link type="application/atom+xml;type=feed" rel="self"
7589 href="http://cmisexample.oasis-open.org/repl/f083dd6f-1465-4516-97ce-
7590 040ec0c7c05a/3"/>
7591     <atom:link type="application/atomsvc+xml" rel="service"
7592 href="http://cmisexample.oasis-open.org/repl//service"/>
7593     <atom:link type="application/atom+xml;type=entry" rel="via"
7594 href="http://cmisexample.oasis-open.org/repl/f083dd6f-1465-4516-97ce-
7595 040ec0c7c05a"/>
7596     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7597 open.org/ns/cmisis/link/200908/foldertree" href="http://cmisexample.oasis-
7598 open.org/repl/f083dd6f-1465-4516-97ce-040ec0c7c05a/foldertree"/>
7599     <atom:link type="application/atom+xml;type=feed" rel="down"
7600 href="http://cmisexample.oasis-open.org/repl/f083dd6f-1465-4516-97ce-
7601 040ec0c7c05a/children"/>
7602     <atom:link type="application/atom+xml;type=entry" rel="up"
7603 href="http://cmisexample.oasis-open.org/repl/03dcf5b8-5f82-45a1-b276-
7604 44d88069eec3"/>
7605     <cmisra:numItems>1</cmisra:numItems>
7606     <atom:entry>
7607         <atom:author>
7608             <atom:name>Al Brown</atom:name>
7609             <atom:uri>http://www.ibm.com/</atom:uri>
7610             <atom:email>albertcbrown@us.ibm.com</atom:email>
7611         </atom:author>
7612         <atom:content src="http://cmisexample.oasis-open.org/repl/8e5a512c-
7613 8f2d-4387-a283-f3f30bbc312f"/>
7614         <atom:id>urn:uuid:8e5a512c-8f2d-4387-a283-f3f30bbc312f</atom:id>
7615         <atom:title type="text">CMIS Example Folder as Customer
7616 type</atom:title>
7617         <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
7618         <atom:link rel="self" href="http://cmisexample.oasis-
7619 open.org/repl/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
7620         <atom:link rel="edit" href="http://cmisexample.oasis-
7621 open.org/repl/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
7622         <atom:link type="application/cmisis+xml;type=allowableActions"
7623 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7624 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7625 f3f30bbc312f/allowableactions"/>
7626         <atom:link type="application/atom+xml;type=entry" rel="describedby"
7627 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7628 f3f30bbc312f/type"/>
7629         <atom:link type="application/atomsvc+xml" rel="service"
7630 href="http://cmisexample.oasis-open.org/repl//service"/>
7631         <atom:published>2010-01-25T10:20:59.380-08:00</atom:published>
7632         <atom:summary type="html">HTML summary of Entry 8e5a512c-8f2d-4387-
7633 a283-f3f30bbc312f</atom:summary>
7634         <atom:link type="application/atom+xml;type=entry" rel="up"
7635 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7636 f3f30bbc312f/up"/>
7637         <atom:link type="application/atom+xml;type=feed" rel="down"
7638 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7639 f3f30bbc312f/children"/>
7640         <atom:link type="application/cmistree+xml" rel="down"
7641 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7642 f3f30bbc312f/tree"/>

```

```

7643     <atom:link type="application/atom+xml;type=feed"
7644 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7645 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7646 f3f30bbc312f/foldertree"/>
7647     <atom:link type="application/atom+xml;type=feed"
7648 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7649 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7650 f3f30bbc312f/relationships"/>
7651     <atom:link type="application/atom+xml;type=feed"
7652 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7653 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7654 f3f30bbc312f/policies"/>
7655     <atom:link type="application/cmisac+xml" rel="http://docs.oasis-
7656 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
7657 open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f/acl"/>
7658     <cmisra:object>
7659       <cmis:properties>
7660         <cmis:propertyId localName="rep-cmis:objectId"
7661 propertyDefinitionId="cmis:objectId">
7662           <cmis:value>8e5a512c-8f2d-4387-a283-
7663 f3f30bbc312f</cmis:value>
7664         </cmis:propertyId>
7665         <cmis:propertyId localName="rep-cmis:objectTypeId"
7666 propertyDefinitionId="cmis:objectTypeId">
7667           <cmis:value>customer</cmis:value>
7668         </cmis:propertyId>
7669         <cmis:propertyString localName="rep-cmis:name"
7670 propertyDefinitionId="cmis:name">
7671           <cmis:value>CMIS Example Folder as Customer
7672 type</cmis:value>
7673         </cmis:propertyString>
7674         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7675 propertyDefinitionId="cmis:creationDate">
7676           <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7677         </cmis:propertyDateTime>
7678         <cmis:propertyDateTime localName="rep-
7679 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7680           <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7681         </cmis:propertyDateTime>
7682         <cmis:propertyId localName="rep-cmis:baseTypeId"
7683 propertyDefinitionId="cmis:baseTypeId">
7684           <cmis:value>cmis:folder</cmis:value>
7685         </cmis:propertyId>
7686         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7687 propertyDefinitionId="cmis:lastModifiedBy">
7688           <cmis:value>Al Brown</cmis:value>
7689         </cmis:propertyString>
7690         <cmis:propertyString localName="rep-cmis:createdBy"
7691 propertyDefinitionId="cmis:createdBy">
7692           <cmis:value>Al Brown</cmis:value>
7693         </cmis:propertyString>
7694         <cmis:propertyId localName="rep-cmis:parentId"
7695 propertyDefinitionId="cmis:parentId">
7696           <cmis:value>8e5a512c-8f2d-4387-a283-
7697 f3f30bbc312fup</cmis:value>
7698         </cmis:propertyId>
7699       </cmis:properties>
7700     </cmisra:object>
7701     <cmisra:pathSegment>customer</cmisra:pathSegment>
7702     <cmisra:children>
7703       <atom:feed>
7704         <atom:title type="text">CMIS Example Folder as Customer
7705 type</atom:title>
7706         <atom:author>

```

```

7707         <atom:name>Al Brown</atom:name>
7708         <atom:uri>http://www.ibm.com/</atom:uri>
7709         <atom:email>albertcbrown@us.ibm.com</atom:email>
7710     </atom:author>
7711     <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
7712     <atom:id>urn:uuid:67ee5e9f-d2e3-457d-9dec-
7713 be718e780452</atom:id>
7714     <atom:link type="application/atom+xml;type=feed" rel="self"
7715 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7716 f3f30bbc312f/3"/>
7717     <atom:link type="application/atomsvc+xml" rel="service"
7718 href="http://cmisexample.oasis-open.org/repl//service"/>
7719     <atom:link type="application/atom+xml;type=entry" rel="via"
7720 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7721 f3f30bbc312f"/>
7722     <atom:link type="application/atom+xml;type=feed"
7723 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
7724 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7725 f3f30bbc312f/foldertree"/>
7726     <atom:link type="application/atom+xml;type=feed" rel="down"
7727 href="http://cmisexample.oasis-open.org/repl/8e5a512c-8f2d-4387-a283-
7728 f3f30bbc312f/children"/>
7729     <atom:link type="application/atom+xml;type=entry" rel="up"
7730 href="http://cmisexample.oasis-open.org/repl/f083dd6f-1465-4516-97ce-
7731 040ec0c7c05a"/>
7732     <cmisra:numItems>1</cmisra:numItems>
7733     <atom:entry>
7734         <atom:author>
7735             <atom:name>Al Brown</atom:name>
7736             <atom:uri>http://www.ibm.com/</atom:uri>
7737             <atom:email>albertcbrown@us.ibm.com</atom:email>
7738         </atom:author>
7739         <atom:content src="http://cmisexample.oasis-
7740 open.org/repl/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7741         <atom:id>urn:uuid:8c2dbba5-ea33-469f-a052-
7742 9f01e636c72a</atom:id>
7743         <atom:title type="text">CMIS Example Doc as Invoice
7744 type</atom:title>
7745         <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
7746         <atom:link rel="self" href="http://cmisexample.oasis-
7747 open.org/repl/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7748         <atom:link rel="edit" href="http://cmisexample.oasis-
7749 open.org/repl/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7750         <atom:link
7751 type="application/cmis+xml;type=allowableActions" rel="http://docs.oasis-
7752 open.org/ns/cmis/link/200908/allowableactions" href="http://cmisexample.oasis-
7753 open.org/repl/8c2dbba5-ea33-469f-a052-9f01e636c72a/allowableactions"/>
7754         <atom:link type="application/atom+xml;type=entry"
7755 rel="describedby" href="http://cmisexample.oasis-open.org/repl/8c2dbba5-ea33-
7756 469f-a052-9f01e636c72a/type"/>
7757         <atom:link type="application/atomsvc+xml" rel="service"
7758 href="http://cmisexample.oasis-open.org/repl//service"/>
7759         <atom:published>2010-01-25T10:20:59.380-
7760 08:00</atom:published>
7761         <atom:summary type="html">HTML summary of Entry 8c2dbba5-
7762 ea33-469f-a052-9f01e636c72a</atom:summary>
7763         <atom:link rel="edit-media"
7764 href="http://cmisexample.oasis-open.org/repl/8c2dbba5-ea33-469f-a052-
7765 9f01e636c72a/edit-media"/>
7766         <atom:link rel="alternate" href="http://cmisexample.oasis-
7767 open.org/repl/8c2dbba5-ea33-469f-a052-9f01e636c72a/alternate"/>
7768         <atom:link type="application/atom+xml;type=feed" rel="up"
7769 href="http://cmisexample.oasis-open.org/repl/8c2dbba5-ea33-469f-a052-
7770 9f01e636c72a/parents"/>

```

```

7771         <atom:link type="application/atom+xml;type=feed"
7772 rel="version-history" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-
7773 ea33-469f-a052-9f01e636c72a/allversions"/>
7774         <atom:link type="application/atom+xml;type=entry"
7775 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-
7776 ea33-469f-a052-9f01e636c72a/latest"/>
7777         <atom:link type="application/atom+xml;type=feed"
7778 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
7779 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7780 9f01e636c72a/relationships"/>
7781         <atom:link type="application/atom+xml;type=feed"
7782 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
7783 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7784 9f01e636c72a/policies"/>
7785         <atom:link type="application/cmisacl+xml"
7786 rel="http://docs.oasis-open.org/ns/cmis/link/200908/acl"
7787 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7788 9f01e636c72a/acl"/>
7789         <cmisra:object>
7790             <cmis:properties>
7791                 <cmis:propertyId localName="rep-cmis:objectId"
7792 propertyDefinitionId="cmis:objectId">
7793 <cmis:value>8c2dbba5-ea33-469f-a052-9f01e636c72a</cmis:value>
7794 </cmis:propertyId>
7795                 <cmis:propertyId localName="rep-cmis:objectTypeId"
7796 propertyDefinitionId="cmis:objectTypeId">
7797 <cmis:value>invoice</cmis:value>
7798 </cmis:propertyId>
7799                 <cmis:propertyString localName="rep-cmis:name"
7800 propertyDefinitionId="cmis:name">
7801 <cmis:value>CMIS Example Doc as Invoice type</cmis:value>
7802 </cmis:propertyString>
7803                 <cmis:propertyDateTime localName="rep-
7804 cmis:creationDate" propertyDefinitionId="cmis:creationDate">
7805 <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7806 </cmis:propertyDateTime>
7807                 <cmis:propertyDateTime localName="rep-
7808 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7809 <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7810 </cmis:propertyDateTime>
7811                 <cmis:propertyId localName="rep-cmis:baseTypeId"
7812 propertyDefinitionId="cmis:baseTypeId">
7813 <cmis:value>cmis:document</cmis:value>
7814 </cmis:propertyId>
7815                 <cmis:propertyString localName="rep-
7816 cmis:lastModifiedBy" propertyDefinitionId="cmis:lastModifiedBy">
7817 <cmis:value>Al Brown</cmis:value>
7818 </cmis:propertyString>
7819                 <cmis:propertyString localName="rep-
7820 cmis:createdBy" propertyDefinitionId="cmis:createdBy">
7821 <cmis:value>Al Brown</cmis:value>
7822 </cmis:propertyString>
7823                 <cmis:propertyBoolean localName="rep-
7824 cmis:isLatestVersion" propertyDefinitionId="cmis:isLatestVersion">
7825 <cmis:value>true</cmis:value>
7826 </cmis:propertyBoolean>
7827                 <cmis:propertyBoolean localName="rep-
7828 cmis:isVersionSeriesCheckedOut"
7829 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7830 <cmis:value>false</cmis:value>
7831 </cmis:propertyBoolean>
7832                 <cmis:propertyBoolean localName="rep-
7833 cmis:isMajorVersion" propertyDefinitionId="cmis:isMajorVersion">
7834 <cmis:value>false</cmis:value>

```

```

7835         </cmis:propertyBoolean>
7836         <cmis:propertyBoolean localName="rep-
7837 cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
7838 <cmis:value>>false</cmis:value>
7839         </cmis:propertyBoolean>
7840         <cmis:propertyBoolean localName="rep-
7841 cmis:isImmutable" propertyDefinitionId="cmis:isImmutable">
7842 <cmis:value>>false</cmis:value>
7843         </cmis:propertyBoolean>
7844         <cmis:propertyString localName="rep-
7845 cmis:checkinComment" propertyDefinitionId="cmis:checkinComment">
7846 <cmis:value>Checkin comment</cmis:value>
7847         </cmis:propertyString>
7848         <cmis:propertyString localName="rep-
7849 cmis:versionLabel" propertyDefinitionId="cmis:versionLabel">
7850 <cmis:value>0.1</cmis:value>
7851         </cmis:propertyString>
7852         <cmis:propertyString localName="rep-
7853 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
7854 <cmis:value>text/plain</cmis:value>
7855         </cmis:propertyString>
7856         <cmis:propertyString localName="rep-
7857 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
7858 <cmis:value>text.txt</cmis:value>
7859         </cmis:propertyString>
7860         <cmis:propertyInteger localName="rep-
7861 cmis:contentStreamLength" propertyDefinitionId="cmis:contentStreamLength">
7862 <cmis:value>4234</cmis:value>
7863         </cmis:propertyInteger>
7864         <cmis:propertyString displayName="Keywords for
7865 Document" localName="keywords" propertyDefinitionId="keywords">
7866 <cmis:value>document</cmis:value>
7867 <cmis:value>example</cmis:value>
7868 <cmis:value>sample</cmis:value>
7869 <cmis:value>cmis</cmis:value>
7870         </cmis:propertyString>
7871     </cmis:properties>
7872 </cmisra:object>
7873 <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
7874 </atom:entry>
7875 </atom:feed>
7876 </cmisra:children>
7877 </atom:entry>
7878 </atom:feed>

```

Please also see the example documents included with the schema.

3.9.3.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- depth
- includeAllowableActions
- includeRelationships
- renditionFilter
- includePathSegment

3.9.3.2 DELETE

This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the domain model for more information:

- `continueOnFailure`
- `unfileObjects`

Status Code:

- 200 OK if successful. Body contains entity describing the status
- 202 Accepted, if accepted but deletion not yet taking place
- 204 No Content, if successful with no content
- 403 Forbidden, if permission is denied
- 401 Unauthorized, if not authenticated
- 500 Internal Server Error. The body SHOULD contain an entity describing the status

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.

Note: If the repository does not implement get on this resource, or the `canGetDescendants` is false, there is no mechanism to identify the resources that were not removed.

3.9.4 Folder Tree

This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link relation `foldertree` with media type `application/atom+xml;type=feed`. Please see the Hierarchical Atom Entries for more information on format.

CMIS Services:

GET: `getFolderTree`

DELETE: `deleteTree`

Media Type: `application/atom+xml;type=feed`

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 of this folder's parent
 - If the root folder, this link relation MUST not be included.
 - Media Type: `application/atom+xml;type=entry`
- `down`:
 - `application/atom+xml` : Points to the atom feed document representing the children feed for this same folder
 - `application/cmistree+xml`: Points to the descendants feed of the same folder. If a repository does not support `capabilityGetDescendants`, then this link SHOULD NOT be included.
- paging link relations MAY be included as appropriate: `first`, `next`, `previous`, `last`

7932 ○ Repositories may support these paging link relations on a particular cmisra:children
7933 element.

7934

7935 This feed contains a set of atom entries for each sub-folder in the folder.

7936

7937 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7938 • cmisra:numItems

7939

7940 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7941 • cmisra:object inside atom:entry
- 7942 • cmisra:pathSegment inside atom:entry
- 7943 • cmisra:children inside atom:entry

7944

7945 Example:

```
7946   <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7947   <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7948   xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7949   xmlns:atom="http://www.w3.org/2005/Atom"
7950   xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7951   open.org/ns/cmis/restatom/200908/">
7952     <atom:title type="text">FolderTree Feed of Folder1</atom:title>
7953     <atom:author>
7954       <atom:name>Al Brown</atom:name>
7955       <atom:uri>http://www.ibm.com/</atom:uri>
7956       <atom:email>albertcbrown@us.ibm.com</atom:email>
7957     </atom:author>
7958     <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
7959     <atom:id>urn:uuid:f87e5678-dd24-4214-9e71-635f060beb7d</atom:id>
7960     <atom:link type="application/atom+xml;type=feed" rel="self"
7961   href="http://cmisexample.oasis-open.org/repl/6e327a3c-a246-4cee-8176-
7962   b65edc3e1854/3"/>
7963     <atom:link type="application/atomsvc+xml" rel="service"
7964   href="http://cmisexample.oasis-open.org/repl//service"/>
7965     <atom:link type="application/atom+xml;type=entry" rel="via"
7966   href="http://cmisexample.oasis-open.org/repl/6e327a3c-a246-4cee-8176-
7967   b65edc3e1854"/>
7968     <atom:link type="application/cmistree+xml" rel="down"
7969   href="http://cmisexample.oasis-open.org/repl/6e327a3c-a246-4cee-8176-
7970   b65edc3e1854/tree"/>
7971     <atom:link type="application/atom+xml;type=feed" rel="down"
7972   href="http://cmisexample.oasis-open.org/repl/6e327a3c-a246-4cee-8176-
7973   b65edc3e1854/children"/>
7974     <atom:link type="application/atom+xml;type=entry" rel="up"
7975   href="http://cmisexample.oasis-open.org/repl/3056c4d7-4e16-49cb-a750-
7976   ad7a3844alaa"/>
7977     <cmisra:numItems>1</cmisra:numItems>
7978     <atom:entry>
7979       <atom:author>
7980         <atom:name>Al Brown</atom:name>
7981         <atom:uri>http://www.ibm.com/</atom:uri>
7982         <atom:email>albertcbrown@us.ibm.com</atom:email>
7983       </atom:author>
7984       <atom:content src="http://cmisexample.oasis-open.org/repl/c7b5a83e-
7985   37b6-4f5a-b646-50892252a180"/>
7986       <atom:id>urn:uuid:c7b5a83e-37b6-4f5a-b646-50892252a180</atom:id>
7987       <atom:title type="text">Customer Folder</atom:title>
7988       <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
```

```

7989     <atom:link rel="self" href="http://cmisexample.oasis-
7990 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
7991     <atom:link rel="edit" href="http://cmisexample.oasis-
7992 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
7993     <atom:link type="application/cmismedia+xml;type=allowableActions"
7994 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
7995 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
7996 50892252a180/allowableactions"/>
7997     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7998 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
7999 50892252a180/type"/>
8000     <atom:link type="application/atomsvc+xml" rel="service"
8001 href="http://cmisexample.oasis-open.org/rep1//service"/>
8002     <atom:published>2010-01-25T10:20:59.521-08:00</atom:published>
8003     <atom:summary type="html">HTML summary of Entry c7b5a83e-37b6-4f5a-
8004 b646-50892252a180</atom:summary>
8005     <atom:link type="application/atom+xml;type=entry" rel="up"
8006 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8007 50892252a180/up"/>
8008     <atom:link type="application/atom+xml;type=feed" rel="down"
8009 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8010 50892252a180/children"/>
8011     <atom:link type="application/cmistree+xml" rel="down"
8012 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8013 50892252a180/tree"/>
8014     <atom:link type="application/atom+xml;type=feed"
8015 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
8016 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8017 50892252a180/foldertree"/>
8018     <atom:link type="application/atom+xml;type=feed"
8019 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
8020 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8021 50892252a180/relationships"/>
8022     <atom:link type="application/atom+xml;type=feed"
8023 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
8024 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8025 50892252a180/policies"/>
8026     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8027 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
8028 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180/acl"/>
8029     <cmisra:object>
8030       <cmis:properties>
8031         <cmis:propertyId localName="rep-cmis:objectId">
8032           propertyDefinitionId="cmis:objectId">
8033             <cmis:value>c7b5a83e-37b6-4f5a-b646-
8034 50892252a180</cmis:value>
8035           </cmis:propertyId>
8036           <cmis:propertyId localName="rep-cmis:objectTypeId">
8037             propertyDefinitionId="cmis:objectTypeId">
8038               <cmis:value>customer</cmis:value>
8039             </cmis:propertyId>
8040             <cmis:propertyString localName="rep-cmis:name">
8041               propertyDefinitionId="cmis:name">
8042                 <cmis:value>Customer Folder</cmis:value>
8043               </cmis:propertyString>
8044               <cmis:propertyDateTime localName="rep-cmis:creationDate">
8045                 propertyDefinitionId="cmis:creationDate">
8046                   <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
8047                 </cmis:propertyDateTime>
8048                 <cmis:propertyDateTime localName="rep-
8049 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
8050                   <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
8051                 </cmis:propertyDateTime>

```



```

8052         <cmis:propertyId localName="rep-cmis:baseTypeId"
8053 propertyDefinitionId="cmis:baseTypeId">
8054         <cmis:value>cmis:folder</cmis:value>
8055         </cmis:propertyId>
8056         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
8057 propertyDefinitionId="cmis:lastModifiedBy">
8058         <cmis:value>Al Brown</cmis:value>
8059         </cmis:propertyString>
8060         <cmis:propertyString localName="rep-cmis:createdBy"
8061 propertyDefinitionId="cmis:createdBy">
8062         <cmis:value>Al Brown</cmis:value>
8063         </cmis:propertyString>
8064         <cmis:propertyId localName="rep-cmis:parentId"
8065 propertyDefinitionId="cmis:parentId">
8066         <cmis:value>c7b5a83e-37b6-4f5a-b646-
8067 50892252a180up</cmis:value>
8068         </cmis:propertyId>
8069         </cmis:properties>
8070     </cmisra:object>
8071     <cmisra:pathSegment>customer</cmisra:pathSegment>
8072 </atom:entry>
8073 </atom:feed>

```

Please also see the example documents included with the schema.

3.9.4.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- depth
- includeAllowableActions
- includeRelationships
- renditionFilter

3.9.4.2 DELETE

This is the same as DELETE on Folder Descendants. Please see that section.

3.9.5 AllVersions Feed

This is a feed comprised of all the versions of the given document.

CMIS Services:

GET: getAllVersions

DELETE: deleteAllVersions

Media Type: application/atom+xml;type=feed

The feed SHOULD contain the newest versions at the beginning of the feed.

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 resource generating this collection

- paging link relations as appropriate: first, next, previous, last

8100

8101 This feed contains a set of atom entries for each version in the version series

- cmisra:object inside atom:entry
- cmisra:children inside atom:entry if atom:entry represents a CMIS Folder

8104

8105 Example:

```
8106 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8107 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
8108 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
8109 xmlns:atom="http://www.w3.org/2005/Atom"
8110 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8111 open.org/ns/cmis/restatom/200908/">
8112   <atom:title type="text">AllVersions for Document e8abd7cd-b9ec-4dba-9eaa-
8113   1bce2ae9977f</atom:title>
8114   <atom:author>
8115     <atom:name>Al Brown</atom:name>
8116     <atom:uri>http://www.ibm.com/</atom:uri>
8117     <atom:email>albertcbrown@us.ibm.com</atom:email>
8118   </atom:author>
8119   <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
8120   <atom:id>urn:uuid:5dc3d1c1-3e85-4720-acf8-cf98c96a5830</atom:id>
8121   <atom:link type="application/atom+xml;type=feed" rel="self"
8122 href="http://cmisexample.oasis-open.org/repl/e8abd7cd-b9ec-4dba-9eaa-
8123 1bce2ae9977f/3"/>
8124   <atom:link type="application/atomsvc+xml" rel="service"
8125 href="http://cmisexample.oasis-open.org/repl//service"/>
8126   <atom:link type="application/atom+xml;type=entry" rel="via"
8127 href="http://cmisexample.oasis-open.org/repl/e8abd7cd-b9ec-4dba-9eaa-
8128 1bce2ae9977f"/>
8129   <cmisra:numItems>1</cmisra:numItems>
8130   <atom:entry>
8131     <atom:author>
8132       <atom:name>Al Brown</atom:name>
8133       <atom:uri>http://www.ibm.com/</atom:uri>
8134       <atom:email>albertcbrown@us.ibm.com</atom:email>
8135     </atom:author>
8136     <atom:content src="http://cmisexample.oasis-open.org/repl/197033f2-
8137 ac11-4911-b5a3-60781fa5c281"/>
8138     <atom:id>urn:uuid:197033f2-ac11-4911-b5a3-60781fa5c281</atom:id>
8139     <atom:title type="text">Invoice (Version1)</atom:title>
8140     <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
8141     <atom:link rel="self" href="http://cmisexample.oasis-
8142 open.org/repl/197033f2-ac11-4911-b5a3-60781fa5c281"/>
8143     <atom:link rel="edit" href="http://cmisexample.oasis-
8144 open.org/repl/197033f2-ac11-4911-b5a3-60781fa5c281"/>
8145     <atom:link type="application/cmis+xml;type=allowableActions"
8146 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
8147 href="http://cmisexample.oasis-open.org/repl/197033f2-ac11-4911-b5a3-
8148 60781fa5c281/allowableactions"/>
8149     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8150 href="http://cmisexample.oasis-open.org/repl/197033f2-ac11-4911-b5a3-
8151 60781fa5c281/type"/>
8152     <atom:link type="application/atomsvc+xml" rel="service"
8153 href="http://cmisexample.oasis-open.org/repl//service"/>
8154     <atom:published>2010-01-25T10:20:58.896-08:00</atom:published>
8155     <atom:summary type="html">HTML summary of Entry 197033f2-ac11-4911-
8156 b5a3-60781fa5c281</atom:summary>
8157     <atom:link rel="edit-media" href="http://cmisexample.oasis-
8158 open.org/repl/197033f2-ac11-4911-b5a3-60781fa5c281/edit-media"/>
```

```

8159     <atom:link rel="alternate" href="http://cmisexample.oasis-
8160 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/alternate"/>
8161     <atom:link type="application/atom+xml;type=feed" rel="up"
8162 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8163 60781fa5c281/parents"/>
8164     <atom:link type="application/atom+xml;type=feed" rel="version-history"
8165 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8166 60781fa5c281/allversions"/>
8167     <atom:link type="application/atom+xml;type=entry" rel="current-
8168 version" href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8169 60781fa5c281/latest"/>
8170     <atom:link type="application/atom+xml;type=feed"
8171 rel="http://docs.oasis-open.org/ns/cmis/link/200908/relationships"
8172 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8173 60781fa5c281/relationships"/>
8174     <atom:link type="application/atom+xml;type=feed"
8175 rel="http://docs.oasis-open.org/ns/cmis/link/200908/policies"
8176 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8177 60781fa5c281/policies"/>
8178     <atom:link type="application/cmisac+xml" rel="http://docs.oasis-
8179 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
8180 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/acl"/>
8181     <cmisra:object>
8182       <cmis:properties>
8183         <cmis:propertyId localName="rep-cmis:objectId"
8184 propertyDefinitionId="cmis:objectId">
8185           <cmis:value>197033f2-ac11-4911-b5a3-
8186 60781fa5c281</cmis:value>
8187         </cmis:propertyId>
8188       </cmis:properties>
8189     </cmisra:object>
8190   </atom:entry>
8191 </atom:feed>

```

Please also see the example documents included with the schema.

3.9.5.1 GET

The following arguments may be supplied. Please see the domain model for more information:

- filter
- includeAllowableActions

3.9.5.2 DELETE

This removes the entire version history of the document.

Success HTTP code: 204

3.9.6 Type Descendants Feed

This is a feed described in the service document that contains all the types under a specific type in the repository to a specific depth. If no parent type is specified, then the base types and their descendants are returned in the feed which is equivalent to all types in the repository if depth is infinite. The link relation is <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>.

Types are nested using the CMIS hierarchy extension. Please see section 3.2.3.2 Hierarchy Navigation Internet Draft Link Relations.

8211 CMIS Services:

8212 GET: getTypeDescendants

8213

8214 Media Type: application/atom+xml;type=feed

8215

8216 Link Relations:

- 8217 • service: Points to service document containing the CMIS repository. The service document
- 8218 MUST contain only one workspace element.
 - 8219 ○ Media Type: application/atomsvc+xml
- 8220 • via: points to the type definition whose descendents represent this feed. This link is not present if
- 8221 no parent type is specified.
- 8222 • down: points to the children feed for the same type
- 8223 • up: points to the parent type definition
 - 8224 ○ If this is a descendants feed for a base object type, this link is not present.

8225

8226 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8227 • cmisra:numItems

8228

8229 Example:

```
8230 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8231 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
8232 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
8233 xmlns:atom="http://www.w3.org/2005/Atom"
8234 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8235 open.org/ns/cmisis/restatom/200908/">
8236   <atom:title type="text">Base Types</atom:title>
8237   <atom:author>
8238     <atom:name>Al Brown</atom:name>
8239     <atom:uri>http://www.ibm.com/</atom:uri>
8240     <atom:email>albertcbrown@us.ibm.com</atom:email>
8241   </atom:author>
8242   <atom:updated>2010-01-25T10:20:59.911-08:00</atom:updated>
8243   <atom:id>urn:uuid:c5d3d357-33ec-47c1-8436-563e0d94d2e5</atom:id>
8244   <atom:link type="application/atom+xml;type=feed" rel="self"
8245 href="http://cmisexample.oasis-open.org/repl//3"/>
8246   <atom:link type="application/atomsvc+xml" rel="service"
8247 href="http://cmisexample.oasis-open.org/repl//service"/>
8248   <atom:link type="application/atom+xml;type=entry" rel="via"
8249 href="http://cmisexample.oasis-open.org/repl/" />
8250   <atom:link type="application/atom+xml;type=feed" rel="down"
8251 href="http://cmisexample.oasis-open.org/repl//children"/>
8252   <cmisra:numItems>1</cmisra:numItems>
8253   <atom:entry>
8254     <atom:author>
8255       <atom:name>Al Brown</atom:name>
8256       <atom:uri>http://www.ibm.com/</atom:uri>
8257       <atom:email>albertcbrown@us.ibm.com</atom:email>
8258     </atom:author>
8259     <atom:content>Type Definition for cmis:document</atom:content>
8260     <atom:id>http://cmisexample.oasis-
8261 open.org/repl/type/cmisis/document</atom:id>
8262     <atom:link type="application/atom+xml;type=entry" rel="self"
8263 href="http://cmisexample.oasis-open.org/repl/type/cmisis/document" />
8264     <atom:link type="application/atomsvc+xml" rel="service"
8265 href="http://cmisexample.oasis-open.org/repl/type/cmisis/document" />
```

```

8266         <atom:link type="application/atom+xml;type=entry" rel="describedby"
8267 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
8268         <atom:link type="application/atom+xml;type=entry" rel="up"
8269 href="http://cmisexample.oasis-open.org/repl/type/cmis:document/parent"/>
8270         <atom:link type="application/atom+xml;type=feed" rel="down"
8271 href="http://cmisexample.oasis-
8272 open.org/repl/type/cmis:document/children/flat"/>
8273         <atom:link type="application/cmistree+xml" rel="down"
8274 href="http://cmisexample.oasis-
8275 open.org/repl/type/cmis:document/children/tree"/>
8276         <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
8277         <atom:summary type="html">HTML summary of Type Definition
8278 cmis:document</atom:summary>
8279         <atom:title type="text">Type Definition - cmis:document</atom:title>
8280         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8281         <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8282         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8283 xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="cmis:document">
8284             <cmis:id>dtcmis:document</cmis:id>
8285             <cmis:localName>myrepname-cmis:document</cmis:localName>
8286             <cmis:localNamespace xsi:nil="true"/>
8287             <cmis:displayName>cmis:document</cmis:displayName>
8288             <cmis:queryName>cmis:document</cmis:queryName>
8289             <cmis:description>Description for type definition
8290 cmis:document</cmis:description>
8291             <cmis:baseId>cmis:document</cmis:baseId>
8292             <cmis:parentId>parent</cmis:parentId>
8293             <cmis:creatable>true</cmis:creatable>
8294             <cmis:fileable>true</cmis:fileable>
8295             <cmis:queryable>false</cmis:queryable>
8296             <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8297
8298             <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8299             <cmis:controllablePolicy>true</cmis:controllablePolicy>
8300             <cmis:controllableACL>true</cmis:controllableACL>
8301             <cmis:versionable>true</cmis:versionable>
8302             <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8303         </cmisra:type>
8304         <cmisra:children>
8305             <atom:feed>
8306                 <atom:title type="text">Children for Document</atom:title>
8307                 <atom:author>
8308                     <atom:name>Al Brown</atom:name>
8309                     <atom:uri>http://www.ibm.com/</atom:uri>
8310                     <atom:email>albertcbrown@us.ibm.com</atom:email>
8311                 </atom:author>
8312                 <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8313                 <atom:id>urn:uuid:6f1cdc44-bd89-41c0-8fad-
8314 89f3ad0f8f30</atom:id>
8315                 <atom:link type="application/atom+xml;type=feed" rel="self"
8316 href="http://cmisexample.oasis-open.org/repl/cmis:document/3"/>
8317                 <atom:link type="application/atomsvc+xml" rel="service"
8318 href="http://cmisexample.oasis-open.org/repl//service"/>
8319                 <atom:link type="application/atom+xml;type=entry" rel="via"
8320 href="http://cmisexample.oasis-open.org/repl/cmis:document"/>
8321                 <atom:link type="application/atom+xml;type=feed" rel="down"
8322 href="http://cmisexample.oasis-open.org/repl/cmis:document/children"/>
8323                 <atom:link type="application/atom+xml;type=entry" rel="up"
8324 href="http://cmisexample.oasis-open.org/repl/document"/>
8325                 <cmisra:numItems>1</cmisra:numItems>
8326                 <atom:entry>
8327                     <atom:author>
8328                         <atom:name>Al Brown</atom:name>
8329                         <atom:uri>http://www.ibm.com/</atom:uri>

```

```

8330         <atom:email>albertcbrown@us.ibm.com</atom:email>
8331     </atom:author>
8332     <atom:content>Type Definition for invoice-
8333 document</atom:content>
8334     <atom:id>http://cmisexample.oasis-
8335 open.org/repl/type/invoice-document</atom:id>
8336     <atom:link type="application/atom+xml;type=entry"
8337 rel="self" href="http://cmisexample.oasis-open.org/repl/type/invoice-
8338 document"/>
8339     <atom:link type="application/atomsvc+xml" rel="service"
8340 href="http://cmisexample.oasis-open.org/repl/type/invoice-document"/>
8341     <atom:link type="application/atom+xml;type=entry"
8342 rel="describedby" href="http://cmisexample.oasis-
8343 open.org/repl/type/cmisa:document"/>
8344     <atom:link type="application/atom+xml;type=entry" rel="up"
8345 href="http://cmisexample.oasis-open.org/repl/type/invoice-document/parent"/>
8346     <atom:link type="application/atom+xml;type=feed"
8347 rel="down" href="http://cmisexample.oasis-open.org/repl/type/invoice-
8348 document/children/flat"/>
8349     <atom:link type="application/cmistree+xml" rel="down"
8350 href="http://cmisexample.oasis-open.org/repl/type/invoice-
8351 document/children/tree"/>
8352     <atom:published>2010-01-25T10:20:59.927-
8353 08:00</atom:published>
8354     <atom:summary type="html">HTML summary of Type Definition
8355 invoice-document</atom:summary>
8356     <atom:title type="text">Type Definition - invoice-
8357 document</atom:title>
8358     <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8359     <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8360     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8361 instance" xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="invoice-
8362 document">
8363         <cmis:id>dtinvoice-document</cmis:id>
8364         <cmis:localName>myrepname-invoice-
8365 document</cmis:localName>
8366         <cmis:localNamespace xsi:nil="true"/>
8367         <cmis:displayName>invoice-document</cmis:displayName>
8368         <cmis:queryName>invoice-document</cmis:queryName>
8369         <cmis:description>Description for type definition
8370 invoice-document</cmis:description>
8371         <cmis:baseId>cmis:document</cmis:baseId>
8372         <cmis:parentId>parent</cmis:parentId>
8373         <cmis:creatable>true</cmis:creatable>
8374         <cmis:fileable>true</cmis:fileable>
8375         <cmis:queryable>false</cmis:queryable>
8376         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8377
8378 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8379
8380 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8381     <cmis:controllableACL>true</cmis:controllableACL>
8382     <cmis:versionable>true</cmis:versionable>
8383
8384 <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8385     </cmisra:type>
8386 </atom:entry>
8387 </atom:feed>
8388 </cmisra:children>
8389 </atom:entry>
8390 <atom:entry>
8391     <atom:author>
8392         <atom:name>Al Brown</atom:name>
8393         <atom:uri>http://www.ibm.com/</atom:uri>

```

```

8394         <atom:email>albertcbrown@us.ibm.com</atom:email>
8395     </atom:author>
8396     <atom:content>Type Definition for cmis:folder</atom:content>
8397     <atom:id>http://cmisexample.oasis-
8398 open.org/repl/type/cmis:folder</atom:id>
8399     <atom:link type="application/atom+xml;type=entry" rel="self"
8400 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8401     <atom:link type="application/atomsvc+xml" rel="service"
8402 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8403     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8404 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder"/>
8405     <atom:link type="application/atom+xml;type=entry" rel="up"
8406 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/parent"/>
8407     <atom:link type="application/atom+xml;type=feed" rel="down"
8408 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/children/flat"/>
8409     <atom:link type="application/cmistree+xml" rel="down"
8410 href="http://cmisexample.oasis-open.org/repl/type/cmis:folder/children/tree"/>
8411     <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
8412     <atom:summary type="html">HTML summary of Type Definition
8413 cmis:folder</atom:summary>
8414     <atom:title type="text">Type Definition - cmis:folder</atom:title>
8415     <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8416     <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8417     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8418 xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="cmis:folder">
8419         <cmis:id>dtcmis:folder</cmis:id>
8420         <cmis:localName>myrepname-cmis:folder</cmis:localName>
8421         <cmis:localNamespace xsi:nil="true"/>
8422         <cmis:displayName>cmis:folder</cmis:displayName>
8423         <cmis:queryName>cmis:folder</cmis:queryName>
8424         <cmis:description>Description for type definition
8425 cmis:folder</cmis:description>
8426         <cmis:baseId>cmis:folder</cmis:baseId>
8427         <cmis:parentId>parent</cmis:parentId>
8428         <cmis:creatable>true</cmis:creatable>
8429         <cmis:fileable>true</cmis:fileable>
8430         <cmis:queryable>false</cmis:queryable>
8431         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8432
8433     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8434     <cmis:controllablePolicy>true</cmis:controllablePolicy>
8435     <cmis:controllableACL>true</cmis:controllableACL>
8436 </cmisra:type>
8437 <cmisra:children>
8438     <atom:feed>
8439         <atom:title type="text">Children for Folder</atom:title>
8440         <atom:author>
8441             <atom:name>Al Brown</atom:name>
8442             <atom:uri>http://www.ibm.com/</atom:uri>
8443             <atom:email>albertcbrown@us.ibm.com</atom:email>
8444         </atom:author>
8445         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8446         <atom:id>urn:uuid:361a3ac1-f7f7-47cb-b941-
8447 ae1200213fe0</atom:id>
8448         <atom:link type="application/atom+xml;type=feed" rel="self"
8449 href="http://cmisexample.oasis-open.org/repl/cmis:folder/3"/>
8450         <atom:link type="application/atomsvc+xml" rel="service"
8451 href="http://cmisexample.oasis-open.org/repl//service"/>
8452         <atom:link type="application/atom+xml;type=entry" rel="via"
8453 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8454         <atom:link type="application/atom+xml;type=feed" rel="down"
8455 href="http://cmisexample.oasis-open.org/repl/cmis:folder/children"/>
8456         <atom:link type="application/atom+xml;type=entry" rel="up"
8457 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>

```

```

8458         <cmisra:numItems>1</cmisra:numItems>
8459         <atom:entry>
8460             <atom:author>
8461                 <atom:name>Al Brown</atom:name>
8462                 <atom:uri>http://www.ibm.com/</atom:uri>
8463                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8464             </atom:author>
8465             <atom:content>Type Definition for customer-
8466 folder</atom:content>
8467             <atom:id>http://cmisexample.oasis-
8468 open.org/repl/type/customer-folder</atom:id>
8469             <atom:link type="application/atom+xml;type=entry"
8470 rel="self" href="http://cmisexample.oasis-open.org/repl/type/customer-
8471 folder"/>
8472             <atom:link type="application/atomsvc+xml" rel="service"
8473 href="http://cmisexample.oasis-open.org/repl/type/customer-folder"/>
8474             <atom:link type="application/atom+xml;type=entry"
8475 rel="describedby" href="http://cmisexample.oasis-
8476 open.org/repl/type/cmisis:folder"/>
8477             <atom:link type="application/atom+xml;type=entry" rel="up"
8478 href="http://cmisexample.oasis-open.org/repl/type/customer-folder/parent"/>
8479             <atom:link type="application/atom+xml;type=feed"
8480 rel="down" href="http://cmisexample.oasis-open.org/repl/type/customer-
8481 folder/children/flat"/>
8482             <atom:link type="application/cmistree+xml" rel="down"
8483 href="http://cmisexample.oasis-open.org/repl/type/customer-
8484 folder/children/tree"/>
8485             <atom:published>2010-01-25T10:20:59.927-
8486 08:00</atom:published>
8487             <atom:summary type="html">HTML summary of Type Definition
8488 customer-folder</atom:summary>
8489             <atom:title type="text">Type Definition - customer-
8490 folder</atom:title>
8491             <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8492             <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8493             <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8494 instance" xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="customer-
8495 folder">
8496                 <cmis:id>dtcustomer-folder</cmis:id>
8497                 <cmis:localName>myrepname-customer-
8498 folder</cmis:localName>
8499                 <cmis:localNamespace xsi:nil="true"/>
8500                 <cmis:displayName>customer-folder</cmis:displayName>
8501                 <cmis:queryName>customer-folder</cmis:queryName>
8502                 <cmis:description>Description for type definition
8503 customer-folder</cmis:description>
8504                 <cmis:baseId>cmis:folder</cmis:baseId>
8505                 <cmis:parentId>parent</cmis:parentId>
8506                 <cmis:creatable>true</cmis:creatable>
8507                 <cmis:fileable>true</cmis:fileable>
8508                 <cmis:queryable>false</cmis:queryable>
8509                 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8510
8511             <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8512
8513             <cmis:controllablePolicy>true</cmis:controllablePolicy>
8514                 <cmis:controllableACL>true</cmis:controllableACL>
8515             </cmisra:type>
8516         </atom:entry>
8517     </atom:feed>
8518 </cmisra:children>
8519 </atom:entry>
8520 <atom:entry>
8521     <atom:author>

```



```

8522         <atom:name>Al Brown</atom:name>
8523         <atom:uri>http://www.ibm.com/</atom:uri>
8524         <atom:email>albertcbrown@us.ibm.com</atom:email>
8525     </atom:author>
8526     <atom:content>Type Definition for cmis:relationship</atom:content>
8527     <atom:id>http://cmisexample.oasis-
8528 open.org/repl/type/cmis:relationship</atom:id>
8529     <atom:link type="application/atom+xml;type=entry" rel="self"
8530 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8531     <atom:link type="application/atomsvc+xml" rel="service"
8532 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8533     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8534 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship"/>
8535     <atom:link type="application/atom+xml;type=entry" rel="up"
8536 href="http://cmisexample.oasis-open.org/repl/type/cmis:relationship/parent"/>
8537     <atom:link type="application/atom+xml;type=feed" rel="down"
8538 href="http://cmisexample.oasis-
8539 open.org/repl/type/cmis:relationship/children/flat"/>
8540     <atom:link type="application/cmistree+xml" rel="down"
8541 href="http://cmisexample.oasis-
8542 open.org/repl/type/cmis:relationship/children/tree"/>
8543     <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
8544     <atom:summary type="html">HTML summary of Type Definition
8545 cmis:relationship</atom:summary>
8546     <atom:title type="text">Type Definition -
8547 cmis:relationship</atom:title>
8548     <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8549     <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8550     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8551 xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8552 cmisra:id="cmis:relationship">
8553         <cmis:id>dtcmis:relationship</cmis:id>
8554         <cmis:localName>myrepname-cmis:relationship</cmis:localName>
8555         <cmis:localNamespace xsi:nil="true"/>
8556         <cmis:displayName>cmis:relationship</cmis:displayName>
8557         <cmis:queryName>cmis:relationship</cmis:queryName>
8558         <cmis:description>Description for type definition
8559 cmis:relationship</cmis:description>
8560         <cmis:baseId>cmis:relationship</cmis:baseId>
8561         <cmis:parentId>parent</cmis:parentId>
8562         <cmis:creatable>true</cmis:creatable>
8563         <cmis:fileable>false</cmis:fileable>
8564         <cmis:queryable>false</cmis:queryable>
8565         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8566
8567         <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8568         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8569         <cmis:controllableACL>true</cmis:controllableACL>
8570         <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8571         <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8572         <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8573     </cmisra:type>
8574     <cmisra:children>
8575         <atom:feed>
8576             <atom:title type="text">Children for Relationship</atom:title>
8577             <atom:author>
8578                 <atom:name>Al Brown</atom:name>
8579                 <atom:uri>http://www.ibm.com/</atom:uri>
8580                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8581             </atom:author>
8582             <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8583             <atom:id>urn:uuid:9394ff3d-87c4-48c9-a951-
8584 ba725560faac</atom:id>

```

```

8585         <atom:link type="application/atom+xml;type=feed" rel="self"
8586 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/3"/>
8587         <atom:link type="application/atomsvc+xml" rel="service"
8588 href="http://cmisexample.oasis-open.org/repl//service"/>
8589         <atom:link type="application/atom+xml;type=entry" rel="via"
8590 href="http://cmisexample.oasis-open.org/repl/cmis:relationship"/>
8591         <atom:link type="application/atom+xml;type=feed" rel="down"
8592 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/children"/>
8593         <atom:link type="application/atom+xml;type=entry" rel="up"
8594 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8595         <cmisra:numItems>1</cmisra:numItems>
8596         <atom:entry>
8597             <atom:author>
8598                 <atom:name>Al Brown</atom:name>
8599                 <atom:uri>http://www.ibm.com/</atom:uri>
8600                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8601             </atom:author>
8602             <atom:content>Type Definition for customer-
8603 relationship</atom:content>
8604             <atom:id>http://cmisexample.oasis-
8605 open.org/repl/type/customer-relationship</atom:id>
8606             <atom:link type="application/atom+xml;type=entry"
8607 rel="self" href="http://cmisexample.oasis-open.org/repl/type/customer-
8608 relationship"/>
8609             <atom:link type="application/atomsvc+xml" rel="service"
8610 href="http://cmisexample.oasis-open.org/repl/type/customer-relationship"/>
8611             <atom:link type="application/atom+xml;type=entry"
8612 rel="describedby" href="http://cmisexample.oasis-
8613 open.org/repl/type/cmis:relationship"/>
8614             <atom:link type="application/atom+xml;type=entry" rel="up"
8615 href="http://cmisexample.oasis-open.org/repl/type/customer-
8616 relationship/parent"/>
8617             <atom:link type="application/atom+xml;type=feed"
8618 rel="down" href="http://cmisexample.oasis-open.org/repl/type/customer-
8619 relationship/children/flat"/>
8620             <atom:link type="application/cmistree+xml" rel="down"
8621 href="http://cmisexample.oasis-open.org/repl/type/customer-
8622 relationship/children/tree"/>
8623             <atom:published>2010-01-25T10:20:59.943-
8624 08:00</atom:published>
8625             <atom:summary type="html">HTML summary of Type Definition
8626 customer-relationship</atom:summary>
8627             <atom:title type="text">Type Definition - customer-
8628 relationship</atom:title>
8629             <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8630             <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8631             <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8632 instance" xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8633 cmisra:id="customer-relationship">
8634                 <cmis:id>dtcustomer-relationship</cmis:id>
8635                 <cmis:localName>myrepname-customer-
8636 relationship</cmis:localName>
8637                 <cmis:localNamespace xsi:nil="true"/>
8638                 <cmis:displayName>customer-
8639 relationship</cmis:displayName>
8640                 <cmis:queryName>customer-relationship</cmis:queryName>
8641                 <cmis:description>Description for type definition
8642 customer-relationship</cmis:description>
8643                 <cmis:baseId>cmis:relationship</cmis:baseId>
8644                 <cmis:parentId>parent</cmis:parentId>
8645                 <cmis:creatable>true</cmis:creatable>
8646                 <cmis:fileable>false</cmis:fileable>
8647                 <cmis:queryable>false</cmis:queryable>
8648                 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>

```

```

8649
8650 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8651
8652 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8653 <cmis:controllableACL>true</cmis:controllableACL>
8654
8655 <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8656
8657 <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8658
8659 <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8660 </cmisra:type>
8661 </atom:entry>
8662 </atom:feed>
8663 </cmisra:children>
8664 </atom:entry>
8665 <atom:entry>
8666 <atom:author>
8667 <atom:name>Al Brown</atom:name>
8668 <atom:uri>http://www.ibm.com/</atom:uri>
8669 <atom:email>albertcbrown@us.ibm.com</atom:email>
8670 </atom:author>
8671 <atom:content>Type Definition for cmis:policy</atom:content>
8672 <atom:id>http://cmisexample.oasis-
8673 open.org/repl/type/cmis:policy</atom:id>
8674 <atom:link type="application/atom+xml;type=entry" rel="self"
8675 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8676 <atom:link type="application/atomsvc+xml" rel="service"
8677 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8678 <atom:link type="application/atom+xml;type=entry" rel="describedby"
8679 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8680 <atom:link type="application/atom+xml;type=entry" rel="up"
8681 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/parent"/>
8682 <atom:link type="application/atom+xml;type=feed" rel="down"
8683 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/flat"/>
8684 <atom:link type="application/cmistree+xml" rel="down"
8685 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/tree"/>
8686 <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
8687 <atom:summary type="html">HTML summary of Type Definition
8688 cmis:policy</atom:summary>
8689 <atom:title type="text">Type Definition - cmis:policy</atom:title>
8690 <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8691 <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8692 <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8693 xsi:type="cmis:cmisTypePolicyDefinitionType" cmisra:id="cmis:policy">
8694 <cmis:id>dtcmis:policy</cmis:id>
8695 <cmis:localName>myrepname-cmis:policy</cmis:localName>
8696 <cmis:localNamespace xsi:nil="true"/>
8697 <cmis:displayName>cmis:policy</cmis:displayName>
8698 <cmis:queryName>cmis:policy</cmis:queryName>
8699 <cmis:description>Description for type definition
8700 cmis:policy</cmis:description>
8701 <cmis:baseId>cmis:policy</cmis:baseId>
8702 <cmis:parentId>parent</cmis:parentId>
8703 <cmis:creatable>true</cmis:creatable>
8704 <cmis:fileable>false</cmis:fileable>
8705 <cmis:queryable>false</cmis:queryable>
8706 <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8707
8708 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8709 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8710 <cmis:controllableACL>true</cmis:controllableACL>
8711 </cmisra:type>
8712 </atom:entry>

```

8713 `</atom:feed>`

8714

8715 Please also see the example documents included with the schema.

8716 **3.9.6.1 GET**

8717 The following arguments may be supplied. Please see the domain model for more information:

- 8718 • includePropertyDefinitions
- 8719 • depth

8720 **3.10 Resources**

8721 For any HTTP verb not specified on a resource, each implementation MAY choose to implement that HTTP
8722 verb in a repository-specific manner.

8723 **3.10.1 Type Entry**

8724 This represents a type definition in the repository. This is enclosed as an atom entry

8725

8726 CMIS Services:

8727 GET: getTypeDefinition

8728 Media Type: application/atom+xml;type=entry

8729

8730 Link Relations:

- 8731 • service: Points to service document containing the CMIS repository. The service document
8732 MUST contain only one workspace element.
 - 8733 ○ Media Type: application/atomsvc+xml
- 8734 • up: Points to the parent type as atom entry if applicable
- 8735 • down: Points to the children of this type as atom feed if applicable
 - 8736 ○ (Children) Media Type: application/atom+xml;type=feed points to the atom feed
8737 document representing the children feed for this same type
 - 8738 ○ (Descendants) Media Type: application/cmistree+xml points to the atom feed document
8739 representing the descendants feed for this same type
- 8740 • describedby: Points to the type definition atom entry of the base type of this type definition.

8741

8742 The following CMIS Atom extension element MUST be included inside the atom entry:

- 8743 • cmisra:type

8744 **3.10.1.1 GET**

8745 There are no optional arguments for this resource.

8746

8747 Request:

8748 `GET /obj/5070f89a-6f00-4acf-84e9-d8836a6c7d23 HTTP/1.1`
8749 `Host: example.org`

8750

8751

8752 Response:

8753 `HTTP/1.1 200 Ok`

```

Date: Mon, 25 Jan 2010 10:21:00 -0800
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/atomsvc+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"/>
  <atom:link type="application/cmistree+xml" rel="down"
href="http://cmisexample.oasis-
open.org/repl/type/cmis:document/children/tree"/>
  <atom:published>2010-01-25T10:21:00.380-08:00</atom:published>
  <atom:summary type="html">HTML summary of Type Definition
cmis:document</atom:summary>
  <atom:title type="text">Type Definition - cmis:document</atom:title>
  <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
  <app:edited>2010-01-25T10:21:00.380-08:00</app:edited>
  <cmisra:type xsi:type="cmis:cmisTypeDocumentDefinitionType"
cmisra:id="cmis:document" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance">
    <cmis:id>dtcmis:document</cmis:id>
    <cmis:localName>myrepname-cmis:document</cmis:localName>
    <cmis:localNamespace xsi:nil="true"/>
    <cmis:displayName>cmis:document</cmis:displayName>
    <cmis:queryName>cmis:document</cmis:queryName>
    <cmis:description>Description for type definition
cmis:document</cmis:description>
    <cmis:baseId>cmis:document</cmis:baseId>
    <cmis:parentId>parent</cmis:parentId>
    <cmis:creatable>true</cmis:creatable>
    <cmis:fileable>true</cmis:fileable>
    <cmis:queryable>false</cmis:queryable>
    <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
    <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
    <cmis:controllablePolicy>true</cmis:controllablePolicy>
    <cmis:controllableACL>true</cmis:controllableACL>
    <cmis:versionable>true</cmis:versionable>
    <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
  </cmisra:type>
</atom:entry>

```

8818
8819
8820

Please also see the example documents included with the schema.

8821 3.10.2 Document Entry

8822 This is a CMIS Document instance.

8823

8824 CMIS Services:

8825 GET: getObject, getObjectOfLatestVersion (getObject)

8826 PUT: updateProperties

8827 DELETE: deleteObject

8828 Media Type: application/atom+xml;type=entry

8829

8830 Link Relations:

- 8831 • self: Points to an URI that returns the atom entry for this document. Please see Atom for more
8832 information.
- 8833 • edit: Points to an URI that accepts PUT of atom entry. Please see AtomPub for more information.
- 8834 • service: Points to service document containing the CMIS repository. The service document
8835 MUST contain only one workspace element.
 - 8836 ○ Media Type: application/atomsvc+xml
- 8837 • up: Points to the atom feed containing the set of parents. If there is only one parent, the
8838 repository MAY point this link relation directly to the atom entry of the parent.
- 8839 • version-history: Points to atom feed containing the versions of this document
 - 8840 ○ If the document is not versionable, this link relation may not be on the resource
- 8841 • current-version: Points to the latest version of the document
 - 8842 ○ Uses query parameter 'returnVersion' and enumReturnVersion
 - 8843 ○ If this version is the current-version, this link relation may not be on the resource
- 8844 • edit-media:
 - 8845 ○ Same as setContentStream. Allows updating the content stream on this document
 - 8846 ○ Please see AtomPub for more information
- 8847 • working-copy: Points to the private working copy if it exists.
- 8848 • describedby: Points to the type definition as an atom entry for the type of this document entry.
- 8849 • alternate: this is used to identify the renditions available for the specified object. Please see the
8850 Renditions section.
- 8851 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions
8852 document for this object.
- 8853 • <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for
8854 this object
- 8855 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 8856 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

8857

8858 The following CMIS Atom extension element MUST be included inside the atom entry:

- 8859 • cmisra:object

8860

8861 3.10.2.1 GET

8862 The following arguments may be supplied. Please see the domain model for more information:

- 8863
- 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.
- 8874
- 8875

8876 Request:

```
8877 GET /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId HTTP/1.1
8878 Host: example.org
8879
```

8880

8881 Response:

```
8882 HTTP/1.1 200 Ok
8883 Date: Mon, 25 Jan 2010 10:21:00 -0800
8884 Content-Length: 3403
8885 Content-Type: application/atom+xml;type=entry
8886 Location: /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId
8887
8888
8889 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8890 <atom:entry xmlns:app="http://www.w3.org/2007/app"
8891 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
8892 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
8893 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
8894 open.org/ns/cmis/restatom/200908/">
8895   <atom:author>
8896     <atom:name>Al Brown</atom:name>
8897     <atom:uri>http://www.ibm.com/</atom:uri>
8898     <atom:email>albertcbrown@us.ibm.com</atom:email>
8899   </atom:author>
8900   <atom:content src="http://cmisexample.oasis-open.org/repl/7c088887-5991-
8901 4b3a-9ad3-16379127e647"/>
8902   <atom:id>urn:uuid:7c088887-5991-4b3a-9ad3-16379127e647</atom:id>
8903   <atom:title type="text">Invoice</atom:title>
8904   <atom:updated>2010-01-25T10:21:00.193-08:00</atom:updated>
8905   <atom:link rel="self" href="http://cmisexample.oasis-
8906 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647"/>
8907   <atom:link rel="edit" href="http://cmisexample.oasis-
8908 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647"/>

```

```

8909     <atom:link type="application/cmism+xml;type=allowableActions"
8910     rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
8911     href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
8912     16379127e647/allowableactions"/>
8913     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8914     href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
8915     16379127e647/type"/>
8916     <atom:link type="application/atomsvc+xml" rel="service"
8917     href="http://cmisexample.oasis-open.org/repl//service"/>
8918     <atom:published>2010-01-25T10:21:00.193-08:00</atom:published>
8919     <atom:summary type="html">HTML summary of Entry 7c088887-5991-4b3a-9ad3-
8920     16379127e647</atom:summary>
8921     <atom:link rel="edit-media" href="http://cmisexample.oasis-
8922     open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/edit-media"/>
8923     <atom:link rel="alternate" href="http://cmisexample.oasis-
8924     open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/alternate"/>
8925     <atom:link type="application/atom+xml;type=feed" rel="up"
8926     href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
8927     16379127e647/parents"/>
8928     <atom:link type="application/atom+xml;type=feed" rel="version-history"
8929     href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
8930     16379127e647/allversions"/>
8931     <atom:link type="application/atom+xml;type=entry" rel="current-version"
8932     href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
8933     16379127e647/latest"/>
8934     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
8935     open.org/ns/cmism/link/200908/relationships" href="http://cmisexample.oasis-
8936     open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/relationships"/>
8937     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
8938     open.org/ns/cmism/link/200908/policies" href="http://cmisexample.oasis-
8939     open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/policies"/>
8940     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8941     open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
8942     open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/acl"/>
8943     <cmisra:object>
8944         <cmis:properties>
8945             <cmis:propertyId localName="rep-cmis:objectId"
8946             propertyDefinitionId="cmis:objectId">
8947                 <cmis:value>7c088887-5991-4b3a-9ad3-16379127e647</cmis:value>
8948             </cmis:propertyId>
8949         </cmis:properties>
8950     </cmisra:object>
8951 </atom:entry>
8952

```

Please also see the example documents included with the schema.

3.10.2.2 PUT

This does a replacement of the atom entry with the atom entry document specified. If readwrite properties are not included, the repository SHOULD NOT modify them.

The server SHOULD respond with:

- HTTP Status Code 200
- Response Body containing the updated atom entry

8964 3.10.2.3 DELETE

8965 This removes the document.

8966 Success HTTP code: 204

8967 3.10.3 Document Private Working Copy (PWC) Entry

8968 This is the private working copy of the document (checkedout version of document)

8969 CMIS Services:

8970 GET: getObject

8971 PUT: updateProperties or checkIn

8972 DELETE: cancelCheckOut

8973 Media Type: application/atom+xml;type=entry

8974

8975 Link relations:

- 8976 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 8977 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more
- 8978 information.
- 8979 • service: Points to service document containing the CMIS repository. The service document
- 8980 MUST contain only one workspace element.
 - 8981 ○ Media Type: application/atomsvc+xml
- 8982 • up: Points to the atom feed containing the set of parents. If there is only one parent, the
- 8983 repository MAY point this link relation directly to the atom entry of the parent.
- 8984 • version-history
 - 8985 ○ Points to an URI that returns the feed associated with the version history
- 8986 • edit-media
 - 8987 ○ Same as setContentStream. Allows updating the content stream on this document
 - 8988 ○ Please see AtomPub for more information
- 8989 • via: atom entry that created this private working copy
- 8990 • describedby: Points to the type definition as an atom entry for the type of this PWC entry.
- 8991 • alternate: this is used to identify the renditions available for the specified object. Please see the
- 8992 Renditions section.
- 8993 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions
- 8994 document for this object.
- 8995 • <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for
- 8996 this object
- 8997 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 8998 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

8999

9000 The following element MUST be included inside the atom entry:

- 9001 • cmisra:object

9002

9003 3.10.3.1 GET

9004 The following arguments may be supplied. Please see the domain model for more information:

- 9005 • filter
- 9006 • includeAllowableActions
- 9007 • includeRelationships
- 9008 • renditionFilter
 - 9009 ○ If not specified, renditions will not be included.

9010

9011 Request:

```
9012 GET /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId HTTP/1.1
9013 Host: example.org
9014
```

9015

9016 Response:

```
9017 HTTP/1.1 200 Ok
9018 Date: Mon, 25 Jan 2010 10:21:00 -0800
9019 Content-Length: 3564
9020 Content-Type: application/atom+xml;type=entry
9021 Location: /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId
9022
9023
9024 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9025 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9026 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9027 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9028 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9029 open.org/ns/cmis/restatom/200908/">
9030   <atom:author>
9031     <atom:name>Al Brown</atom:name>
9032     <atom:uri>http://www.ibm.com/</atom:uri>
9033     <atom:email>albertcbrown@us.ibm.com</atom:email>
9034   </atom:author>
9035   <atom:content src="http://cmisexample.oasis-open.org/repl/3240a476-6de6-
9036 4ab2-978d-85ca2f4f3206"/>
9037   <atom:id>urn:uuid:3240a476-6de6-4ab2-978d-85ca2f4f3206</atom:id>
9038   <atom:title type="text">Invoice</atom:title>
9039   <atom:updated>2010-01-25T10:21:00.333-08:00</atom:updated>
9040   <atom:link rel="self" href="http://cmisexample.oasis-
9041 open.org/repl/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>
9042   <atom:link rel="edit" href="http://cmisexample.oasis-
9043 open.org/repl/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>
9044   <atom:link type="application/cmis+xml;type=allowableActions"
9045 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9046 href="http://cmisexample.oasis-open.org/repl/3240a476-6de6-4ab2-978d-
9047 85ca2f4f3206/allowableactions"/>
9048   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9049 href="http://cmisexample.oasis-open.org/repl/3240a476-6de6-4ab2-978d-
9050 85ca2f4f3206/type"/>
9051   <atom:link type="application/atomsvc+xml" rel="service"
9052 href="http://cmisexample.oasis-open.org/repl//service"/>
9053   <atom:published>2010-01-25T10:21:00.333-08:00</atom:published>
9054   <atom:summary type="html">HTML summary of Entry 3240a476-6de6-4ab2-978d-
9055 85ca2f4f3206</atom:summary>
9056   <atom:link rel="edit-media" href="http://cmisexample.oasis-
9057 open.org/repl/3240a476-6de6-4ab2-978d-85ca2f4f3206/edit-media"/>
9058   <atom:link rel="alternate" href="http://cmisexample.oasis-
9059 open.org/repl/3240a476-6de6-4ab2-978d-85ca2f4f3206/alternate"/>
9060   <atom:link type="application/atom+xml;type=feed" rel="up"
9061 href="http://cmisexample.oasis-open.org/repl/3240a476-6de6-4ab2-978d-
9062 85ca2f4f3206/parents"/>
```

```

9063     <atom:link type="application/atom+xml;type=feed" rel="version-history"
9064 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9065 85ca2f4f3206/allversions"/>
9066     <atom:link type="application/atom+xml;type=entry" rel="current-version"
9067 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9068 85ca2f4f3206/latest"/>
9069     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9070 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9071 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/relationships"/>
9072     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9073 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9074 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/policies"/>
9075     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9076 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9077 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/acl"/>
9078     <atom:link type="application/atom+xml;type=feed" rel="working-copy"
9079 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-
9080 85ca2f4f3206/pwc"/>
9081     <cmisra:object>
9082       <cmis:properties>
9083         <cmis:propertyId localName="rep-cmis:objectId"
9084 propertyDefinitionId="cmis:objectId">
9085           <cmis:value>3240a476-6de6-4ab2-978d-85ca2f4f3206</cmis:value>
9086         </cmis:propertyId>
9087       </cmis:properties>
9088     </cmisra:object>
9089   </atom:entry>
9090

```

Please also see the example documents included with the schema.

3.10.3.2 PUT

This does a replacement of the atom entry with the atom entry document specified. If modifiable properties (whencheckedout or readwrite) are not included, the repository SHOULD NOT modify them.

The following arguments may be supplied. Please see the domain model for more information:

- checkinComment
- major
- checkin
 - Used to differentiate between updateProperties() or checkin() services. If TRUE, execute checkin service.
 - If this argument is specified as TRUE, then the body to PUT MAY be omitted if there are no modifications to be made during checkin

The server SHOULD respond with:

- HTTP Status Code 200
- Location header of the resource (if changed via checkin)
- Response Body containing the updated atom entry

3.10.3.3 DELETE

This removes the document entry, in this case, cancels the check out. The PWC will be removed.

9114 Success HTTP code: 204

9115 3.10.4 Folder Entry

9116 This is a CMIS Folder instance. The properties of a folder map onto the feed tag.

9117 CMIS Services:

9118 GET: getObject

9119 PUT: updateProperties

9120 DELETE: deleteObject (this is deletion of the folder only and not any contained objects)

9121 Media Type: application/atom+xml;type=entry

9122

9123 Link Relations:

- 9124 • self: Points to the URI to retrieve this atom entry. Please see Atom for more informationedit:
9125 Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
- 9126 • service: Points to service document containing the CMIS repository. The service document
9127 MUST contain only one workspace element.
 - 9128 ○ Media Type: application/atomsvc+xml
- 9129 • describedby: Points to the type definition as an atom entry for the type of this folder entry.
- 9130 • down: Points to the children of this folder
 - 9131 ○ application/atom+xml : Points to the atom feed document representing the children feed
9132 for this same folder
 - 9133 ○ application/cmistree+xml: Points to the descendants feed of the same folder
- 9134 • up: Points to the atom entry for the parent
 - 9135 ○ If the root folder, this link will not be present
- 9136 • alternate: this is used to identify the renditions available for the specified object. Please see the
9137 Renditions section.
- 9138 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions
9139 document for this object.
- 9140 • <http://docs.oasis-open.org/ns/cmis/link/200908/relationships>: Points to the relationships feed for
9141 this object
- 9142 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 9143 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object
- 9144 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder

9145

9146 The following CMIS Atom extension element MUST be included inside the atom entry:

- 9147 • cmisra:object

9148

9149 3.10.4.1 GET

9150 The following arguments may be supplied. Please see the domain model for more information:

- 9151 • filter
- 9152 • includeAllowableActions
- 9153 • includeRelationships
- 9154 • renditionFilter
- 9155 ○ If not specified, renditions will not be included.

9156

9157 Request:

```
9158 GET /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId HTTP/1.1
9159 Host: example.org
9160
```

9161

9162 Response:

```
9163 HTTP/1.1 200 Ok
9164 Date: Mon, 25 Jan 2010 10:21:00 -0800
9165 Content-Length: 3332
9166 Content-Type: application/atom+xml;type=entry
9167 Location: /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId
9168
9169
9170 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9171 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9172 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9173 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9174 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9175 open.org/ns/cmis/restatom/200908/">
9176   <atom:author>
9177     <atom:name>Al Brown</atom:name>
9178     <atom:uri>http://www.ibm.com/</atom:uri>
9179     <atom:email>albertcbrown@us.ibm.com</atom:email>
9180   </atom:author>
9181   <atom:content src="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-
9182 471d-b4ba-6d8756cd5093"/>
9183   <atom:id>urn:uuid:cfc03a28-8240-471d-b4ba-6d8756cd5093</atom:id>
9184   <atom:title type="text">Customer Folder</atom:title>
9185   <atom:updated>2010-01-25T10:21:00.208-08:00</atom:updated>
9186   <atom:link rel="self" href="http://cmisexample.oasis-
9187 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
9188   <atom:link rel="edit" href="http://cmisexample.oasis-
9189 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
9190   <atom:link type="application/cmis+xml;type=allowableActions"
9191 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9192 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9193 6d8756cd5093/allowableactions"/>
9194   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9195 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9196 6d8756cd5093/type"/>
9197   <atom:link type="application/atomsvc+xml" rel="service"
9198 href="http://cmisexample.oasis-open.org/repl//service"/>
9199   <atom:published>2010-01-25T10:21:00.208-08:00</atom:published>
9200   <atom:summary type="html">HTML summary of Entry cfc03a28-8240-471d-b4ba-
9201 6d8756cd5093</atom:summary>
9202   <atom:link type="application/atom+xml;type=entry" rel="up"
9203 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9204 6d8756cd5093/up"/>
9205   <atom:link type="application/atom+xml;type=feed" rel="down"
9206 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9207 6d8756cd5093/children"/>
9208   <atom:link type="application/cmistree+xml" rel="down"
9209 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9210 6d8756cd5093/tree"/>
9211   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9212 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
9213 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093/foldertree"/>
9214   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9215 open.org/ns/cmis/link/200908/relationships" href="http://cmisexample.oasis-
9216 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093/relationships"/>
```

```

9217     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9218 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9219 open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093/policies"/>
9220     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9221 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9222 open.org/rep1/cfc03a28-8240-471d-b4ba-6d8756cd5093/acl"/>
9223     <cmisra:object>
9224       <cmis:properties>
9225         <cmis:propertyId localName="rep-cmis:objectId"
9226 propertyDefinitionId="cmis:objectId">
9227           <cmis:value>cfc03a28-8240-471d-b4ba-6d8756cd5093</cmis:value>
9228         </cmis:propertyId>
9229       </cmis:properties>
9230     </cmisra:object>
9231   </atom:entry>
9232

```

Please also see the example documents included with the schema.

3.10.4.2 PUT

This does a replacement of the atom entry with the atom entry document specified. If readwrite properties are not included, the repository SHOULD NOT modify them.

The server SHOULD respond with:

- HTTP Status Code 200
- Response Body containing the updated atom entry

3.10.4.3 DELETE

This removes the object (folder) from the repository.

Success HTTP code: 204

3.10.5 Relationship Entry

This is a CMIS relationship instance. These objects are exposed via 'relationships' link type.

CMIS Services:

GET: getObject

PUT: updateProperties

DELETE: deleteObject

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

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/atomsvc+xml
- describedby: Points to the type definition as an atom entry for the type of this relationship entry.

- 9263 • <http://docs.oasis-open.org/ns/cmis/link/200908/target>
- 9264 • <http://docs.oasis-open.org/ns/cmis/link/200908/source>
- 9265 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions
- 9266 document for this object.
- 9267 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.
- 9268 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

9269

9270 The following element MUST be included inside the atom entry:

- 9271 • `cmisra:object`

9272

9273 3.10.5.1 GET

9274 The following arguments may be supplied. Please see the domain model for more information:

- 9275 • `filter`
- 9276 • `includeAllowableActions`

9277

9278 Request:

```
9279 GET /obj/ad443afd-aala-4071-9735-1a49aac4e439?filter=cmis:objectId HTTP/1.1
9280 Host: example.org
9281
```

9282

9283 Response:

```
9284 HTTP/1.1 200 Ok
9285 Date: Mon, 25 Jan 2010 10:21:00 -0800
9286 Content-Length: 2861
9287 Content-Type: application/atom+xml;type=entry
9288 Location: /obj/ad443afd-aala-4071-9735-1a49aac4e439?filter=cmis:objectId
9289
9290
9291 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9292 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9293 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9294 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9295 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9296 open.org/ns/cmis/restatom/200908/">
9297   <atom:author>
9298     <atom:name>Al Brown</atom:name>
9299     <atom:uri>http://www.ibm.com/</atom:uri>
9300     <atom:email>albertcbrown@us.ibm.com</atom:email>
9301   </atom:author>
9302   <atom:content src="http://cmisexample.oasis-open.org/repl/ad443afd-aala-
9303 4071-9735-1a49aac4e439"/>
9304   <atom:id>urn:uuid:ad443afd-aala-4071-9735-1a49aac4e439</atom:id>
9305   <atom:title type="text">Customer Relationship</atom:title>
9306   <atom:updated>2010-01-25T10:21:00.349-08:00</atom:updated>
9307   <atom:link rel="self" href="http://cmisexample.oasis-
9308 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439"/>
9309   <atom:link rel="edit" href="http://cmisexample.oasis-
9310 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439"/>
9311   <atom:link type="application/cmis+xml;type=allowableActions"
9312 rel="http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions"
9313 href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9314 1a49aac4e439/allowableactions"/>
```

```

9315     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9316 href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9317 1a49aac4e439/type"/>
9318     <atom:link type="application/atomsvc+xml" rel="service"
9319 href="http://cmisexample.oasis-open.org/repl//service"/>
9320     <atom:published>2010-01-25T10:21:00.365-08:00</atom:published>
9321     <atom:summary type="html">HTML summary of Entry ad443afd-aala-4071-9735-
9322 1a49aac4e439</atom:summary>
9323     <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9324 open.org/ns/cmis/link/200908/source" href="http://cmisexample.oasis-
9325 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/source"/>
9326     <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9327 open.org/ns/cmis/link/200908/target" href="http://cmisexample.oasis-
9328 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/target"/>
9329     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9330 open.org/ns/cmis/link/200908/policies" href="http://cmisexample.oasis-
9331 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/policies"/>
9332     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9333 open.org/ns/cmis/link/200908/acl" href="http://cmisexample.oasis-
9334 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/acl"/>
9335     <cmisra:object>
9336       <cmis:properties>
9337         <cmis:propertyId localName="rep-cmis:objectId"
9338 propertyDefinitionId="cmis:objectId">
9339           <cmis:value>ad443afd-aala-4071-9735-1a49aac4e439</cmis:value>
9340         </cmis:propertyId>
9341       </cmis:properties>
9342     </cmisra:object>
9343 </atom:entry>
9344

```

Please also see the example documents included with the schema.

3.10.5.2 PUT

This does a replacement of the atom entry with the atom entry document specified. If readwrite properties are not included, the repository SHOULD NOT modify them.

The server SHOULD respond with:

- HTTP Status Code 200
- Response Body containing the updated atom entry

3.10.5.3 DELETE

This removes the relationship entry.

Successful HTTP code: 204

3.10.6 Policy Entry

This is a CMIS policy instance.

CMIS Services:

- GET: getObject
- PUT: updateProperties
- DELETE: deleteObject or removePolicy

9365 Media Type: application/atom+xml;type=entry

9366

9367 Link Relations:

9368 • self

9369 • edit

9370 • service: Points to service document containing the CMIS repository. The service document
9371 MUST contain only one workspace element.

9372 ◦ Media Type: application/atomsvc+xml

9373 • describedby: Points to the type definition as an atom entry for the type of this policy entry.

9374 • alternate: this is used to identify the renditions available for the specified object. Please see the
9375 Renditions section.

9376 • <http://docs.oasis-open.org/ns/cmis/link/200908/allowableactions>: Points to the allowable actions
9377 document for this object.

9378 • <http://docs.oasis-open.org/ns/cmis/link/200908/policies>: Points to the policy feed for this object.

9379 • <http://docs.oasis-open.org/ns/cmis/link/200908/acl>: Points to ACL document for this object

9380

9381 The following element MUST be included inside the atom entry:

9382 • `cmisra:object`

9383

9384 3.10.6.1 GET

9385 The following arguments may be supplied. Please see the domain model for more information:

9386 • filter

9387 • includeAllowableActions

9388 • includeRelationships

9389 • renditionFilter

9390 ◦ If not specified, renditions will not be included.

9391

9392 Request:

9393 GET /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId HTTP/1.1
9394 Host: example.org
9395

9396

9397 Response:

9398 HTTP/1.1 200 Ok
9399 Date: Mon, 25 Jan 2010 10:21:00 -0800
9400 Content-Length: 2608
9401 Content-Type: application/atom+xml;type=entry
9402 Location: /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId
9403
9404
9405 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9406 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9407 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9408 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9409 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9410 open.org/ns/cmis/restatom/200908/">
9411 <atom:author>
9412 <atom:name>Al Brown</atom:name>

```

9413     <atom:uri>http://www.ibm.com/</atom:uri>
9414     <atom:email>albertcbrown@us.ibm.com</atom:email>
9415     </atom:author>
9416     <atom:content src="http://cmisexample.oasis-open.org/repl/a09ed524-5f1b-
9417 4940-b2f0-4e4cd4631bf0"/>
9418     <atom:id>urn:uuid:a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</atom:id>
9419     <atom:title type="text">Security Policy</atom:title>
9420     <atom:updated>2010-01-25T10:21:00.318-08:00</atom:updated>
9421     <atom:link rel="self" href="http://cmisexample.oasis-
9422 open.org/repl/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
9423     <atom:link rel="edit" href="http://cmisexample.oasis-
9424 open.org/repl/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
9425     <atom:link type="application/cmisacl+xml;type=allowableActions"
9426 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions"
9427 href="http://cmisexample.oasis-open.org/repl/a09ed524-5f1b-4940-b2f0-
9428 4e4cd4631bf0/allowableactions"/>
9429     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9430 href="http://cmisexample.oasis-open.org/repl/a09ed524-5f1b-4940-b2f0-
9431 4e4cd4631bf0/type"/>
9432     <atom:link type="application/atomsvc+xml" rel="service"
9433 href="http://cmisexample.oasis-open.org/repl//service"/>
9434     <atom:published>2010-01-25T10:21:00.318-08:00</atom:published>
9435     <atom:summary type="html">HTML summary of Entry a09ed524-5f1b-4940-b2f0-
9436 4e4cd4631bf0</atom:summary>
9437     <atom:link type="application/atom+xml;type=feed" rel="up"
9438 href="http://cmisexample.oasis-open.org/repl/a09ed524-5f1b-4940-b2f0-
9439 4e4cd4631bf0/parents"/>
9440     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9441 open.org/ns/cmisacl/link/200908/relationships" href="http://cmisexample.oasis-
9442 open.org/repl/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0/relationships"/>
9443     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9444 open.org/ns/cmisacl/link/200908/acl" href="http://cmisexample.oasis-
9445 open.org/repl/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0/acl"/>
9446     <cmisra:object>
9447         <cmis:properties>
9448             <cmis:propertyId localName="rep-cmis:objectId"
9449 propertyDefinitionId="cmis:objectId">
9450                 <cmis:value>a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</cmis:value>
9451             </cmis:propertyId>
9452         </cmis:properties>
9453     </cmisra:object>
9454 </atom:entry>
9455

```

Please also see the example documents included with the schema.

3.10.6.2 PUT

This does a replacement of the atom entry with the atom entry document specified. If read/write properties are not included, the repository SHOULD NOT modify them.

The server SHOULD respond with:

- HTTP Status Code 200
- Response Body containing the updated atom entry

9467 **3.10.6.3 DELETE**

9468 This removes the policy entry. If this policy entry was discovered through a policy collection on an object,
9469 then removePolicy() is performed rather than deleteObject() on the policy itself.

9470

9471 Success HTTP code: 204

9472 **3.10.7 Content Stream**

9473 This is the content stream portion of the document object.

9474 CMIS Services:

9475 GET: getContentStream

9476 PUT: setContentStream

9477 DELETE: deleteContentStream

9478 Media Type: Mime/Type of resource (mime type of content stream on document)

9479 **3.10.7.1 GET**

9480 This returns the content stream.

9481

9482 It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED
9483 that HTTP compression is also supported.

9484

9485 Please see RFC2616 for more information on HTTP Range requests.

9486 **3.10.7.2 PUT**

9487 This does a replacement of the content stream.

9488

9489 The following optional arguments may be supplied. Please see the domain model for more information:

9490

- overwriteFlag.

9491

- If not specified, this defaults to 'true' in this binding and behaves consistent with

9492 AtomPub.

9493

9494 Success HTTP code: 200 (with content), 204 (without content) or 201 if a new resource is created.

9495 Please see the HTTP specification for more information.

9496

9497 Returns headers:

9498

- Content-Location: URI for content stream

9499

- Location: URI for content stream

9500 **3.10.7.3 DELETE**

9501 This removes the content stream.

9502 **3.10.8 ACL Resource**

9503 CMIS Services:

9504 GET: getACL

9505 PUT: applyACL

9506
9507 Media Type: application/cmisacl+xml
9508

9509 3.10.8.1 GET

9510 This returns the CMIS ACL for a specified object. The client will follow the link on the atom entry to get
9511 the CMIS ACL for that object.

9512
9513 Request:

```
9514 GET /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef HTTP/1.1  
9515 Host: example.org  
9516
```

9517
9518 Response:

```
9519 HTTP/1.1 200 Ok  
9520 Date: Mon, 25 Jan 2010 10:21:00 -0800  
9521 Content-Length: 758  
9522 Content-Type: application/cmisacl+xml  
9523 Location: /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef  
9524  
9525  
9526 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
9527 <cmis:acl xmlns:app="http://www.w3.org/2007/app"  
9528 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
9529 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-  
9530 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-  
9531 open.org/ns/cmis/restatom/200908/">  
9532   <cmis:permission>  
9533     <cmis:principal>  
9534       <cmis:principalId>Al Brown</cmis:principalId>  
9535     </cmis:principal>  
9536     <cmis:permission>cmis:read</cmis:permission>  
9537     <cmis:permission>cmis:write</cmis:permission>  
9538     <cmis:permission>cmis:all</cmis:permission>  
9539     <cmis:permission>publish</cmis:permission>  
9540     <cmis:direct>true</cmis:direct>  
9541   </cmis:permission>  
9542 </cmis:acl>  
9543
```

9544
9545 Please also see the example documents included with the schema.

4 Web Services Binding

4.1 Overview

All services and operations defined in the Domain Model are presented in the Web Services binding.

The WSDL for these services reference two XSD documents. One defines elements for the primary data types of documents, folders, relationships and policies as well as collections of these types of objects.

The second XSD defines the message formats for each of the CMIS services; the messages often refer to the data types defined in the first XSD schema. The WSDL presents exactly the abstract services defined in the Services section.

The normative CMIS Web Services binding is defined by the WSDL and XSD as well as the details given here in this part of the CMIS specification except the examples.

4.1.1 WS-I

A CMIS Web Services binding **MUST** comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

4.1.2 Authentication

A CMIS Web Services binding **SHOULD** support WS-Security 1.1 for Username Token Profile 1.1 and **MAY** also support other authentication mechanisms. A CMIS repository **MAY** grant access to all or a subset of the CMIS services to unauthenticated clients.

4.1.3 Content Transfer

All endpoints of the Web Services binding **MUST** be MTOM enabled.

4.1.4 Reporting Errors

Services **MUST** report errors via SOAP faults. The CMIS-Messaging.xsd defines a basic fault structure that includes an error code and an error message and the WSDL for each service defines specific messages that have the basic fault format.

4.2 Web Services Binding Mapping

The Domain Model defines all services, operations, parameters and objects of CMIS. The Web Services binding is an exact one-to-one mapping of this definition with small exceptions that are explained in the next section. Operations and parameters are named exactly after their counterparts in the Services section. All rules and exceptions defined there apply to the Web Services binding. Optional parameters and optional return values are not set if they are missing or their value is NULL.

4.3 Additions to the Services section

4.3.1 updateProperties and checkIn Semantics

This binding supports partial properties updates. All properties passed to updateProperties or checkIn will be updated to their new values. Properties that are passed without a value will be set to their default value or un-set if no default value is defined. All others property values remain untouched.

4.3.2 Content Ranges

This binding supports the retrieval of content ranges. The operation getContentStream accepts two optional parameters:

- 9582 • **Integer offset:** The first byte of the content to retrieve. Default value is 0.
- 9583 • **Integer length:** The length of the range in bytes. Default value is the size of the content minus
- 9584 the offset.

9585

9586 If the **offset** value is greater than the size of the content the repository SHOULD throw a *constraint*

9587 exception.

9588 If **offset + length** is greater than the size of the content the repository should deliver the content from the

9589 offset to the end of the content.

9590

9591 4.3.3 Extensions

9592 On all input messages and some output messages exists an element called extension. This element is

9593 used to provide vendor or repository-specific information between client and server.

9594 All of the types referenced by the schema also support xs:any for vendor or repository-specific

9595 information.

9596 4.3.4 Web Services Specific Structures

9597 This binding requires specific structures that are not part of the general CMIS schema.

9598 Please also see the example request and response documents included with the schema.

9599 4.3.4.1 cmisFaultType and cmisFault

9600 *cmisFaultType* and *cmisFault* SHOULD be used to generate SOAP faults. See 4.1.4 Reporting

9601 Errors.

9602 4.3.4.2 cmisRepositoryEntryType

9603 *cmisRepositoryEntryType* is the return structure of *getRepositories*. It contains the id and the name

9604 of a repository.

9605 4.3.4.3 cmisTypeContainer

9606 *cmisTypeContainer* is the return structure of *getTypeDescendants*. It holds a type hierarchy.

9607 4.3.4.4 cmisTypeDefinitionListType

9608 *cmisTypeDefinitionListType* is the return structure of *getTypeChildren*. It contains a list of types,

9609 the *hasMoreItems* flag and the *numItem* element.

9610 4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and

9611 cmisObjectInFolderContainerType

9612 *cmisObjectInFolderType* holds, in addition to a *cmisObjectType* object, a path segment string. It

9613 is used in all operations that support the *includePathSegments* parameter.

9614 *cmisObjectParentsType* is similar but has a relative path segment string instead of a path segment.

9615 For details about path segments and relative path segments see section 2.1.5.3 Paths.

9616 *cmisObjectInFolderContainerType* contains a folder hierarchy.

9617 4.3.4.6 cmisObjectListType and cmisObjectInFolderListType

9618 *cmisObjectListType* and *cmisObjectInFolderListType* hold lists of *cmisObjectType* and

9619 *cmisObjectInFolderType* structures. They also contain the *hasMoreItems* flag and the *numItems*

9620 element that are returned by operations that return these lists.

9621 **4.3.4.7 cmisContentStreamType**

9622 `cmisContentStreamType` wraps a content stream and additional information about the stream.

		Client to Repository	Repository to Client
length	Length of the content stream in bytes. If set it MUST be a positive number. If the length is unknown it MUST NOT be set.	SHOULD be set	SHOULD be set
contentType	MIME Media Type of the content stream. For the primary content of a document it SHOULD match the value of the property <code>cmis:contentStreamMimeType</code> .	SHOULD be set	MUST be set
filename	Filename of the content stream. For the primary content of a document it SHOULD match the value of the property <code>cmis:contentStreamFileName</code> .	SHOULD be set	SHOULD be set
stream	The content stream. MUST be present even if the content stream has 0 bytes.	MUST be set	MUST be set

9623

9624 **4.3.4.8 cmisACLType**

9625 `cmisACLType` is the return structure of `getACL` and `applyACL`. It contains the current Access Control List
9626 (ACL) of the object and the `exact` flag that indicates if the ACL fully describes the permission of this
9627 object.

9628 **4.3.4.9 cmisExtensionType**

9629 `cmisExtensionType` is a placeholder for extensions. See 4.3.3 Extensions.

9630

5 IANA Considerations

5.1 Content-Type Registration

5.1.1 CMIS Query

A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

MIME media type name: application

MIME subtype name: cmisquery+xml

Mandatory parameters: None

Optional parameters:

"charset": This parameter has semantics identical to the charset parameter of the "application/xml" media type as specified in [RFC3023].

Encoding considerations:

Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

Security considerations: As defined in this specification.

In addition, as this media type uses the "+xml" convention, it shares the same security considerations as described in [RFC3023], Section 10.

Interoperability considerations:

There are no known interoperability issues.

Published specification: This specification.

Applications that use this media type:

No known applications currently use this media type.

Additional information:

Magic number(s):

As specified for "application/xml" in [RFC3023], Section 3.2.

File extension: .cmisquery

Fragment identifiers:

As specified for "application/xml" in [RFC3023], Section 5.

Base URI:

As specified in [RFC3023], Section 6.

Macintosh File Type code: TEXT

Person and email address to contact for further information:

[OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)

Intended usage: COMMON

Author/Change controller: IESG

5.1.2 CMIS AllowableActions

A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following media type:

9669 MIME media type name: application
9670 MIME subtype name: cmisallowableactions +xml
9671 Mandatory parameters: None.
9672 Optional parameters:
9673 "charset": This parameter has semantics identical to the charset parameter of the
9674 "application/xml" media type as specified in [RFC3023].
9675 Encoding considerations:
9676 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.
9677 Security considerations: As defined in this specification.
9678 In addition, as this media type uses the "+xml" convention, it shares the same security
9679 considerations as described in [RFC3023], Section 10.
9680 Interoperability considerations:
9681 There are no known interoperability issues.
9682 Published specification: This specification.
9683 Applications that use this media type:
9684 No known applications currently use this media type.
9685 Additional information:
9686 Magic number(s):
9687 As specified for "application/xml" in [RFC3023], Section 3.2.
9688 File extension: .cmisallowableactions
9689 Fragment identifiers:
9690 As specified for "application/xml" in [RFC3023], Section 5.
9691 Base URI:
9692 As specified in [RFC3023], Section 6.
9693 Macintosh File Type code: TEXT
9694 Person and email address to contact for further information:
9695 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
9696 Intended usage: COMMON
9697 Author/Change controller: IESG
9698

9699 5.1.3 CMIS Tree

9700 A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:
9701
9702 MIME media type name: application
9703 MIME subtype name: cmistree +xml
9704 Mandatory parameters: None.
9705 Optional parameters:
9706 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media
9707 type as specified in [RFC3023].
9708 Encoding considerations:
9709 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.
9710 Security considerations: As defined in this specification.

9711 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as
9712 described in [RFC3023], Section 10.

9713 Interoperability considerations:

9714 There are no known interoperability issues.

9715 Published specification: This specification.

9716 Applications that use this media type:

9717 No known applications currently use this media type.

9718 Additional information:

9719 Magic number(s):

9720 As specified for "application/xml" in [RFC3023], Section 3.2.

9721 File extension: .cmistree

9722 Fragment identifiers:

9723 As specified for "application/xml" in [RFC3023], Section 5.

9724 Base URI:

9725 As specified in [RFC3023], Section 6.

9726 Macintosh File Type code: TEXT

9727 Person and email address to contact for further information:

9728 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)

9729 Intended usage: COMMON

9730 Author/Change controller: IESG

9731

9732 **5.1.4 CMIS Atom**

9733 A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:

9734

9735 MIME media type name: application

9736 MIME subtype name: cmisatom+xml

9737 Mandatory parameters: None.

9738 Optional parameters:

9739 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media
9740 type as specified in [RFC3023].

9741 "type": This parameter has semantics identical to the type parameter of the "application/atom+xml" as
9742 specified in [RFC4287]

9743 Encoding considerations:

9744 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

9745 Security considerations: As defined in this specification.

9746 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as
9747 described in [RFC3023], Section 10.

9748 Interoperability considerations:

9749 There are no known interoperability issues.

9750 Published specification: This specification.

9751 Applications that use this media type:

9752 No known applications currently use this media type.

9753 Additional information:

9754 Magic number(s):
9755 As specified for "application/xml" in [RFC3023], Section 3.2.
9756 File extension: .cmisatom
9757 Fragment identifiers:
9758 As specified for "application/xml" in [RFC3023], Section 5.
9759 Base URI:
9760 As specified in [RFC3023], Section 6.
9761 Macintosh File Type code: TEXT
9762 Person and email address to contact for further information:
9763 [OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)
9764 Intended usage: COMMON
9765 Author/Change controller: IESG
9766
9767 Please see section 3.1.1 on why this media type is needed above the Atom Media Type.

9768 **5.1.5 CMIS ACL**

9769 A CMIS ACL Document, when serialized as XML 1.0, can be identified with the following media type:
9770
9771 MIME media type name: application
9772 MIME subtype name: cmisacl+xml
9773 Mandatory parameters: None.
9774 Optional parameters:
9775 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media
9776 type as specified in [RFC3023].
9777 Encoding considerations:
9778 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.
9779 Security considerations: As defined in this specification.
9780 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as
9781 described in [RFC3023], Section 10.
9782 Interoperability considerations:
9783 There are no known interoperability issues.
9784 Published specification: This specification.
9785 Applications that use this media type:
9786 No known applications currently use this media type.
9787 Additional information:
9788 Magic number(s):
9789 As specified for "application/xml" in [RFC3023], Section 3.2.
9790 File extension: .cmisacl
9791 Fragment identifiers:
9792 As specified for "application/xml" in [RFC3023], Section 5.
9793 Base URI:
9794 As specified in [RFC3023], Section 6.
9795 Macintosh File Type code: TEXT

9796 Person and email address to contact for further information:
9797 [OASIS](mailto:cmis@lists.oasis-open.org) CMIS TC <cmis@lists.oasis-open.org>
9798 Intended usage: COMMON
9799 Author/Change controller: IESG
9800

6 Conformance

An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level requirements defined within this specification.

Specification:

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.

An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level requirements defined within this specification.

Domain Model:

Normative text within this specification takes precedence over the CMIS Core 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.

Clients:

Client implementations MAY implement either Restful AtomPub Binding or the Web Services Binding.

Repositories:

Repositories MUST implement the following CMIS protocol bindings:

- Restful AtomPub Binding
- Web Services Binding

Rest Binding:

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.

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.

The CMIS RestAtom XML takes precedence over any examples or non-normative outlines included either in this document or as standalone examples.

9844

9845 Web Services Binding:

9846 Normative text within this specification takes precedence over the CMIS Messaging XML and
9847 CMIS WSDL. That is, the normative text in this specification further constrains the schemas and
9848 WSDL that are part of this specification; and this specification contains further constraints on the
9849 elements defined in referenced schemas.

9850 The CMIS Messaging XML and CMIS WSDL takes precedence over any examples or non-
9851 normative outlines included either in this document or as standalone examples.

A. Acknowledgements

The following individuals have participated in the creation of this specification and are gratefully acknowledged:

Participants:

Philippe Allart, Adullact
Florian Bartels, fme AG
Fred Boiscuvier, Exalead, Inc.
Al Brown, IBM
Jay Brown, IBM
Mark Carlson, Sun Microsystems
Derek Carr, IBM
David Caruana, Alfresco Software
Eric Chan, Oracle Corporation
Sameer Charles, Magnolia International AG
Derek Chow, Genus Technologies, LLC
David Choy, EMC Corporation
Scott Conroy, Individual
Cornelia Davis, EMC Corporation
Doug Domeny, Ektron
Kevin Dorr, Flatirons Solutions Corporation
Jason Dubreuil, Fidelity Investments
Michael Duerig, Day Software
Randy Dufault, Genus Technologies, LLC
Will Ezell, dotCMS
Betsy Fanning, AIIM
Steffen Frederiksen, Content Technologies ApS
Stephan Friedl, Quark
Dustin Friesenhahn, Microsoft Corporation
Gary Gershon, Individual
Paul Goetz, SAP AG
Jens Goldhammer, fme AG
Gregory Grefenstette, Exalead, Inc.
Florent Guillaume, Nuxeo
Ethan Gur-esh, Microsoft Corporation
Alexander Haag, WeWebU Software AG
Dennis Hamilton, Individual
Martin Hermes, SAP AG
Jens Huebel, Open Text Corporation
David Izatt, Structured Software Systems Limited (3SL)
Gershon Janssen, Individual
Raphael Jean, Entropysoft
Volker John, Saperion AG
Shane Johnson, Citytech, Inc.
Christophe Kijewska, Adullact
IJonas Kisselbach, Vamosa
Mark Klamerus, Individual
Stephan Klevenz, SAP AG
Boris Kraft, Magnolia International AG
Alison Macmillan, Oracle Corporation
Michael Marth, Day Software
Mary McRae, OASIS
Ryan McVeigh, Oracle Corporation

9905 Juerg Meier, fme AG
 9906 Gregory Melahn, IBM
 9907 Pat Miller, Microsoft Corporation
 9908 Florian Müller, Open Text Corporation
 9909 Thomas Mueller, Day Software
 9910 John Newton, Alfresco Software
 9911 David Nuescheler, Day Software
 9912 Conleth O'Connell, Vignette Corporation
 9913 Marc Pallot, ESoCE-NET
 9914 Rainer Pausch, WeWebU Software AG
 9915 Dominique Pfister, Day Software
 9916 Peeter Piegaze, Day Software
 9917 David Pitfield, Oracle Corporation
 9918 Thomas Pole, Harris Corp
 9919 Norrie Quinn, EMC Corporation
 9920 Craig Randall, Adobe Corporation
 9921 Julian Reschke, Greenbytes GmbH
 9922 Celso Rodriguez, ASG Software Solutions
 9923 Steve Roth, Oracle Corporation
 9924 Patrick Ryan, IBM
 9925 Angela Schreiber, Day Software
 9926 Spencer Shearer, Exalead, Inc.
 9927 Madi Solomon, Pearson PLC
 9928 Wojciech Specht, fme AG
 9929 Dmitri Tcherevik, FatWire
 9930 Jason Tesser, dotCMS
 9931 David Torres, dotCMS
 9932 Maik Uhlenberg, fme AG
 9933 Oliver Walthard, Day Software
 9934 Patrick Ward, Booz Allen Hamilton

9935
 9936 Original Authors of the initial contribution:
 9937 Al Brown, IBM
 9938 David Choy, EMC
 9939 Cornelia Davis, EMC
 9940 Ethan Gur-Esh, Microsoft

9941
 9942 Original Acknowledgements of the initial contribution:
 9943 Al Brown, IBM
 9944 David Caruana, Alfresco
 9945 Derek Carr, IBM
 9946 David Choy, EMC
 9947 Cornelia Davis, EMC
 9948 Paul Goetz, SAP
 9949 Ethan Gur-Esh, Microsoft
 9950 Martin Hermes, SAP
 9951 Jens Hubel, OpenText
 9952 Jay Brown, IBM
 9953 Ryan McVeigh, Oracle
 9954 Gregory Melahn, IBM
 9955 Florian Müller, OpenText
 9956 John Newton, Alfresco
 9957 Norrie Quinn, EMC
 9958 Steve Roth, Oracle
 9959 Craig Randall, EMC

9960

B. Non-Normative Text

9961

C. Revision History

Revision	Date	Editor	Changes Made
1.0	01/11/2010	Al Brown	First specification