

# Content Management Interoperability Services (CMIS) Version 1.0

## OASIS Standard Incorporating Approved Errata 01

~~1 May 2010~~

04 November 2011

### Specification URIs

#### This version:

<http://docs.oasis-open.org/cmisis/CMIS/v1.0/errata-01/os/cmisis-spec-v1.0-errata-01-os-complete.doc> (Authoritative)  
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/errata-01/os/cmisis-spec-v1.0-errata-01-os-complete.html>  
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/errata-01/os/cmisis-spec-v1.0-errata-01-os-complete.pdf>

#### Previous version:

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

#### 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:

David Choy ([mdavid.choy@emc.com](mailto:mdavid.choy@emc.com)), EMC

#### Editors:

~~Al Brown, IBM~~  
~~Ethan Gur-Esh, Microsoft~~  
Ryan McVeigh ([rmcveigh@ziaconsulting.com](mailto:rmcveigh@ziaconsulting.com)), Zia Consulting  
Florian Müller ([florian.mueller@alfresco.com](mailto:florian.mueller@alfresco.com)), Alfresco

#### Additional artifacts:

This prose specification is one component of a Work Product which also includes:

- XML schemas and WSDL: <http://docs.oasis-open.org/cmisis/CMIS/v1.0/errata-01/os/schemas/>
- XML examples: <http://docs.oasis-open.org/cmisis/CMIS/v1.0/errata-01/os/examples/>

#### Related work:

This specification is related to:

*Content Management Interoperability Services (CMIS) Version 1.0*. OASIS Standard.  
<http://docs.oasis-open.org/cmisis/CMIS/v1.0/os/cmisis-spec-v1.0.html>

#### Declared XML namespaces:

- <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 OASIS Content Management Interoperability Services (CMIS) TC on the above date. The level of approval is also listed above. Check the "Latest 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/jpr.php>).

**Citation format:**

When referencing this specification the following citation format should be used:

**[CMIS-v1.0-With-Errata-1]**

*Content Management Interoperability Services (CMIS) Version 1.0*. 04 November 2011. OASIS Standard Incorporating Approved Errata 01. <http://docs.oasis-open.org/cmisis/CMIS/v1.0/errata-01/os/cmisis-spec-v1.0-errata-01-os-complete.html>.

---

## Notices

Copyright © OASIS Open 2011. 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 name "OASIS" is a trademark of [OASIS](#), the owner and developer of this specification, and should be used only to refer to the organization and its official outputs. OASIS welcomes reference to, and implementation and use of, specifications, while reserving the right to enforce its marks against misleading uses. Please see <http://www.oasis-open.org/who/trademark.php> for above guidance.

---

# Table of Contents

1	Introduction .....	10
1.1	Terminology .....	10
1.2	Normative References .....	10
1.3	Non-Normative References .....	10
2	Domain Model .....	11
2.1	Data Model.....	11
2.1.1	Repository .....	11
2.1.1.1	Optional Capabilities .....	11
2.1.1.2	Implementation Information .....	14
2.1.2	Object .....	14
2.1.2.1	Property .....	15
2.1.3	Object-Type .....	16
2.1.3.1	Object-Type Hierarchy and Inheritance .....	17
2.1.3.2	Object-Type Attributes .....	17
2.1.3.3	Object-Type Property Definitions .....	19
2.1.4	Document Object.....	23
2.1.4.1	Content Stream.....	24
2.1.4.2	Renditions.....	24
2.1.4.3	Document Object-Type Definition .....	25
2.1.5	Folder Object .....	33
2.1.5.1	File-able Objects .....	34
2.1.5.2	Folder Hierarchy .....	35
2.1.5.3	Paths.....	36
2.1.5.4	Folder Object-Type Definition .....	37
2.1.6	Relationship Object .....	41
2.1.6.1	Relationship Object-Type Definition.....	42
2.1.7	Policy Object.....	47
2.1.7.1	Policy Object-Type Definition .....	47
2.1.8	Access Control .....	51
2.1.8.1	ACL, ACE, Principal, and Permission .....	51
2.1.8.2	CMIS Permissions .....	51
2.1.8.3	ACL Capabilities .....	52
2.1.9	Versioning.....	61
2.1.9.1	Version Series.....	61
2.1.9.2	Latest Version.....	61
2.1.9.3	Major Versions .....	61
2.1.9.4	Services that modify Version Series .....	62
2.1.9.5	Versioning Properties on Document Objects .....	63
2.1.9.6	Document Creation and Initial Versioning State .....	64
2.1.9.7	Version Specific/Independent membership in Folders .....	64
2.1.9.8	Version Specific/Independent membership in Relationships .....	64
2.1.9.9	Versioning visibility in Query Services .....	65
2.1.10	Query.....	65
2.1.10.1	Relational View Projection of the CMIS Data Model .....	66
2.1.10.2	Query Language Definition .....	67

2.1.10.3 Escaping .....	76
2.1.11 Change Log .....	77
2.1.11.1 Completeness of the Change Log.....	77
2.1.11.2 Change Log Token .....	78
2.1.11.3 Change Event .....	78
2.2 Services .....	78
2.2.1 Common Service Elements.....	78
2.2.1.1 Paging.....	79
2.2.1.2 Retrieving additional information on objects in CMIS service calls .....	79
2.2.1.3 Change Tokens.....	81
2.2.1.4 Exceptions .....	82
2.2.1.5 ACLs .....	85
2.2.2 Repository Services .....	85
2.2.2.1 getRepositories .....	86
2.2.2.2 getRepositoryInfo.....	86
2.2.2.3 getTypeChildren.....	87
2.2.2.4 getTypeDescendants .....	88
2.2.2.5 getTypeDefinition .....	89
2.2.3 Navigation Services.....	89
2.2.3.1 getChildren .....	89
2.2.3.2 getDescendants .....	90
2.2.3.3 getFolderTree .....	91
2.2.3.4 getFolderParent .....	92
2.2.3.5 getObjectParents .....	93
2.2.3.6 getCheckedOutDocs.....	93
2.2.4 Object Services .....	94
2.2.4.1 createDocument.....	94
2.2.4.2 createDocumentFromSource.....	96
2.2.4.3 createFolder.....	97
2.2.4.4 createRelationship .....	98
2.2.4.5 createPolicy .....	99
2.2.4.6 getAllowableActions.....	100
2.2.4.7 getObject .....	101
2.2.4.8 getProperties.....	101
2.2.4.9 getObjectByPath .....	102
2.2.4.10 getContentStream.....	102
2.2.4.11 getRenditions .....	103
2.2.4.12 updateProperties.....	103
2.2.4.13 moveObject.....	104
2.2.4.14 deleteObject.....	105
2.2.4.15 deleteTree.....	105
2.2.4.16 setContentStream .....	106
2.2.4.17 deleteContentStream .....	107
2.2.5 Multi-filing Services .....	107
2.2.5.1 addObjectToFolder .....	107
2.2.5.2 removeObjectFromFolder .....	108
2.2.6 Discovery Services .....	108
2.2.6.1 query.....	108
2.2.6.2 getContentChanges .....	109

2.2.7	Versioning Services.....	111
2.2.7.1	checkOut.....	111
2.2.7.2	cancelCheckOut.....	111
2.2.7.3	checkIn .....	112
2.2.7.4	getObjectOfLatestVersion .....	113
2.2.7.5	getPropertiesOfLatestVersion .....	113
2.2.7.6	getAllVersions.....	114
2.2.8	Relationship Services.....	115
2.2.8.1	getObjectRelationships.....	115
2.2.9	Policy Services.....	116
2.2.9.1	applyPolicy.....	116
2.2.9.2	removePolicy .....	116
2.2.9.3	getAppliedPolicies.....	116
2.2.10	ACL Services.....	117
2.2.10.1	getACL.....	117
2.2.10.2	applyACL .....	117
3	Restful AtomPub Binding .....	119
3.1	Overview .....	119
3.1.1	Namespaces.....	119
3.1.2	Authentication.....	119
3.1.3	Response Formats .....	119
3.1.4	Optional Arguments.....	120
3.1.5	Errors and Exceptions .....	120
3.1.6	Renditions.....	120
3.1.7	Content Streams .....	120
3.1.8	Paging of Feeds .....	120
3.1.9	Services not exposed .....	120
3.2	HTTP.....	121
3.2.1	Entity Tag .....	121
3.2.2	HTTP Range.....	122
3.2.3	HTTP OPTIONS Method.....	122
3.2.4	HTTP Status Codes .....	122
3.2.4.1	General CMIS Exceptions.....	122
3.2.4.2	Notable HTTP Status Codes.....	122
3.3	Media Types .....	122
3.3.1	CMIS Atom .....	123
3.3.2	CMIS Query.....	124
3.3.3	CMIS Allowable Actions .....	124
3.3.4	CMIS Tree .....	125
3.3.5	CMIS ACL.....	129
3.4	Atom Extensions for CMIS.....	130
3.4.1	Atom Element Extensions .....	130
3.4.1.1	AtomPub Workspace .....	130
3.4.1.2	Atom Feed .....	130
3.4.1.3	Atom Entry.....	130
3.4.2	Attributes .....	131
3.4.2.1	cmisra:id .....	131

3.4.2.2 cmisra:renditionKind .....	132
3.4.3 CMIS Link Relations .....	132
3.4.3.1 Existing Link Relations .....	132
3.4.3.2 Hierarchy Navigation Internet Draft Link Relations .....	134
3.4.3.3 Versioning Internet Draft Link Relations .....	134
3.4.3.4 CMIS Specific Link Relations .....	134
3.5 Atom Resources .....	136
3.5.1 Feeds .....	136
3.5.2 Entries .....	137
3.5.2.1 Hierarchical Atom Entries .....	138
3.6 AtomPub Service Document (Repository) .....	140
3.6.1 URI Templates .....	141
3.6.1.1 Object By Id .....	142
3.6.1.2 Object By Path .....	143
3.6.1.3 Query .....	144
3.6.1.4 Type By Id .....	145
3.6.2 HTTP Methods .....	145
3.6.2.1 GET .....	145
3.7 Service Collections .....	146
3.7.1 Root Folder Collection .....	146
3.7.2 Query Collection .....	146
3.7.2.1 POST .....	147
3.7.3 Checked Out Collection .....	149
3.7.3.1 GET .....	149
3.7.3.2 POST .....	149
3.7.4 Unfiled Collection .....	153
3.7.4.1 POST .....	153
3.7.5 Types Children Collection .....	157
3.7.5.1 GET .....	157
3.8 Collections .....	158
3.8.1 Relationships Collection .....	158
3.8.1.1 GET .....	158
3.8.1.2 POST .....	159
3.8.2 Folder Children Collection .....	161
3.8.2.1 GET .....	162
3.8.2.2 POST .....	162
3.8.3 Policies Collection .....	170
3.8.3.1 GET .....	171
3.8.3.2 POST .....	171
3.8.3.3 DELETE .....	171
3.9 Feeds .....	173
3.9.1 Object Parents Feed .....	173
3.9.1.1 GET .....	176
3.9.2 Changes .....	176
3.9.2.1 GET .....	181
3.9.3 Folder Descendants .....	181
3.9.3.1 GET .....	187

3.9.3.2 DELETE .....	187
3.9.4 Folder Tree .....	188
3.9.4.1 GET .....	191
3.9.4.2 DELETE .....	191
3.9.5 AllVersions Feed .....	191
3.9.5.1 GET .....	193
3.9.5.2 DELETE .....	193
3.9.6 Type Descendants Feed .....	193
3.9.6.1 GET .....	201
3.10 Resources .....	201
3.10.1 Type Entry .....	201
3.10.1.1 GET .....	202
3.10.2 Document Entry .....	203
3.10.2.1 GET .....	204
3.10.2.2 PUT .....	206
3.10.2.3 DELETE .....	206
3.10.3 Document Private Working Copy (PWC) Entry .....	206
3.10.3.1 GET .....	207
3.10.3.2 PUT .....	209
3.10.3.3 DELETE .....	209
3.10.4 Folder Entry .....	209
3.10.4.1 GET .....	210
3.10.4.2 PUT .....	212
3.10.4.3 DELETE .....	212
3.10.5 Relationship Entry .....	212
3.10.5.1 GET .....	213
3.10.5.2 PUT .....	214
3.10.5.3 DELETE .....	214
3.10.6 Policy Entry .....	214
3.10.6.1 GET .....	215
3.10.6.2 PUT .....	216
3.10.6.3 DELETE .....	216
3.10.7 Content Stream .....	216
3.10.7.1 GET .....	217
3.10.7.2 PUT .....	217
3.10.7.3 DELETE .....	217
3.10.8 ACL Resource .....	217
3.10.8.1 GET .....	217
4 Web Services Binding .....	219
4.1 Overview .....	219
4.1.1 WS-I .....	219
4.1.2 Authentication .....	219
4.1.3 Content Transfer .....	219
4.1.4 Reporting Errors .....	219
4.2 Web Services Binding Mapping .....	219
4.3 Additions to the Services section .....	219
4.3.1 updateProperties and checkIn Semantics .....	219

4.3.2 Content Ranges .....	219
4.3.3 Extensions .....	220
4.3.4 Web Services Specific Structures .....	220
4.3.4.1 cmisFaultType and cmisFault .....	220
4.3.4.2 cmisRepositoryEntryType .....	220
4.3.4.3 cmisTypeContainer .....	220
4.3.4.4 cmisTypeDefinitionListType .....	220
4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and cmisObjectInFolderContainerType .....	220
4.3.4.6 cmisObjectListType and cmisObjectInFolderListType .....	221
4.3.4.7 cmisContentStreamType .....	221
4.3.4.8 cmisACLType.....	221
4.3.4.9 cmisExtensionType.....	221
5 IANA Considerations .....	222
5.1 Content-Type Registration.....	222
5.1.1 CMIS Query.....	222
5.1.2 CMIS AllowableActions .....	222
5.1.3 CMIS Tree .....	223
5.1.4 CMIS Atom .....	224
5.1.5 CMIS ACL.....	225
6 Conformance.....	227
A. Acknowledgements .....	229
B. Non-Normative Text .....	231
C. Revision History.....	232

---

# 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><http://www.ietf.org/id/draft-brown-versioning-link-relations-08.txt>, 2010
- [ID-WebLinking] M. Nottingham, *Web Linking*, <http://tools.ietf.org/id/draft-nottingham-http-link-header-07.txt><http://tools.ietf.org/id/draft-nottingham-http-link-header-08.txt>, 2010

## 1.3 Non-Normative References

---

## 2 Domain Model

### 2.1 Data Model

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

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

#### 2.1.1 Repository

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

##### 2.1.1.1 Optional Capabilities

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

#### Navigation Capabilities:

##### **capabilityGetDescendants**

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

[See section: 2.2.3.2 getDescendants](#)

##### **capabilityGetFolderTree**

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

[See section: 2.2.3.3 getFolderTree](#)

## 78 **Object Capabilities:**

79 **capabilityContentStreamUpdatability** (**enumCapabilityContentStreamUpdates**)

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

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

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

- 86 | • Versioning.

87 | See Section: 2.1.4.1 Content Stream

88

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

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

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

99 | See Section: 2.1.11 Change Log

100

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

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

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

105

## 106 **Filing Capabilities:**

107 **capabilityMultifiling**

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

109 | See Section: 2.1.5 Folder Object

110

111 **capabilityUnfiling**

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

113 | See Section: 2.1.5 Folder Object

114

115 **capabilityVersionSpecificFiling**

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

117 | See Section: 0

118 | Versioning

119

## 120 **Versioning Capabilities:**

121       **capabilityPWCUpdatable**  
122            Ability for an application to update the `"Private Working Copy"` of a checked-out document  
123    See Section: 0  
124            Versioning

125  
126       **capabilityPWCSearchable**  
127            Ability of the Repository to include the "Private Working Copy" of checked-out documents in  
128            query search scope; otherwise PWC's are not searchable  
129    See Section: 0  
130            Versioning

131  
132       **capabilityAllVersionsSearchable**  
133            Ability of the Repository to include all versions of document. If False, typically either the latest or  
134            the latest major version will be searchable.  
135    See Section: 0  
136            Versioning

#### 138    **Query Capabilities:**

139       **capabilityQuery (enumCapabilityQuery)**  
140            Indicates the types of queries that the Repository has the ability to fulfill. Query support levels are:  
141            

- 142            • **none:** No queries of any kind can be fulfilled.
- 143            • **metadataonly:** Only queries that filter based on object properties can be fulfilled. Specifically, the CONTAINS() predicate function is not supported.
- 144            • **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.
- 145            • **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.
- 146            • **bothcombined:** The repository can fulfill queries that filter on both the full-text content of documents and their properties in the same query.

  
147            See Section: 2.1.10 Query

148  
149  
150       **capabilityJoin (enumCapabilityJoin)**  
151            Indicates the types of JOIN keywords that the Repository can fulfill in queries. Support levels are:  
152            

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

  
156            See Section: 2.1.10 Query

#### 163    **ACL Capabilities:**

164       **capabilityACL (enumCapabilityACL)**

165 Indicates the level of support for ACLs by the repository  
166 |

- **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)

168 |  
169 |  
170 See Section: 2.8 Access Control

## 171 2.1.1.2 Implementation Information

172 | The `Get Repository Information` service MUST also return implementation information including vendor  
173 | name, product name, product version, version of CMIS that it supports, the root folder ID (see section  
174 | 2.1.5.2 Folder Hierarchy), and MAY include other implementation-specific information. The version of  
175 | CMIS that the repository supports MUST be expressed as a Decimal that matches the specification  
176 | version.

## 177 2.1.2 Object

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

- 180 | • A *document object* represents a standalone information asset. Document objects are the  
181 | elementary entities managed by a CMIS repository.
- 182 | • A *folder object* represents a logical container for a collection of `file-able` objects, which include  
183 | folder objects and document objects. Folder objects are used to organize file-able objects.  
184 | Whether or not an object is file-able is specified in its ~~object-type definition~~object-type definition.
- 185 | • A *relationship object* represents an instance of directional relationship between two objects. The  
186 | support for relationship objects is optional, and may be discovered via the `Get Type Children`  
187 | service.
- 188 | • A *policy object* represents an administrative policy, which may be `applied` to one or more  
189 | `controllablePolicy` objects. Whether or not an object is controllable is specified in its object-  
190 | type definition. The support for policy objects is optional, and may be discovered via the `Get`  
191 | `Type Children` service.

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

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

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

204 In addition, a document object MAY have a **Content-Stream**, which may be used to hold a raw digital  
205 | asset such as an image or a word-processing document. A repository MUST specify, in each object-type  
206 | definition, whether document objects of that type MAY, MUST, or MUST NOT have a content-stream. A

207 document MAY also have one or more **Renditions** associated with it. A rendition can be a thumbnail or  
208 an alternate representation of the content stream.

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

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

214 CMIS objects MAY expose additional information, such as vendor-specific workflow data, beyond the  
215 attributes described above. In this respect, the data model can be extended as desired. This specification  
216 does not standardize such extensions.

### 217 2.1.2.1 Property

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

222 If a value is not provided for a property, the property is in a "value not set" state. There is no "null" value  
223 for a property. Through protocol binding, a property is either single-valued or multi-valued, MAY be in a  
224 "not set" state. CMIS does not support "null" property value.

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

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

231 Every property is typed. The Property-type defines the data type of the data value(s) held by the property.  
232 CMIS specifies the following Property-types. They include the following data types defined by XML  
233 Schema Part 2: Datatypes Second Edition<sup>3</sup> (W3C Recommendation, 28 October 2004,  
234 <http://www.w3.org/TR/xmlschema-2/>):

- 235 • string (xsd:string)
- 236 • boolean (xsd:boolean)
- 237 • decimal (see section 2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions)
- 238 • integer (xsd:integer)
- 239 • datetime (xsd:dateTime and see section 2.1.3.3.5 Attributes specific to Decimal Object-Type  
240 Property Definitions)
- 241 • uri (xsd:anyURI)

242  
243

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

- 245 • id
  - 246 • html
- 247 Individual protocol bindings MAY override or re-specify these property types.

248

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

- 252 |
- 253 |       The string MUST NOT contain:
- 254 |       • whitespace `" "`,
- 255 |       • comma `,`
- 256 |       • double quotes `"` and `"`,
- 257 |       • single quotes `'` and `'`,
- 258 |       • backslash `\`
- 259 |       • the period `.` character or,
- 260 |       • the open `(` or close `)` parenthesis characters.
- 261 |

### 262 | **2.1.2.1.1 ID Property**

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

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

### 278 | **2.1.2.1.2 HTML Property**

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

## 282 | **2.1.3 Object-Type**

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

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

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

290 | A CMIS repository MUST expose exactly one collection of Object-Types via the `Repository` services  
291 | (`getTypeChildren`, `getTypeDescendants`, `getTypeDefinition`).

292 | While a repository MAY define additional object-types beyond the [CMIS-Base-Object-Types, CMIS Base](#)  
293 | [Object-Types](#), these Object-Types MUST NOT extend or alter the behavior or semantics of a CMIS

294 service (for example, by adding new services). A repository MAY attach additional constraints to an  
295 object-type underneath CMIS, provided that the effect visible through the CMIS interface is consistent  
296 with the allowable behavior of CMIS.

### 297 2.1.3.1 Object-Type Hierarchy and Inheritance

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

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

### 328 2.1.3.2 Object-Type Attributes

#### 329 2.1.3.2.1 Attributes common to ALL Object-Type Definitions

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

331	<code>id</code>	ID
332	This opaque attribute identifies this object-type in the repository.	
333		
334	<code>localName</code>	String <del>(optional)</del>

335 | This attribute represents the underlying repository's name for the object-type. This field is  
336 | opaque and has no uniqueness constraint imposed by this specification.  
337 | Two properties with the same localName and localNamespace MUST have the same semantic  
338 | equality.

339 |  
340 | **localNamespace** String (optional)

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

343 |  
344 | **queryName** String

345 | Used for query and filter operations on object-types. This is an opaque String with limitations.  
346 | This string SHOULD NOT contain any characters that negatively interact with the BNF grammar.

347 |  
348 | The string MUST NOT contain:

- 349 | • whitespace `" "`
- 350 | • comma `","`
- 351 | • double quotes `"""`
- 352 | • single quotes `'`
- 353 | • backslash `"\"`
- 354 | • the period `."` character or,
- 355 | • the open `"("` or close `)"` parenthesis characters.

356 |  
357 | **displayName** String (optional)

358 | Used for presentation by application.

359 |  
360 | **baseId** Enum

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

363 |  
364 | **parentId** ID

365 | The ID of the Object-Type's immediate parent type.

366 | It MUST be `"not set"` for a base type.

367 |  
368 | **description** String (optional)

369 | Description of this object-type, such as the nature of content, or its intended use. Used for  
370 | presentation by application.

371 |  
372 | **creatable** Boolean

373 | Indicates whether new objects of this type MAY be created. If the value of this attribute is FALSE,  
374 | the repository MAY contain objects of this type already, but MUST NOT allow new objects of this  
375 | type to be created.

376 |  
377 | **fileable** Boolean

378 | Indicates whether or not objects of this type are ~~file-able~~ file-able.



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

422

423 **queryName** String

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

426

427 **displayName** String (optional)

428 Used for presentation by application.

429

430 **description** String (optional)

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

432

433 **propertyType** Enum

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

436

437 **cardinality** Enum

438 Indicates whether the property can have `"_zero or one_"` or `"_zero or more_"` values.

439 Values:

- 440 • **single**: Property can have zero or one values (if property is not required), or exactly one  
441 value (if property is required)
- 442 • **multi**: Property can have zero or more values (if property is not required), or one or more  
443 values (if property is required).

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

447

448 **updatability** Enum

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

450 Values:

- 451 • **readonly**: The value of this property MUST NOT ever be set directly by an application. It  
452 is a system property that is either maintained or computed by the repository.
  - 453 ○ The value of a `readOnly` property MAY be indirectly modified by other repository  
454 interactions (for example, calling `"_updateProperties_"` on an object will change the  
455 object's last modified date, even though that property cannot be directly set via an  
456 `updateProperties()` service call.)
- 457 • **readwrite**: The property value can be modified using the `updateProperties` service.
- 458 • **whencheckedout**: The property value MUST only be update-able using a `"_private  
459 working copy_"_private working copy_` Document.
  - 460 ○ I.e. the update is either made on a `"_private working copy_"` object or made using a  
461 `"_check in_"` service.
- 462 • **oncreate**: The property value MUST only be update-able during the Create operation on  
463 that Object.

464

465       **inherited**        Boolean

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

467            explicitly defined for this object-type when FALSE.

468

469       **required**        Boolean

470

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

472            by the application.

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

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

475            MUST provide a value for this property.

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

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

478            exception.

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

480            `"required"` SHOULD be set to FALSE.

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

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

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

484

485       **queryable**        Boolean

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

487            statement.

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

489            to FALSE.

490

491       **orderable**        Boolean

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

493            or an ORDERBY parameter.

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

495

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

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

498            If the cardinality of the property definition is `"single"` and the `"openChoice"` attribute is FALSE,

499            then the property value MUST be at most one of the values listed in this attribute.

500            If the cardinality of the property definition is `"single"` and the `"openChoice"` attribute is TRUE,

501            then the property value MAY be one of the values listed in this attribute.

502            If the cardinality of the property definition is `"multi"` and the `"openChoice"` attribute is FALSE,

503            then the property value MUST be zero, one or more than one of the values listed in this attribute.

504            If the cardinality of the property definition is `"multi"` and the `"openChoice"` attribute is TRUE,

505            then the property value MAY be zero, one, or more than one of the values listed in this attribute. If

506            this attribute is `"not set"`, then any valid value for this property based on its type may be used.

507            Each choice includes a `displayName` and a `value`. The `displayName` is used for presentation

508            purpose. The `value` will be stored in the property when selected.

509            Choices MAY be hierarchically presented. For example: a value of `"choices"` for a geographic

510            location would be represented as follows:

511                o Europe:

- 512                           ▪ England
- 513                           ▪ France
- 514                           ▪ Germany
- 515                   ○ North America
- 516                           ▪ Canada
- 517                           ▪ USA
- 518                           ▪ Mexico
- 519    **openChoice**        Boolean

520           This attribute is only applicable to properties that provide a value for the `Choices` attribute.  
 521           If FALSE, then the data value for the property MUST only be one of the values specified in the  
 522           `Choices` attribute. If TRUE, then values other than those included in the `Choices` attribute  
 523           may be set for the property.

524

525    **defaultValue**    <PropertyType>

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

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

532           The attributes on the default value element are the same as the attributes on the property  
 533           definition.

### 534    2.1.3.3.3 Attributes specific to Integer Object-Type Property Definitions

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

539    **minValue**        Integer

540           The minimum value allowed for this property.

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

544    **maxValue**        Integer

545           The maximum value allowed for this property.

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

### 549    2.1.3.3.4 Attributes specific to DateTime Object-Type Property Definitions

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

554    **resolution**       String Enumeration

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

- 556 • Year: Year resolution is persisted
- 557 • Date: Date resolution is persisted
- 558 • Time: Time resolution is persisted
- 559

### 560 2.1.3.3.5 Attributes specific to Decimal Object-Type Property Definitions

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

565 **precision** Integer Enumeration

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

- 567 • 32: 32-bit precision (`"single"` as specified in IEEE-754-1985).
- 568 • 64: 64-bit precision (`"double"` as specified in IEEE-754-1985.)

569

570 **minValue** Decimal

571 The minimum value allowed for this property.

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

574

575 **maxValue** Decimal

576 The maximum value allowed for this property.

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

### 579 2.1.3.3.6 Attributes specific to String Object-Type Property Definitions

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

584 **maxLength** Integer

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

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

## 588 2.1.4 Document Object

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

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

- 591 • **Version-able:** Can be acted upon via the Versioning Services (for example: `checkOut`,  
592 `checkIn`), `checkOut`, `checkIn`).
- 593 • **File-able:** Can be filed in zero, one, or more than one folder via the Multi-filing services.
- 594 • **Query-able:** Can be located via the Discovery Services (`query`).

- 595 • **Controllable-Policy:** Can have Policies applied to it (see section 2.1.7 Policy Object.)
- 596 • **Controllable-ACL:** Can have an ACL applied to it (see section 2.8 Access Control)

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

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

### 601 2.1.4.1 Content Stream

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

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

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

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

### 621 2.1.4.2 Renditions

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

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

#### 634 2.1.4.2.1 Rendition Attributes

635 A rendition has the following attributes:

636	<b>streamId</b>	ID
637	Identifies the rendition stream.	

638		
639	<b>contentType</b>	String
640		The MIME type of the rendition stream.
641		
642	<b>length</b>	Integer (optional)
643		The length of the rendition stream in bytes.
644		
645	<b>title</b>	String (optional)
646		Human readable information about the rendition.
647		
648	<b>kind</b>	String
649		A categorization String associated with the rendition.
650		
651	<b>height</b>	Integer (optional)
652		Typically used for <code>image</code> renditions (expressed as pixels). SHOULD be present if kind =
653		<code>cmis:thumbnail</code> .
654		
655	<b>width</b>	Integer (optional)
656		Typically used for <code>image</code> renditions (expressed as pixels). SHOULD be present if kind =
657		<code>cmis:thumbnail</code> .
658		
659	<b>renditionDocumentId</b>	ID (optional)
660		If specified, then the rendition can also be accessed as a document object in the CMIS services.
661		If not set, then the rendition can only be accessed via the rendition services. Referential integrity
662		of this ID is repository-specific.

#### 663 2.1.4.2.2 Rendition Kind

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

667 CMIS defines the following kind:

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

#### 671 2.1.4.3 Document Object-Type Definition

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

##### 675 2.1.4.3.1 Attributes specific to Document Object-Types

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

678	<b>versionable</b>	Boolean
-----	--------------------	---------

679 Indicates whether or not objects of this type are version-able. (See section 0  
680 Versioning.)

681  
682 **contentStreamAllowed** Enum

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

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

### 689 2.1.4.3.2 Attribute Values

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

691 Notes:

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

696

697 **id**

698 Value: cmis:document

699

700 **localName**

701 Value: <repository-specific>

702

703 **localNamespace**

704 Value: <repository-specific>

705

706 **queryName**

707 Value: cmis:document

708

709 **displayName**

710 Value: <repository-specific>

711

712 **baseId**

713 Value: cmis:document

714

715 **parentId**

716 Value: Not set

717

718 **description**

719 Value: <repository-specific>

720

721       **creatable**  
722           Value: <repository-specific>  
723  
724       **fileable**  
725           Value: TRUE  
726  
727       **queryable**  
728           Value: SHOULD be TRUE  
729  
730       **controllablePolicy**  
731           Value: <repository-specific>  
732  
733       **includedInSupertypeQuery**  
734           Value: <repository-specific>  
735  
736       **versionable**  
737           Value: <repository-specific>  
738  
739       **contentStreamAllowed**  
740           Value: <repository-specific>  
741  
742       **controllableACL**  
743           Value: <repository-specific>  
744  
745       **fulltextIndexed**  
746           Value: <repository-specific>

### 747   **2.1.4.3.3 Property Definitions**

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

752		
753	<b>cmis:name</b>	Name of the object
754	Inherited:	False
755	Property Type:	String
756	Cardinality:	Single
757		
758	<b>cmis:objectId</b>	Id of the object
759	Required:	False
760	Inherited:	False
761	Property Type:	ID
762	Cardinality:	Single
763	Updatability:	Read Only

764	Choices:	Not Applicable
765	Open Choice:	Not Applicable
766	Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them	
767		
768		
769		
770	<b>cmis:baseTypeid</b>	Id of the base object-type for the object
771	Required:	False
772	Inherited:	False
773	Property Type:	ID
774	Cardinality:	Single
775	Updatability:	Read Only
776	Choices:	Not Applicable
777	Open Choice:	Not Applicable
778	Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them	
779		
780		
781	<b>cmis:objectTypeid</b>	Id of the object's type
782	Required:	True
783	Inherited:	False
784	Property Type:	ID
785	Cardinality:	Single
786	Updatability:	oncreate
787	Choices:	Not Applicable
788	Open Choice:	Not Applicable
789	Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them	
790		
791		
792	<b>cmis:createdBy</b>	User who created the object.
793	Required:	False
794	Inherited:	False
795	Property Type:	String
796	Cardinality:	Single
797	Updatability:	Read Only
798	Choices:	Not Applicable
799	Open Choice:	Not Applicable
800	Queryable:	True
801	Orderable:	True
802	Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them	
803		
804		
805	<b>cmis:creationDate</b>	DateTime when the object was created.
806	Required:	False

807	Inherited:	False
808	Property Type:	DateTime
809	Cardinality:	Single
810	Updatability:	Read Only
811	Choices:	Not Applicable
812	Open Choice:	Not Applicable
813	Queryable:	True
814	Orderable:	True
815	Repository MUST return this property with non-empty values when an object is requested and the	
816	property filter does not exclude them	
817		
818	<b>cmis:lastModifiedBy</b>	User who last modified the object.
819	Required:	False
820	Inherited:	False
821	Property Type:	String
822	Cardinality:	Single
823	Updatability:	Read Only
824	Choices:	Not Applicable
825	Open Choice:	Not Applicable
826	Queryable:	True
827	Orderable:	True
828	Repository MUST return this property with non-empty values when an object is requested and the	
829	property filter does not exclude them	
830		
831	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
832	Required:	False
833	Inherited:	False
834	Property Type:	DateTime
835	Cardinality:	Single
836	Updatability:	Read Only
837	Choices:	Not Applicable
838	Open Choice:	Not Applicable
839	Queryable:	True
840	Orderable:	True
841	Repository MUST return this property with non-empty values when an object is requested and the	
842	property filter does not exclude them	
843		
844	<b>cmis:changeToken</b>	Opaque token used for optimistic locking & concurrency
845		checking. (see section 2.2.1.3 Change Tokens)
846	Required:	False
847	Inherited:	False
848	Property Type:	String
849	Cardinality:	Single

850	Updatability:	Read Only
851	Choices:	Not Applicable
852	Open Choice:	Not Applicable
853	Repository MUST return this property with non-empty values when an object is requested and the	
854	property filter does not exclude them. <u>If the repository does not support change tokens, this</u>	
855	<u>property SHOULD not be set.</u>	
856		
857	<b>cmis:isImmutable</b>	TRUE if the repository MUST throw an error at any attempt to
858		update or delete the object.
859	Required:	False
860	Inherited:	False
861	Property Type:	Boolean
862	Cardinality:	Single
863	Updatability:	Read Only
864	Choices:	Not Applicable
865	Open Choice:	Not Applicable
866	Repository MUST return this property with non-empty values when an object is requested and the	
867	property filter does not exclude them	
868		
869	<b>cmis:isLatestVersion</b>	See section 0
870	Versioning.	
871	Required:	False
872	Inherited:	False
873	Property Type:	Boolean
874	Cardinality:	Single
875	Updatability:	Read Only
876	Choices:	Not Applicable
877	Open Choice:	Not Applicable
878	Repository MUST return this property with non-empty values when an object is requested and the	
879	property filter does not exclude them. Version Property Values are repository-specific when a	
880	document is defined as non-versionable.	
881		
882	<b>cmis:isMajorVersion</b>	See section 0
883	Versioning.	
884	Required:	False
885	Inherited:	False
886	Property Type:	Boolean
887	Cardinality:	Single
888	Updatability:	Read Only
889	Choices:	Not Applicable
890	Open Choice:	Not Applicable
891	Repository MUST return this property with non-empty values when an object is requested and the	
892	property filter does not exclude them. Version Property Values are repository-specific when a	
893	document is defined as non-versionable.	
894		

895 **cmis:isLatestMajorVersion** See section 0

896 Versioning.

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

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

907

908 **cmis:versionLabel** See section 0

909 Versioning.

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

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

920

921 **cmis:versionSeriesId** See section 0

922 Versioning.

923 Required: False  
924 Inherited: False  
925 Property Type: ID  
926 Cardinality: Single  
927 Updatability: Read Only  
928 Choices: Not Applicable  
929 Open Choice: Not Applicable

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

933

934 **cmis:isVersionSeriesCheckedOut** See section 0

935 Versioning.

936 Required: False  
937 Inherited: False  
938 Property Type: Boolean

939	Cardinality:	Single
940	Updatability:	Read Only
941	Choices:	Not Applicable
942	Open Choice:	Not Applicable
943	Repository MUST return this property with non-empty values when an object is requested and the	
944	property filter does not exclude them. Version Property Values are repository-specific when a	
945	document is defined as non-versionable.	
946		
947	<b>cmis:versionSeriesCheckedOutBy</b>	See section 0
948	Versioning.	
949	Required:	False
950	Inherited:	False
951	Property Type:	String
952	Cardinality:	Single
953	Updatability:	Read Only
954	Choices:	Not Applicable
955	Open Choice:	Not Applicable
956	Version Property Values are repository-specific when a document is defined as non-versionable.	
957		
958	<b>cmis:versionSeriesCheckedOutId</b>	See section 0
959	Versioning.	
960	Required:	False
961	Inherited:	False
962	Property Type:	ID
963	Cardinality:	Single
964	Updatability:	Read Only
965	Choices:	Not Applicable
966	Open Choice:	Not Applicable
967	Version Property Values are repository-specific when a document is defined as non-versionable.	
968		
969	<b>cmis:checkinComment</b>	See section 0
970	Versioning.	
971	Required:	False
972	Inherited:	False
973	Property Type:	String
974	Cardinality:	Single
975	Updatability:	Read Only
976	Choices:	Not Applicable
977	Open Choice:	Not Applicable
978	Version Property Values are repository-specific when a document is defined as non-versionable.	
979		
980	<b>cmis:contentStreamLength</b>	Length of the content stream (in bytes).
981	Required:	False

982	Inherited:	False
983	Property Type:	Integer
984	Cardinality:	Single
985	Updatability:	Read Only
986	Choices:	Not Applicable
987	Open Choice:	Not Applicable
988	Repository MUST return this property with non-empty values when an object is requested and the	
989	property filter does not exclude them and if the document has a content stream	
990		
991	<b>cmis:contentStreamMimeType</b>	MIME type of the Content Stream
992	Required:	False
993	Inherited:	False
994	Property Type:	String
995	Cardinality:	Single
996	Updatability:	Read Only
997	Choices:	Not Applicable
998	Open Choice:	Not Applicable
999	Repository MUST return this property with non-empty values when an object is requested and the	
1000	property filter does not exclude them and if the document has a content stream	
1001		
1002	<b>cmis:contentStreamFileName</b>	File name of the Content Stream
1003	Required:	False
1004	Inherited:	False
1005	Property Type:	String
1006	Cardinality:	Single
1007	Repository MUST return this property with non-empty values when an object is requested and the	
1008	property filter does not exclude them and if the document has a content stream	
1009		
1010	<b>cmis:contentStreamId</b>	Id of the stream
1011	Required:	False
1012	Inherited:	False
1013	Property Type:	ID
1014	Cardinality:	Single
1015	Updatability:	Read Only
1016	Choices:	Not Applicable
1017	Open Choice:	Not Applicable

## 1018 2.1.5 Folder Object

1019 A folder object serves as the anchor for a collection of *file-able* objects. The folder object has an *implicit*  
1020 hierarchical relationship with each object in its collection, with the anchor folder object being the **Parent**  
1021 object and each object in the collection being a **Child** object. This implicit relationship has specific  
1022 containment semantics which MUST be maintained by the repository with implicit referential integrity.  
1023 (That is, there will never be a dangling parent-relationship or a dangling child-relationship. Furthermore,  
1024 object A is a parent of object B if and only if object B is a child of object A.) This system-maintained

1025 implicit relationship is distinct from an *explicit* relationship which is instantiated by an application-  
1026 maintained Relationship Object. (See section 2.1.6 Relationship Object.)

1027 A folder object does not have a content-stream and is not version-able. A folder object MAY be  
1028 associated with zero or more renditions (see section 2.1.4.2 Renditions).

### 1029 2.1.5.1 File-able Objects

1030 | A *file-able* object is one that MAY be `filed` into a folder. That is, it MAY be a child object of a folder  
1031 object. The following list defines whether the base CMIS Object-types are file-able:

1032 **cmis:folder**

1033 MUST be file-able

1034

1035 **cmis:document**

1036 MUST be file-able

1037

1038 **cmis:relationship**

1039 MUST NOT be file-able

1040

1041 **cmis:policy**

1042 MAY be file-able

#### 1043 2.1.5.1.1 Document Version Series and Filing

1044 | Since document objects are versionable, a document object's membership in a folder MAY be version-  
1045 specific or version-independent. That is, the folder membership MAY be restricted to that particular  
1046 version of the document or MAY apply to all versions of the document. Whether or not a repository  
1047 | supports version-specific filing is discoverable via the `Get Repository Information` service  
1048 (*getRepositoryInfo*).

1049 When the child objects of a folder are retrieved, a specific version of a document MAY be returned. If the  
1050 repository supports version-specific filing, the specific version filed in that folder is returned. If the  
1051 repository does not support version-specific filing, the latest version of the document is returned.

1052 Likewise, this version sensitivity in child-binding also affects the behavior of parent retrieval for a  
1053 document object, as well as the scope of the `IN_FOLDER()` and `IN_TREE()` function calls in a query. For  
1054 non-versionable fileable objects, their membership in a folder does not have version sensitivity.

#### 1055 2.1.5.1.2 Filing Restrictions by Object-Type

1056 | A folder collection's membership MAY be restricted by object-type. Each folder object has a multi-valued  
1057 *AllowedChildObjectTypes* property, which specifies that only objects of these types are allowed to be  
1058 | its children. If this property is `not set`, then objects of any file-able type MAY be filed in the Folder. It is  
1059 repository-specific if subtypes of the types listed in the *AllowedChildObjectTypes* property MAY be filed  
1060 in the folder.

1061 Because of these filing constraints, when a new folder object is created, an existing folder object MUST  
1062 be specified as its parent.

1063 When a non-file-able object is created, a parent folder MUST NOT be specified.

1064 When a file-able object is deleted, it is removed from any folder collection in which the object is a  
1065 member. In other words, when an object is deleted, all implicit parent-child relationships with the deleted  
1066 object as a child cease to exist.

## 1067 2.1.5.2 Folder Hierarchy

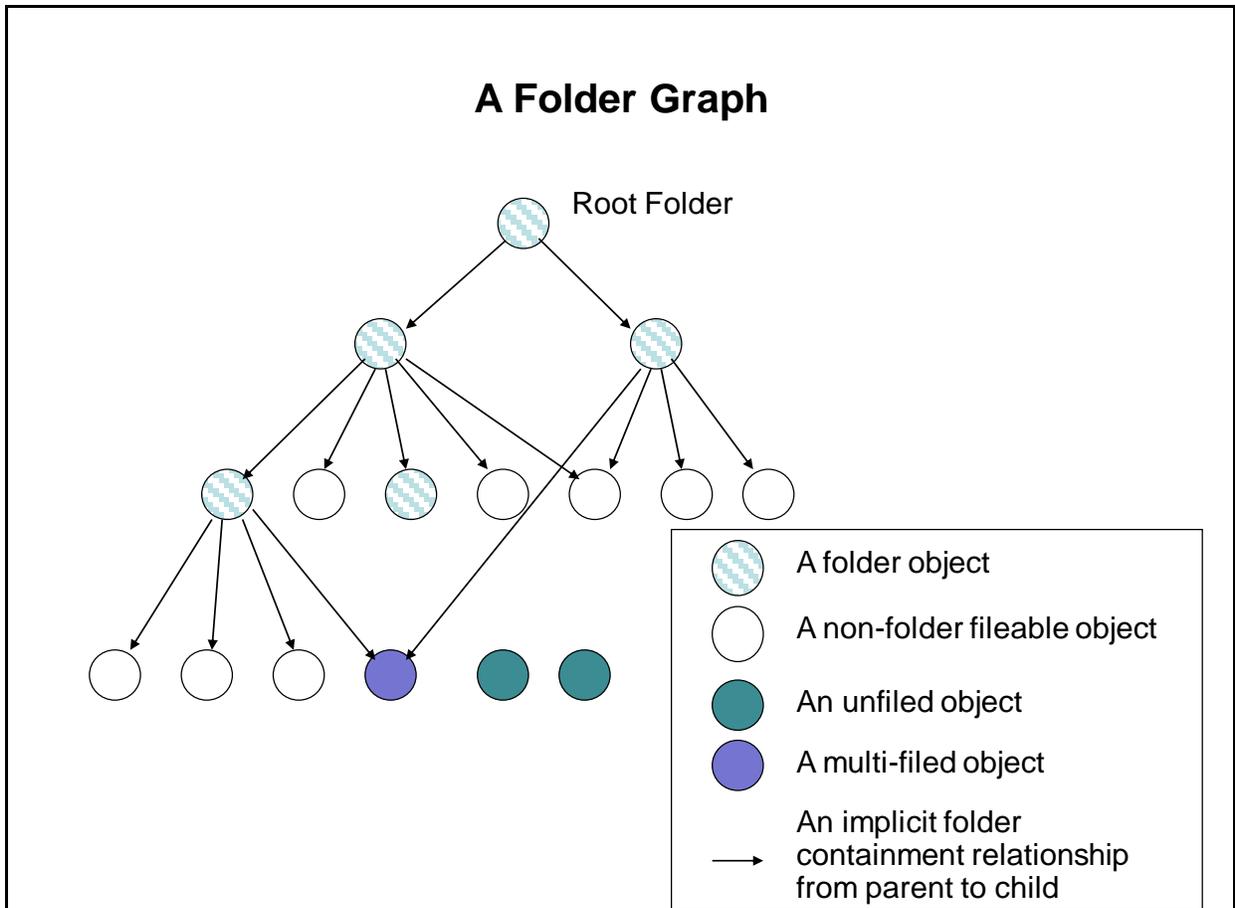
1068 CMIS imposes the following constraints on folder objects:

- 1069 | • Every folder object, except for one which is called the **Root Folder**, MUST have one and only  
1070 | one parent folder. The Root Folder does not have a parent.
- 1071 | • A cycle in folder containment relationships is not allowed. That is, a folder object cannot have  
1072 | itself as one of its descendant objects.
- 1073 | • A child object that is a folder object can itself be the parent object of other file-able objects.

1074 With these constraints, the folder objects in a CMIS repository necessarily form a strict hierarchy, with the  
1075 Root Folder being the root of the hierarchy.

1076 The child objects of a given folder object, their child objects, and grandchild objects, etc., are called  
1077 **Descendant** objects of the given folder object. A folder object together with all its descendant objects are  
1078 collectively called a **Tree** rooted at that folder object.

1079 A non-folder object does not have any descendant object. Thus, a **Folder Graph** that consists of all  
1080 fileable objects as nodes, and all the implicit folder containment relationships as directed edges from  
1081 parent to child, is a directed acyclic graph, possibly with some disconnected (orphan) nodes. It follows  
1082 that the tree rooted at any given folder object is also a directed acyclic graph, although a non-folder object  
1083 in the tree MAY have ancestors that are not ancestors of the rooted folder.



1084

1085 Folder objects are handled using ~~the basic CRUD services for objects,~~[the basic CRUD services for](#)  
 1086 [objects](#), and the folder graph is traversed using the ~~Navigation Services.~~[Navigation Services](#).

1087 The **Root Folder** is a special folder such that it cannot be created, deleted, or moved using CMIS  
 1088 services. Otherwise, it behaves like any other folder object.

### 1089 2.1.5.3 Paths

1090 A folder hierarchy MAY be represented in a canonical notation such as path. For CMIS, a path is  
 1091 represented by:

- 1092 • ~~"/~~ for the root folder
- 1093 • All paths start with the root folder.
- 1094 • A set of the folder and object path segments separated by ~~"/~~ in order of closest to the root.
- 1095 • Folder and object path segments are specified by pathSegment tokens which can be retrieved by  
 1096 all services that take an `includePathSegments` parameter.
- 1097 • A pathSegment token MUST not include a ~~"/~~ character.
  - 1098 ○ It is repository specific how a repository chooses the value for pathSegment.
  - 1099 Repositories might choose to use `cmis:name` or content stream filename for
  - 1100 pathSegment token.
- 1101 • The pathSegment token for each item MUST uniquely identify the item in the folder.

1102

1103 E.g., if folder A is under the root, and folder B is under A, then the path would be /A/B.

1104 | A path for an object may be calculated by taking the item's parent folder cmis:path property and  
1105 | appending the `"#"` character and the object's pathSegment. This constructed path may be given as  
1106 | input to the `getObjectByPath` service for object by path retrieval.

1107 | The `getObjectParents` service returns `relativePathSegment` tokens. These tokens are the  
1108 | pathSegment of the input object relative to the parent folders.

## 1109 | 2.1.5.4 Folder Object-Type Definition

1110 | This section describes the definition of the Folder Object-Type's attribute values and property definitions  
1111 | which must be present on Folder instance objects. All attributes and property definitions are listed by  
1112 | their ID.

### 1113 | 2.1.5.4.1 Attribute Values

1114 | The Folder Object-Type MUST have the following attribute values.

1115 | Notes:

- 1116 | • A value of `<repository-specific>` indicates that the value of the property MAY be set to any valid  
1117 | value for the attribute type.
- 1118 | • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the  
1119 | Object-Type definition.

1120

1121 | **id**

1122 | Value: `cmis:folder`

1123

1124 | **localName**

1125 | Value: `<repository-specific>`

1126

1127 | **localNamespace**

1128 | Value: `<repository-specific>`

1129

1130 | **queryName**

1131 | Value: `cmis:folder`

1132

1133 | **displayName**

1134 | Value: `<repository-specific>`

1135

1136 | **baseId**

1137 | Value: `cmis:folder`

1138

1139 | **parentId**

1140 | Value: Not set

1141

1142 | **description**

1143 | Value: `<repository-specific>`

1144

1145 | **creatable**

1146 | Value: `<repository-specific>`

1147  
 1148       **fileable**  
 1149           Value: TRUE  
 1150  
 1151       **queryable**  
 1152           Value: SHOULD be TRUE  
 1153  
 1154       **controllablePolicy**  
 1155           Value: <repository-specific>  
 1156  
 1157       **includedInSupertypeQuery**  
 1158           Value: <repository-specific>  
 1159  
 1160       **controllableACL**  
 1161           Value: <repository-specific>  
 1162  
 1163       **fulltextIndexed**  
 1164           Value: <repository-specific>  
 1165

#### 1166 2.1.5.4.2 Property Definitions

1167 The Folder base Object-Type MUST have the following property definitions, and MAY include additional  
 1168 property definitions. Any attributes not specified for the Property Definition are repository specific. For all  
 1169 property definitions on base types, the query name MUST be the same as the property ID. The  
 1170 repository MUST have the following property definitions on the Folder Type:

1171	<b>cmis:name</b>	Name of the object
1172	Inherited:	False
1173	Property Type:	String
1174	Cardinality:	Single
1175	Required:	True
1176		
1177	<b>cmis:objectId</b>	Id of the object
1178	Required:	False
1179	Inherited:	False
1180	Property Type:	ID
1181	Cardinality:	Single
1182	Updatability:	Read Only
1183	Choices:	Not Applicable
1184	Open Choice:	Not Applicable
1185	Repository MUST return this property with non-empty values when an object is requested and the	
1186	property filter does not exclude them	
1187		
1188	<b>cmis:baseTypeid</b>	Id of the base object-type for the object
1189	Required:	False

1190	Inherited:	False
1191	Property Type:	ID
1192	Cardinality:	Single
1193	Updatability:	Read Only
1194	Choices:	Not Applicable
1195	Open Choice:	Not Applicable
1196	Repository MUST return this property with non-empty values when an object is requested and the	
1197	property filter does not exclude them	
1198		
1199	<b>cmis:objectType</b>	Id of the object's type
1200	Required:	<del>False</del> True
1201	Inherited:	False
1202	Property Type:	ID
1203	Cardinality:	Single
1204	Updatability:	oncreate
1205	Choices:	Not Applicable
1206	Open Choice:	Not Applicable
1207	Repository MUST return this property with non-empty values when an object is requested and the	
1208	property filter does not exclude them	
1209		
1210	<b>cmis:createdBy</b>	User who created the object.
1211	Required:	False
1212	Inherited:	False
1213	Property Type:	String
1214	Cardinality:	Single
1215	Updatability:	Read Only
1216	Choices:	Not Applicable
1217	Open Choice:	Not Applicable
1218	Queryable:	True
1219	Orderable:	True
1220	Repository MUST return this property with non-empty values when an object is requested and the	
1221	property filter does not exclude them	
1222		
1223	<b>cmis:creationDate</b>	DateTime when the object was created.
1224	Required:	False
1225	Inherited:	False
1226	Property Type:	DateTime
1227	Cardinality:	Single
1228	Updatability:	Read Only
1229	Choices:	Not Applicable
1230	Open Choice:	Not Applicable
1231	Queryable:	True
1232	Orderable:	True

1233 Repository MUST return this property with non-empty values when an object is requested and the  
1234 property filter does not exclude them

1235

1236 **cmis:lastModifiedBy** User who last modified the object.

1237 Required: False

1238 Inherited: False

1239 Property Type: String

1240 Cardinality: Single

1241 Updatability: Read Only

1242 Choices: Not Applicable

1243 Open Choice: Not Applicable

1244 Queryable: True

1245 Orderable: True

1246 Repository MUST return this property with non-empty values when an object is requested and the  
1247 property filter does not exclude them

1248

1249 **cmis:lastModificationDate** DateTime when the object was last modified.

1250 Required: False

1251 Inherited: False

1252 Property Type: DateTime

1253 Cardinality: Single

1254 Updatability: Read Only

1255 Choices: Not Applicable

1256 Open Choice: Not Applicable

1257 Queryable: True

1258 Orderable: True

1259 MUST be set on the object

1260

1261 **cmis:changeToken** Token used for optimistic locking & concurrency checking.  
1262 (see section 2.2.1.3 Change Tokens)

1263 Required: False

1264 Inherited: False

1265 Property Type: String

1266 Cardinality: Single

1267 Updatability: Read Only

1268 Choices: Not Applicable

1269 Open Choice: Not Applicable

1270 Repository MUST return this property with non-empty values when an object is requested and the  
1271 property filter does not exclude them. If the repository does not support change tokens, this  
1272 property SHOULD not be set.

1273

1274 **cmis:parentId** ID of the parent folder of the folder.

1275 Required: False

1276 Inherited: False

1277	Property Type:	ID
1278	Cardinality:	Single
1279	Updatability:	Read Only
1280	Choices:	Not Applicable
1281	Open Choice:	Not Applicable
1282	Repository MUST return this property with non-empty values when an object is requested and the	
1283	property filter does not exclude them	
1284		
1285	<b>cmis:path</b>	The fully qualified path to this folder. See section 2.1.5.3
1286		Paths.
1287	Required:	False
1288	Inherited:	False
1289	Property Type:	String
1290	Cardinality:	Single
1291	Updatability:	Read Only
1292	Choices:	Not Applicable
1293	Open Choice:	Not Applicable
1294	Repository MUST return this property with non-empty values when an object is requested and the	
1295	property filter does not exclude them	
1296		
1297	<b>cmis:allowedChildObjectTypes</b>	Id's of the set of Object-types that can be created, moved or
1298		filed into this folder.
1299	Required:	False
1300	Inherited:	False
1301	Property Type:	ID
1302	Cardinality:	Multi
1303	Updatability:	Read Only
1304	Choices:	Not Applicable
1305	Open Choice:	Not Applicable

## 1306 2.1.6 Relationship Object

1307 A relationship object is semantically a *dependent* object. A relationship object MUST NOT have a  
1308 content-stream, and MUST NOT be versionable, MAY be queryable, and MUST NOT be fileable,  
1309 although it MAY be controllable.

1310 If a repository does not support relationship objects, the relationship base object-type SHOULD NOT be  
1311 returned by a `cmis:GetTypes` service call.

1312 A **Relationship Object** instantiates an explicit, binary, directional, non-invasive, and typed relationship  
1313 between a **Source Object** and a **Target Object**. The source object and the target object MUST both be  
1314 independent objects, such as a document object, a folder object, or a policy object. Whether a policy  
1315 object is allowed to be the source or target object of a relationship object is repository-specific.

1316 The relationship instantiated by a relationship object is *explicit* since it is explicitly represented by an  
1317 object and is explicitly managed by application.

1318 This relationship is *non-invasive* in the sense that creating or removing this relationship SHOULD NOT  
1319 modify either the source or the target object. That is, it SHOULD NOT require an update capability (or  
1320 permission) on either object; SHOULD NOT affect the versioning state of either object; and SHOULD  
1321 NOT change their `"_Last Modification Date"`.

1322 Explicit relationships can be used to create an arbitrary relationship graph among independent objects.  
1323 Such a relationship graph is only structural in nature. No inheritance or transitive properties are attached  
1324 to a relationship graph.

1325 The notion of a source object and a target object of a relationship is used solely to indicate the direction of  
1326 the relationship. No semantics or implementation bias is implied by this terminology.

1327 The binding of a relationship object to a source document object or to a target document object MAY be  
1328 either version-specific or version-independent. This version sensitivity is repository-specific, and is largely  
1329 transparent to CMIS. An independent object MAY participate in any number of explicit relationships, as  
1330 the source object for some and as the target object for others. Multiple relationships MAY exist between  
1331 the same pair of source and target objects.

1332 Referential integrity, either between the source object and the target object, or between the relationship  
1333 object and the source or target object, is repository-specific. Therefore, creating an explicit relationship  
1334 between two objects MAY impose a constraint on any of the three objects, and removing a relationship or  
1335 deleting either the source or the target object MAY be restricted by such a constraint. If the source or the  
1336 target object of a relationship is deleted, the repository MAY automatically delete the relationship object.

1337 Like all CMIS objects, relationship objects are typed. Typing relationship allows them to be grouped,  
1338 identified, and traversed by type id, and for properties to be defined for individual relationship types.

1339 Additionally, a relationship object-type MAY specify that only Objects of a specific Object-Type can  
1340 participate as the source object or target object for relationship objects of that type. If no such constraints  
1341 are specified, then an independent object of any type MAY be the source or the target of a relationship  
1342 object of that type.

1343 When a relationship object is created, the source object ID and the target object ID MUST reference valid  
1344 non-relationship CMIS objects.

1345 When a relationship object is retrieved, its source object or target object MAY no longer exist, since  
1346 referential integrity MAY not be maintained by a repository.

1347 In addition to object CRUD services, a `"Get Relationships"` service (*getObjectRelationships*) may be  
1348 used to return a set of relationship objects in which a given independent object is identified as the source  
1349 or the target object, according to the binding semantics maintained by the repository (i.e., either a  
1350 version-specific or a version-independent binding as described above).

### 1351 2.1.6.1 Relationship Object-Type Definition

1352 This section describes the definition of the Relationship Object-Type's attribute values and property  
1353 definitions which must be present on Relationship instance objects. All attributes and property definitions  
1354 are listed by their ID.

#### 1355 2.1.6.1.1 Attributes specific to Relationship Object-Types

1356 The following Object **attributes** MUST only apply to Object-Type definitions whose baseld is the  
1357 cmis:relationship Object-Type, in addition to the common attributes specified above:

1358 **allowedSourceTypes** ID (multi-valued)

1359 A list of object-type IDs, indicating that the source object of a relationship object of this type  
1360 MUST only be one of the types listed.

1361 | If this attribute is "not set", then the source object MAY be of any type.

1362

1363 | **allowedTargetTypes** ID (multi-valued)

1364 | A list of object-type IDs, indicating that the target object of a relationship object of this type MUST  
1365 | only be one of the types listed.

1366 | If this attribute is "not set", then the target object MAY be of any type.

### 1367 | 2.1.6.1.2 Attribute Values

1368 | The Relationship Object-Type MUST have the following attribute values.

1369 | Notes:

- 1370 | • A value of <repository-specific> indicates that the value of the property MAY be set to any  
1371 | valid value for the attribute type.
- 1372 | • Unless explicitly stated otherwise, all values specified in the table MUST be followed for the  
1373 | Object-Type definition.

1374

1375 | **id**

1376 | Value: cmis:relationship

1377

1378 | **localName**

1379 | Value: <repository-specific>

1380

1381 | **localNamespace**

1382 | Value: <repository-specific>

1383

1384 | **queryName**

1385 | Value: cmis:relationship

1386

1387 | **displayName**

1388 | Value: <repository-specific>

1389

1390 | **baseId**

1391 | Value: cmis:relationship

1392

1393 | **parentId**

1394 | Value: Not set

1395

1396 | **description**

1397 | Value: <repository-specific>

1398

1399 | **creatable**

1400 | Value: <repository-specific>

1401

1402 | **fileable**

1403 Value: FALSE  
 1404  
 1405 **queryable**  
 1406 Value: <repository-specific>  
 1407  
 1408 **includedInSupertypeQuery**  
 1409 Value: <repository-specific>  
 1410  
 1411 **controllablePolicy**  
 1412 Value: <repository-specific>  
 1413  
 1414 **allowedSourceTypes**  
 1415 Value: <repository-specific>  
 1416  
 1417 **allowedTargetTypes**  
 1418 Value: <repository-specific>  
 1419  
 1420 **controllableACL**  
 1421 Value: <repository-specific>  
 1422  
 1423 **fulltextIndexed**  
 1424 Value: <repository-specific>  
 1425

1426 **2.1.6.1.3 Property Definitions**

1427 The Relationship base Object-Type MUST have the following property definitions, and MAY include  
 1428 additional property definitions. Any attributes not specified by the Property Definitions are repository  
 1429 specific. For all property definitions on base types, the query name MUST be the same as the property  
 1430 ID. The repository MUST have the following property definitions on the Relationship Type:

1431		
1432	<b>cmis:name</b>	Name of the object
1433	Inherited:	False
1434	Property Type:	String
1435	Cardinality:	Single
1436		
1437	<b>cmis:objectId</b>	Id of the object
1438	Required:	False
1439	Inherited:	False
1440	Property Type:	ID
1441	Cardinality:	Single
1442	Updatability:	Read Only
1443	Choices:	Not Applicable
1444	Open Choice:	Not Applicable

1445	Repository MUST return this property with non-empty values when an object is requested and the	
1446	property filter does not exclude them	
1447		
1448	<b>cmis:baseTypeId</b>	Id of the base object-type for the object
1449	Required:	False
1450	Inherited:	False
1451	Property Type:	ID
1452	Cardinality:	Single
1453	Updatability:	Read Only
1454	Choices:	Not Applicable
1455	Open Choice:	Not Applicable
1456	Repository MUST return this property with non-empty values when an object is requested and the	
1457	property filter does not exclude them	
1458		
1459	<b>cmis:objectTypeId</b>	Id of the object's type
1460	Required:	<del>False</del> <u>True</u>
1461	Inherited:	False
1462	Property Type:	ID
1463	Cardinality:	Single
1464	Updatability:	oncreate
1465	Choices:	Not Applicable
1466	Open Choice:	Not Applicable
1467	Repository MUST return this property with non-empty values when an object is requested and the	
1468	property filter does not exclude them	
1469		
1470	<b>cmis:createdBy</b>	User who created the object.
1471	Required:	False
1472	Inherited:	False
1473	Property Type:	String
1474	Cardinality:	Single
1475	Updatability:	Read Only
1476	Choices:	Not Applicable
1477	Open Choice:	Not Applicable
1478	Repository MUST return this property with non-empty values when an object is requested and the	
1479	property filter does not exclude them	
1480		
1481	<b>cmis:creationDate</b>	DateTime when the object was created.
1482	Required:	False
1483	Inherited:	False
1484	Property Type:	DateTime
1485	Cardinality:	Single
1486	Updatability:	Read Only
1487	Choices:	Not Applicable

1488	Open Choice:	Not Applicable
1489	Repository MUST return this property with non-empty values when an object is requested and the	
1490	property filter does not exclude them	
1491		
1492	<b>cmis:lastModifiedBy</b>	User who last modified the object.
1493	Required:	False
1494	Inherited:	False
1495	Property Type:	String
1496	Cardinality:	Single
1497	Updatability:	Read Only
1498	Choices:	Not Applicable
1499	Open Choice:	Not Applicable
1500	Repository MUST return this property with non-empty values when an object is requested and the	
1501	property filter does not exclude them	
1502		
1503	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
1504	Required:	False
1505	Inherited:	False
1506	Property Type:	DateTime
1507	Cardinality:	Single
1508	Updatability:	Read Only
1509	Choices:	Not Applicable
1510	Open Choice:	Not Applicable
1511	Repository MUST return this property with non-empty values when an object is requested and the	
1512	property filter does not exclude them	
1513		
1514	<b>cmis:changeToken</b>	Opaque token used for optimistic locking & concurrency
1515		checking. (see section 2.2.1.3 Change Tokens)
1516	Required:	False
1517	Inherited:	False
1518	Property Type:	String
1519	Cardinality:	Single
1520	Updatability:	Read Only
1521	Choices:	Not Applicable
1522	Open Choice:	Not Applicable
1523	<u>Repository MUST return this property with non-empty values when an object is requested and the</u>	
1524	<u>property filter does not exclude them. If the repository does not support change tokens, this</u>	
1525	<u>property SHOULD not be set.</u>	
1526		
1527	<b>cmis:sourceld</b>	ID of the source object of the relationship.
1528	Required:	True
1529	Inherited:	False
1530	Property Type:	ID
1531	Cardinality:	Single

1532	Choices:	Not Applicable
1533	Open Choice:	Not Applicable
1534		
1535	<b>cmis:targetId</b>	ID of the target object of the relationship.
1536	Required:	True
1537	Inherited:	False
1538	Property Type:	ID
1539	Cardinality:	Single
1540	Choices:	Not Applicable
1541	Open Choice:	Not Applicable

## 1542 2.1.7 Policy Object

1543 A policy object represents an administrative policy that can be enforced by a repository, such as a  
 1544 retention management policy. CMIS 1.0 does not specify what kinds of administrative policies that are  
 1545 specifically supported, nor attempts to model administrative policy of any particular kind. Only a base  
 1546 object-type is specified for policy objects. Each policy object holds the text of an administrative policy as a  
 1547 repository-specific string, which is opaque to CMIS and which may be used to support policies of various  
 1548 kinds. A repository may create subtypes of this base type to support different kinds of administrative  
 1549 policies more specifically. If a repository does not support policy objects, the policy base object-type  
 1550 | SHOULD NOT be returned by a `“Get Types”` service call. This is an extension point for repositories that  
 1551 want to expose other capabilities via CMIS that are not supported directly in CMIS 1.0.

1552 Aside from allowing an application to create and maintain policy objects, CMIS allows an application to  
 1553 | `“apply”` a policy to an object, and to remove an applied policy from an object. An object to which a policy  
 1554 may be applied is called a *controllable* object. A policy MAY be applied to multiple controllable objects.  
 1555 Conversely, a repository MAY allow multiple policies applied to a controllable object. (A repository may,  
 1556 for example, impose constraints such as only one policy of each kind can be applied to an object.)  
 1557 | Whether or not an object is controllable is specified by the object's type definition. Applying a policy to an  
 1558 object is to place the object under the control of that policy (while the object may also be under the control  
 1559 of other policies at the same time), and removing an applied policy from one of its controlled objects is to  
 1560 remove the corresponding control from that object. This control may change the state of the object, may  
 1561 impose certain constraints on service calls operating on this object, or may cause certain management  
 1562 actions to take place. The effect of this control, when this effect takes place, and how this control interacts  
 1563 with other controls, are repository-specific. Only directly/explicitly applied policies are covered by CMIS  
 1564 1.0. Indirectly applying policy to an object, e.g. through inheritance, is outside the scope of CMIS 1.0.

1565 A policy object does not have a content-stream and is not versionable. It may be fileable, queryable or  
 1566 controllable. Policy objects are handled using the basic CRUD services for objects. If a policy is updated,  
 1567 the change may alter the corresponding control on objects that the policy is currently applied to. If a  
 1568 controlled object is deleted, all the policies applied to that object, if there are any, are removed from that  
 1569 object. A policy object that is currently applied to one or more controllable objects CAN NOT be deleted.  
 1570 That is, there is an implicit referential constraint from a controlled object to its controlling policy object(s).  
 1571 | Besides the basic CRUD services, the `“Apply Policy”` (*applyPolicy*) and the `“Remove Policy”`  
 1572 (*removePolicy*) services may be used to apply a policy object to a controllable object and respectively to  
 1573 | remove an applied policy from one of its controlled objects. In addition, the `“Get Applied Policies”`  
 1574 (*getAppliedPolicies*) service may be used to obtain the policy objects that are currently applied to a  
 1575 controllable object.

### 1576 2.1.7.1 Policy Object-Type Definition

1577 | This section describes the definition of the Policy Object-Type's attribute values and property definitions  
 1578 which must be present on Policy instance objects. All attributes and property definitions are listed by their  
 1579 ID.

### 1580 2.1.7.1.1 Attribute Values

1581 The Policy Object-Type MUST have the following attribute values.

1582 Notes:

- 1583 | • A value of <repository-specific> indicates that the value of the property MAY be set to  
1584 | any valid value for the attribute type.
- 1585 | • Unless explicitly stated otherwise, all values specified in the table MUST be followed for  
1586 | the Object-Type definition.

1587

1588 **id**

1589 Value: cmis:policy

1590

1591 **localName**

1592 Value: <repository-specific>

1593

1594 **localNamespace**

1595 Value: <repository-specific>

1596

1597 **queryName**

1598 Value: cmis:policy

1599

1600 **displayName**

1601 Value: <repository-specific>

1602

1603 **baseId**

1604 Value: cmis:policy

1605

1606 **parentId**

1607 Value: Not set

1608

1609 **description**

1610 Value: <repository-specific>

1611

1612 **creatable**

1613 Value: <repository-specific>

1614

1615 **fileable**

1616 Value: <repository-specific>

1617

1618 **queryable**

1619 Value: <repository-specific>

1620

1621 **includedInSupertypeQuery**

1622 Value: <repository-specific>

1623  
 1624       **controllablePolicy**  
 1625           Value: <repository-specific>  
 1626  
 1627       **controllableACL**  
 1628           Value: <repository-specific>  
 1629  
 1630       **fulltextIndexed**  
 1631           Value: <repository-specific>  
 1632

1633   **2.1.7.1.2 Property Definitions**

1634   The Policy base Object-Type MUST have the following property definitions, and MAY include additional  
 1635   property definitions. Any attributes not specified by the Property Definitions are repository specific. For  
 1636   all property definitions on base types, the query name MUST be the same as the property ID. The  
 1637   repository MUST have the following property definitions on the Policy Type:

1638		
1639	<b>cmis:name</b>	Name of the object
1640	Inherited:	False
1641	Property Type:	String
1642	Cardinality:	Single
1643		
1644	<b>cmis:objectId</b>	Id of the object
1645	Required:	False
1646	Inherited:	False
1647	Property Type:	ID
1648	Cardinality:	Single
1649	Updatability:	Read Only
1650	Choices:	Not Applicable
1651	Open Choice:	Not Applicable
1652		
1653	<b>cmis:baseTypeid</b>	Id of the base object-type for the object
1654	Required:	False
1655	Inherited:	False
1656	Property Type:	ID
1657	Cardinality:	Single
1658	Updatability:	Read Only
1659	Choices:	Not Applicable
1660	Open Choice:	Not Applicable
1661		
1662	<b>cmis:objectTypeid</b>	Id of the object's type
1663	Required:	<del>False</del> True
1664	Inherited:	False

1665	Property Type:	ID
1666	Cardinality:	Single
1667	Updatability:	oncreate
1668	Choices:	Not Applicable
1669	Open Choice:	Not Applicable
1670		
1671	<b>cmis:createdBy</b>	User who created the object.
1672	Required:	False
1673	Inherited:	False
1674	Property Type:	String
1675	Cardinality:	Single
1676	Updatability:	Read Only
1677	Choices:	Not Applicable
1678	Open Choice:	Not Applicable
1679		
1680	<b>cmis:creationDate</b>	DateTime when the object was created.
1681	Required:	False
1682	Inherited:	False
1683	Property Type:	DateTime
1684	Cardinality:	Single
1685	Updatability:	Read Only
1686	Choices:	Not Applicable
1687	Open Choice:	Not Applicable
1688		
1689	<b>cmis:lastModifiedBy</b>	User who last modified the object.
1690	Required:	False
1691	Inherited:	False
1692	Property Type:	String
1693	Cardinality:	Single
1694	Updatability:	Read Only
1695	Choices:	Not Applicable
1696	Open Choice:	Not Applicable
1697		
1698	<b>cmis:lastModificationDate</b>	DateTime when the object was last modified.
1699	Required:	False
1700	Inherited:	False
1701	Property Type:	DateTime
1702	Cardinality:	Single
1703	Updatability:	Read Only
1704	Choices:	Not Applicable
1705	Open Choice:	Not Applicable
1706		

1707	<b>cmis:changeToken</b>	Opaque token used for optimistic locking & concurrency checking. (see section 2.2.1.3 Change Tokens)
1708		
1709	Required:	False
1710	Inherited:	False
1711	Property Type:	String
1712	Cardinality:	Single
1713	Updatability:	Read Only
1714	Choices:	Not Applicable
1715	Open Choice:	Not Applicable
1716		<u>Repository MUST return this property with non-empty values when an object is requested and the property filter does not exclude them. If the repository does not support change tokens, this property SHOULD not be set.</u>
1717		
1718		
1719		
1720	<b>cmis:policyText</b>	User-friendly description of the policy
1721	Required:	True
1722	Inherited:	False
1723	Property Type:	String
1724	Cardinality:	Single
1725	Choices:	Not Applicable
1726	Open Choice:	Not Applicable

## 1727 2.1.8 Access Control

1728 A repository can support either a base set of CMIS-defined permissions and/or its own set of repository  
1729 specific permissions.

1730 The getACL service allows the requestor to specify that the result be expressed using only the CMIS  
1731 defined permissions. Without this restriction, the response may include, or be solely expressed in  
1732 repository specific permissions. The applyACL service permits either CMIS permissions or repository  
1733 permissions, or a combination of both, to be used.

### 1734 2.1.8.1 ACL, ACE, Principal, and Permission

1735 An **ACL** is a list of **Access Control Entries** (ACEs) and MAY hold zero or more ACEs. If an ACL has no  
1736 ACEs, the behavior is the same as if the ACL is not set.

1737 An **ACE** holds:

- 1738 • one **Principal**: A principal represents a user management object, e.g. a user, group, or role.  
1739 It holds one **String** with the **principalid**.
- 1740 • One or more **Strings** with the names of the **permissions**.
- 1741 • a **Boolean** flag **direct**, which indicates if TRUE the ACE is directly assigned to the object. If  
1742 FALSE, that the ACE is somehow derived.

### 1743 2.1.8.2 CMIS Permissions

1744 There are three basic permissions predefined by CMIS:

- 1745 • **cmis:read**: to be used to express "permission to read". A Repository SHOULD express  
1746 the permission for reading properties AND reading content with this permission.
- 1747 • **cmis:write**: to be used to express "permission to write". SHOULD be used to express  
1748 permission to write properties and content of an object. MAY include other basic CMIS  
1749 permissions.

- 1750 |                   • **cmis:all**: SHOULD be used to express all the permissions of a repository. SHOULD  
1751 | include all other basic CMIS permissions.

1752 | How these basic permissions can be mapped to the allowable actions is repository specific. However, the  
1753 | actual repository semantics for the basic permissions with regard to allowable actions can be discovered  
1754 | by the *mappings* parameter returned by *getRepositoryInfo* (see below).

1755 | Repositories MAY extend this set with repository-specific permissions.

### 1756 | 2.1.8.3 ACL Capabilities

1757 | Whether a repository supports ACLs at all, may be discovered via *capabilityACL* returned by  
1758 | *getRepositoryInfo* (see section 2.1.1.1 Optional Capabilities). If *capabilityACL* is `none`, ACLs are not  
1759 | supported by the repository.

1760 | If *capabilityACL* is `discover` or `manage`, additional information about the repositories permission model  
1761 | and how changes to ACL are handled, can be discovered via the *getRepositoryInfo* service:

- 1762 |                   • **<Array> Enum propagation**: specifies, how non-direct ACEs can be handled by the  
1763 | repository using the following values (see section 2.2.10.2 applyACL):
- 1764 |                   ○ **objectonly** indicates, that the repository is able to apply ACEs to a document or folder,  
1765 | without changing the ACLs of other objects.
  - 1766 |                   ○ **propagate**: indicates that the ACEs is to be applied to the given object and all inheriting  
1767 | objects. *Propagate incorporates the support for objectonly.*
  - 1768 |                   ○ **repositorydetermined** indicates, that the repository has its own mechanism of  
1769 | computing how changing an ACL for an object influences the non-direct ACEs of other  
1770 | objects.
- 1771 |                   • **<Array> PermissionDefinition repositoryPermissions**: is a list with names and  
1772 | descriptions of the supported permissions.
- 1773 |                   • **<Array> PermissionMapping mappings**: contains a list with mappings for the basic CMIS  
1774 | permissions to allowed actions.

#### 1775 | 2.1.8.3.1 Supported Permissions

1776 | The list of permission definitions returned by *getRepositoryInfo* lists all the *permissions* a repository  
1777 | supports. This list also includes the CMIS permissions if supported by the repository.

1778 | A *PermissionDefinition* holds:

- 1779 |                   • **String permission**: the (technical) name of the permission (unique within the list of permission  
1780 | definitions).
- 1781 |                   • **(Optional) String description**: an optional description of the permission that should be used as  
1782 | the permission's name to be presented to the user.

#### 1783 | 2.1.8.3.2 AllowableActions & Permission Mapping

1784 | CMIS provides a mechanism called `AllowableActions` which allows an application to discover the set of  
1785 | service operations that can currently be performed on a particular object, without having to actually invoke  
1786 | the service.

1787 | The set of allowable actions on an object at a point in time are affected not only by CMIS ACLs, but also  
1788 | by other factors such as:

- 1789 |                   • 1. Constraints inherent in the CMIS Domain Model based on the object's base type or current  
1790 | versioning state.
- 1791 |                   • 2. Policies or other control mechanisms that are opaque to CMIS.
- 1792 |

1793 CMIS defines several services that applications can use at run-time to discover the AllowableActions for  
1794 an object.

1795 If a Repository supports ACLs, then the repository MUST provide a mapping table that defines how the  
1796 permissions supported by the repository interact with the CMIS allowable actions, i.e. which permissions  
1797 are necessary for a principal to have on one or more objects in order to potentially perform each action,  
1798 subject to the other constraints on allowable actions above.

1799 This section defines both the allowable actions as well as how those actions are presented in the  
1800 PermissionMapping table.

1801 The Permission Mapping table contains a set of (*key, permissions*) pairs:

1802 |     • **String Key:** Because several allowable actions may require permissions on more than one  
1803 |     object – for example, moving a document from one folder to another may require permissions  
1804 |     on the document and each of the folders – the mapping table is defined in terms of  
1805 |     permission “\_keys\_”, where each key combines the name of the allowable action as the  
1806 |     object for which the principal needs the required permission.

1807 |     ○ For example – the canMoveObject.Source key indicates the permissions that the  
1808 |     principal must have on the “\_source folder\_” to move an object from that folder into  
1809 |     another folder.

1810 |     • **<Array> String permissions:** The names of one or more permissions that the principal MUST  
1811 |     have. If more than one permission is specified, then the principal MUST be allowed to  
1812 |     perform the operation if they have ANY of the listed permissions.

1813 The list below defines all mapping keys, as well as a permissions mapping that repositories SHOULD  
1814 use. Repositories MAY require additional permissions.

1815 For convenience, the list below groups all mapping entries by the underlying Allowable Actions, and  
1816 includes descriptive information. For each Allowable Action the following information is given:

1817	Description:	The description and name of the service the AllowableAction enables.
1818	Base Object:	The base object-types for which the allowable action MAY be TRUE.
1819	Operand:	The object the permission applies to.
1820	Key:	The permission mapping key.
1821	Permissions:	The permission values.

1822

### 1823 **Navigation Services:**

#### 1824 **canGetDescendants**

1825	Description:	Can get the descendants of the folder ( <i>getDescendants</i> )
1826	Base Object:	cmis:folder
1827	Operand:	cmis:folder
1828	Key:	canGetDescendants.Folder
1829	Permission:	Read

1830

#### 1831 **canGetFolderTree**

1832	Description:	Can get the sub-folder tree of the folder ( <i>getFolderTree</i> )
1833	Base Object:	cmis:folder
1834	Operand:	cmis:folder
1835	Key:	canGetFolderTree.Folder
1836	Permission:	Read

1837

#### 1838 **canGetChildren**

1839	Description:	Can get the children of the folder ( <code>getChildren</code> )
1840	Base Object:	<code>cmis:folder</code>
1841	Operand:	<code>cmis:folder</code>
1842	Key:	<code>canGetChildren.Folder</code>
1843	Permission:	Read
1844		
1845	<b>canGetFolderParent</b>	
1846	Description:	Can get the parent/ancestor folder(s) of the folder ( <code>getFolderParent</code> )
1847	Base Object:	<code>cmis:folder</code>
1848	Operand:	<code>cmis:folder</code>
1849	Key:	<code>canGetFolderParent.FolderObject</code>
1850	Permission:	Read
1851		
1852	<b>canGetObjectParents</b>	
1853	Description:	Can get the parent folders of the object. ( <code>getObjectParents</code> )
1854	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code> , <code>cmis:policy</code>
1855	Operand	Object
1856	Key:	<code>canGetObjectParents.Object</code>
1857	Permission:	Read
1858		
1859	<b>Object Services:</b>	
1860	<b>canCreateDocument</b>	
1861	Description:	Can create a <code>cmis:document</code> Object in the folder ( <code>createDocument</code> )
1862	Base Object:	<code>cmis:folder</code>
1863	Operand:	Folder
1864	Key:	<code>canCreateDocument.Folder</code>
1865	Permission:	Read
1866		
1867	<b>canCreateFolder</b>	
1868	Description:	Can create a <code>cmis:folder</code> Object as a child of the specified folder
1869	( <code>createFolder</code> )	
1870	Base Object:	<code>cmis:folder</code>
1871	Operand:	Folder
1872	Key:	<code>canCreateFolder.Folder</code>
1873	Permission:	Read
1874		
1875	<b>canCreateRelationship</b>	
1876	Description:	Can create a Relationship in which this Object is a source
1877	( <code>createRelationship</code> )	
1878	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code>
1879	Operand:	Object
1880	Key:	<code>canCreateRelationship.Source</code>
1881	Permission:	Read

1882		
1883	<b>canCreateRelationship</b>	
1884	Description:	Can create a Relationship in which this Object is a target
1885		(createRelationship)
1886	Base Object:	cmis:document, cmis:folder
1887	Operand:	Object
1888	Key:	canCreateRelationship.Target
1889	Permission:	Read
1890		
1891	<b>canGetProperties</b>	
1892	Description:	Can read the properties of this object (getProperties)
1893	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1894	Operand:	Object
1895	Key:	canGetProperties.Object
1896	Permission:	Read
1897		
1898	<del><b>canGetRenditions</b></del>	
1899	<del>Description:</del>	<del>Can retrieve the renditions of this object (getRenditions)</del>
1900	<del>Base Object:</del>	<del>cmis:document, or cmis:folder</del>
1901	<del>Operand:</del>	<del>Object</del>
1902	<del>Key:</del>	<del>canGetRenditions.Object</del>
1903	<del>Permission:</del>	<del>Read</del>
1904		
1905	<del><b>canGetContentStream</b></del>	
1906	<del>Description:</del>	<del>Can get the content stream for the Document object</del>
1907		<del>(getContentStream)</del>
1908	<del>Base Object:</del>	<del>cmis:document</del>
1909	<del>Operand:</del>	<del>Object</del>
1910	<del>Key:</del>	<del>canGetContentStream.Object</del>
1911	<del>Permission:</del>	<del>Read</del>
1912		
1913	<b>canUpdateProperties</b>	
1914	Description:	Can update the properties of this object (updateProperties)
1915	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1916	Operand:	Object
1917	Key:	canUpdateProperties.Object
1918	Permission:	Write
1919		
1920	<b>canMoveObject</b>	
1921	Description:	Can move the object (moveObject)
1922	Base Object:	cmis:document, cmis:folder, cmis:policy
1923	Operand:	Object
1924	Key:	canMoveObject.Object

1925	Permission:	Write
1926		
1927	<b>canMoveObject</b>	
1928	Description:	Can move an object into this folder ( <code>moveObject</code> )
1929	Base Object:	cmis:folder
1930	Operand:	Folder
1931	Key:	canMoveObject.Target
1932	Permission:	Read
1933		
1934	<b>canMoveObject</b>	
1935	Description:	Can move an object from this folder ( <code>moveObject</code> )
1936	Base Object:	cmis:folder
1937	Operand:	Folder
1938	Key:	canMoveObject.Source
1939	Permission:	Read
1940		
1941	<b>canDeleteObject</b>	
1942	Description:	Can delete this object ( <code>deleteObject</code> )
1943	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
1944	Operand:	Object
1945	Key:	canDelete.Object
1946	Permission:	Write
1947		
1948		
1949	<del><b>canGetContentStreamcanDeleteObject</b></del>	
1950	<del>Description:</del>	<del>Can delete an object that is a child of this folder (<code>deleteObject</code>)</del>
1951		
1952	Base Object:	cmis: <del>folder</del> document
1953	<del>Action:</del>	<del>Can get the content stream for the Document object</del>
1954		<del>(<code>getContentStream</code>)</del>
1955	Operand:	<del>Folder</del> Object
1956	Key:	<del>canDelete.Folder</del> canViewContent.Object
1957	Permission:	Read
1958		
1959	<b>canSetContentStream</b>	
1960	Description:	Can set the content stream for the Document object
1961		( <code>setContentStream</code> )
1962	Base Object:	cmis:document
1963	Operand:	Object
1964	Key:	canSetContentStream.Document
1965	Permission:	Write
1966		
1967	<b>canDeleteContentStream</b>	

1968	Base Object:	cmis:document
1969	Action:	Can delete the content stream for the Document object (deleteContentStream)
1970		
1971	Operand:	Object
1972	Key:	canDeleteContentStream.Document
1973	Permission:	Write
1974		
1975	<b>canDeleteTree</b>	
1976	Base Object:	cmis:folder
1977	Action:	Can delete the folder and all contained objects (deleteTree)
1978	Operand:	Object
1979	Key:	canDeleteTree.Folder
1980	Permission:	Write
1981		
1982	<b>Filing Services:</b>	
1983	<b>canAddObjectToFolder</b>	
1984	Description:	Can file the document in a folder (addObjectToFolder)
1985	Base Object:	cmis:document, cmis:policy
1986	Operand:	Object
1987	Key:	canAddToFolder.Object
1988	Permission:	Read
1989		
1990	<b>canAddObjectToFolder</b>	
1991	Description:	Can file a document in the specified folder (addObjectToFolder)
1992	Base Object:	cmis:document, cmis:policy
1993	Operand:	Object
1994	Key:	canAddToFolder.Folder
1995	Permission:	Read
1996		
1997	<b>canRemoveObjectFromFolder</b>	
1998	Description:	Can unfile the specified document from a folder
1999		(removeObjectFromFolder)
2000	Base Object:	cmis:document, cmis:policy
2001	Operand:	Object
2002	Key:	canRemoveObjectFromFolder.Object
2003	Permission:	Read
2004		
2005	<b>canRemoveObjectFromFolder</b>	
2006	Description:	Can unfile a document from the specified folder
2007		(removeObjectFromFolder)
2008	Base Object:	cmis:document, cmis:policy
2009	Operand:	Object

2010	Key:	canRemoveObjectFromFolder.Folder
2011	Permission:	Read
2012		
2013	<b>Versioning Services:</b>	
2014	<b>canCheckOut</b>	
2015	Description:	Can check out the Document object ( <code>checkOut</code> )
2016	Base Object:	cmis:document
2017	Operand:	Object
2018	Key:	canCheckOut.Document
2019	Permission:	Write
2020		
2021	<b>canCancelCheckOut</b>	
2022	Description:	Can cancel the check out the Document object ( <code>cancelCheckOut</code> )
2023	Base Object:	cmis:document
2024	Operand:	Object
2025	Key:	canCancelCheckout.Document
2026	Permission:	Write
2027		
2028	<b>canCheckIn</b>	
2029	Description:	Can check in the Document object ( <code>checkIn</code> )
2030	Base Object:	cmis:document
2031	Operand:	Object
2032	Key:	canCheckin.Document
2033	Permission:	Write
2034		
2035	<b>canGetAllVersions</b>	
2036	Description:	Can get the version series for the Document object ( <code>getAllVersions</code> )
2037	Base Object:	cmis:document
2038	Operand:	Object
2039	Key:	canGetAllVersions. <u>DocumentVersionSeries</u>
2040	Permission:	Read
2041		
2042	<b>Relationship Services:</b>	
2043	<b>canGetObjectRelationships</b>	
2044	Description:	Can get the relationship in which this object is a source/target
2045		( <code>getObjectRelationships</code> )
2046	Base Object:	cmis:document, cmis:folder, cmis:policy
2047	Operand:	Object
2048	Key:	canGetObjectRelationships.Object
2049	Permission:	Read

2050		
2051	<b>Policy Services:</b>	
2052	<b>canApplyPolicy</b>	
2053	Description:	Can apply a policy to the Object ( <code>applyPolicy</code> )
2054	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code>
2055	Operand:	Object
2056	Key:	<code>canAddPolicy.Object</code>
2057	Permission:	Read
2058		
2059	<b>canApplyPolicy</b>	
2060	Description:	Can apply the specified policy to an Object ( <code>applyPolicy</code> )
2061	Base Object:	<code>cmis:policy</code>
2062	Operand:	Object
2063	Key:	<code>canAddPolicy.Policy</code>
2064	Permission:	Read
2065		
2066	<b>canRemovePolicy</b>	
2067	Description:	Can remove a policy from the specified Object ( <code>removePolicy</code> )
2068	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code>
2069	Operand:	Object
2070	Key:	<code>canRemovePolicy.Object</code>
2071	Permission:	Read
2072		
2073	<b>canRemovePolicy</b>	
2074	Description:	Can remove the specified policy from an Object ( <code>removePolicy</code> )
2075	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code>
2076	Operand:	<code>cmis:policy</code>
2077	Key:	<code>canRemovePolicy.Policy</code>
2078	Permission:	Read
2079		
2080	<b>canGetAppliedPolicies</b>	
2081	Description:	Can get the list of Policies applied to the Object
2082		( <code>getAppliedPolicies</code> )
2083	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code>
2084	Operand:	Object
2085	Key:	<code>canGetAppliedPolicies.Object</code>
2086	Permission:	Read
2087		
2088	<b>ACL Services:</b>	
2089	<b>canGetACL</b>	
2090	Description:	Can get ACL for object ( <code>getACL</code> )
2091	Base Object:	<code>cmis:document</code> , <code>cmis:folder</code> , <code>cmis:relationship</code> , <code>cmis:policy</code>

2092	Operand:	Object
2093	Key:	canGetACL.Object
2094	Permission:	Read
2095		
2096	<b>canApplyACL</b>	
2097	Description:	Can apply ACL to this object ( <code>applyACL</code> )
2098	Base Object:	cmis:document, cmis:folder, cmis:relationship, cmis:policy
2099	Operand:	Object
2100	Key:	canApplyACL.Object
2101	Permission:	Write
2102		

2103

## 2104 2.1.9 Versioning

2105 CMIS supports versioning of Document objects. Folder objects, relationship objects, and policy objects  
2106 cannot be versioned.

2107 Whether or not a Document object is versionable (i.e. whether or not operations performed on the object  
2108 | via the Versioning Services MUST be allowed) is specified by the `“_versionable”` attribute on its Object-  
2109 type.

2110 | A **version** of a Document object is an explicit `“/”` `“deep”` copy of the object, preserving its state at a certain  
2111 point in time. Each version of a Document object is itself a Document object, i.e. has its own *ObjectId*,  
2112 property values, MAY be acted upon using all CMIS services that act upon Document objects, etc.

### 2113 2.1.9.1 Version Series

2114 A **version series** for a Document object is a transitively closed collection of all Document objects that  
2115 have been created from an original Document in the Repository. Each version series has a unique,  
2116 system-assigned, and immutable **version series ID**.

2117 The version series has transitive closure -- that is, if object B is a version of object A, and object C is a  
2118 version of object B, then object C is also a version of object A. The objects in a version series can be  
2119 conceptually sequenced by their respective *CreationDate* properties.

2120 Additionally, the repository MAY expose a textual **VersionLabel** that describes to a user the position of  
2121 an individual object with respect to the version series. (For example, version 1.0).

2122 *Note:* A Document object that is NOT versionable will always have a single object in its Version Series. A  
2123 versionable Document object MAY have one or more objects in its Version Series.

### 2124 2.1.9.2 Latest Version

2125 The version that has the most recent *LastModificationDate* is called the **Latest Version** of the series, or  
2126 equivalently, the latest version of any Document object in the series.

2127 When the latest version of a version series is deleted, a previous version (if there is one) becomes the  
2128 latest version.

#### 2129 2.1.9.2.1 Behavioral constraints on non-Latest Versions

2130 Repositories NEED NOT allow the non-latest versions in a Version Series to be updated, queried, or  
2131 searched.

### 2132 2.1.9.3 Major Versions

2133 A Document object in a Version Series MAY be designated as a **Major Version**.

2134 The CMIS specification does not define any semantic/behavioral differences between Major and non-  
2135 Major versions in a Version Series. Repositories may enforce/apply additional constraints or semantics for  
2136 Major versions, if the effect on CMIS services remains consistent with an allowable behavior of the CMIS  
2137 model.

2138 If the Version Series contains one or more Major versions, the one that has the most recent  
2139 *LastModificationDate* is the **Latest Major Version** of the version series.

2140 (Note that while a Version Series MUST always have a *Latest Version*, it NEED NOT have a *Latest Major*  
2141 *Version*.)

2142 When the latest major version is deleted, a previous major version (if there is one) becomes the latest  
2143 major version.

## 2144 2.1.9.4 Services that modify Version Series

### 2145 2.1.9.4.1 Checkout

2146 | A new version of a versionable Document object is created when the ~~checkIn~~*checkIn* service is invoked  
2147 | on the Private Working copy (PWC) of this object. A PWC is created by invoking *checkOut* on a  
2148 | versionable Document object. A repository MAY allow *any* Document object in a version series to be  
2149 | checked out, or MAY only allow the *Latest Version* to be checked out.

2150 | The effects of invoking the *checkout* service MUST be as follows:

- 2151 | • A new Document object, referred to herein as the **Private Working Copy (PWC)**, is created.
  - 2152 | ○ The PWC NEED NOT be visible to users who have permissions to view other Document
  - 2153 | objects in the Version Series.
  - 2154 | ○ Until it is checked in (using the *checkIn* service), the PWC MUST NOT be considered the
  - 2155 | *LatestMajorVersion* in the Version Series.
  - 2156 | ○ The property values for the PWC SHOULD be identical to the properties of the Document
  - 2157 | object on which the *checkout* service was invoked. Certain properties such as *cmis:objectId*
  - 2158 | may be different. Properties such as *cmis:creationDate* most likely will be different. The
  - 2159 | content-stream of the PWC MAY be identical to the content-stream of the Document object
  - 2160 | on which the *checkout* service was invoked, or MAY be `"not set"`.

2161 | After a successful *checkout* operation is completed, and until such time when the PWC is deleted (via the  
2162 | *cancelCheckOut* service) or checked-in (via the *checkIn*) service, the effects on other Documents in the  
2163 | Version Series MUST be as follows:

- 2164 | • The repository MUST throw an exception if the *checkout* service is invoked on any Document in  
2165 | the Version Series. (I.e. there can only be one PWC for a version series at a time.)
- 2166 | • The value of the *cmis:isVersionSeriesCheckedOut* property MUST be TRUE.
- 2167 | • The value of the *cmis:versionSeriesCheckedOutBy* property MAY be set to a value indicating  
2168 | which user created the PWC. (The Repository MAY still show the `"not set"` value for this  
2169 | property.)
- 2170 | • The value of the *cmis:versionSeriesCheckedOutId* property MAY be set to the *ObjectId* of the  
2171 | PWC. (The Repository MAY still show the `"not set"` value for this property).
- 2172 | • The repository MAY prevent operations that modify or delete the other Documents in the Version  
2173 | Series.

### 2174 2.1.9.4.2 Updates to the Private Working Copy

2175 | If the repository supports the optional `"PWCUpdatable"` capability, then the repository MUST allow  
2176 | authorized users to modify the PWC Object using the Object services (e.g. *UpdateProperties*).

2177 | If the repository does NOT support the `"PWCUpdatable"` capability, then the PWC object can only be  
2178 | modified as part of the *checkIn* service call.

### 2179 2.1.9.4.3 Discarding Check out

2180 | An authorized user MAY discard the check-out using the *cancelCheckOut* service on any Document in  
2181 | the Version Series or by using the *deleteObject* service on the PWC Object. The effects of discarding a  
2182 | check-out MUST be as follows:

- 2183 | • The PWC Object MUST be deleted.
- 2184 | • For all other Documents in the Version Series:
  - 2185 | ○ The value of the *cmis:isVersionSeriesCheckedOut* property MUST be FALSE.
  - 2186 | ○ The value of the *cmis:versionSeriesCheckedOutBy* property MUST be `"not set"`.
  - 2187 | ○ The value of the *cmis:versionSeriesCheckedOutId* property MUST be `"not set"`.
  - 2188 | ○ The repository MUST allow authorized users to invoke the *checkout* service.

2189 **2.1.9.4.4 Checkin**

2190 | An authorized user/application MAY “check in” the Private Working Copy object via the *checkIn* service.  
2191 | The *checkIn* service allows users/applications to provide update property values and a content-stream for  
2192 | the PWC object.

2193 | The effects of the checkIn service MUST be as follows for successful checkins:

- 2194 | • The PWC object MUST be updated as specified by the inputs to the *checkIn* service. (Note that for  
2195 | repositories that do NOT support the “PWCUpdatable” property, this is the only way to update the  
2196 | PWC object.)
- 2197 | • The Document object resulting from the *checkIn* operation MUST be considered the *Latest Version*  
2198 | in the Version Series.
- 2199 | • If the inputs to the *checkIn* service specified that the PWC MUST be a “major version”, then the  
2200 | PWC MUST be considered the *Latest Major Version* in the Version Series.
- 2201 | • If the checkin returns a new cmis:objected, then the PWC object MUST disappear if the *checkIn* call  
2202 | was successful and the new checked in version will use the new specified id.
- 2203 | • For all Documents in the Version Series:
  - 2204 | ○ The value of the cmis:isVersionSeriesCheckedOut property MUST be FALSE.
  - 2205 | ○ The value of the cmis:versionSeriesCheckedOutBy property MUST be “not set”.
  - 2206 | ○ The value of the cmis:versionSeriesCheckedOutId property MUST be “not set”.
  - 2207 | ○ The repository MUST allow authorized users to invoke the *checkout* service.

2208 | *Note:* The Repository MAY change the ID of the PWC upon completion of the *checkin* service invocation.

2209 | *Note:* A repository MAY automatically create new versions of Document objects without an explicit  
2210 | invocation of the checkout/checkin services.

2211 **2.1.9.5 Versioning Properties on Document Objects**

2212 | All Document objects will have the following read-only property values pertaining to versioning:

- 2213
- 2214 | **cmis:isLatestVersion** Boolean  
2215 | TRUE if the Document object is the *Latest Version* in its *Version Series*. FALSE otherwise.  
2216
- 2217 | **cmis:isMajorVersion** Boolean  
2218 | TRUE if the Document object is a *Major Version* in its *Version Series*. FALSE otherwise.  
2219
- 2220 | **cmis:isLatestMajorVersion** Boolean  
2221 | TRUE if the Document object is the *Latest Major Version* in its *Version Series*. FALSE otherwise.  
2222
- 2223 | **cmis:versionLabel** String (optional)  
2224 | Optional textual description the position of an individual object with respect to the version series.  
2225 | (For example, version 1.0).  
2226
- 2227 | **cmis:versionSeriesId** ID  
2228 | ID of the Version Series for this Object.  
2229
- 2230 | **cmis:isVersionSeriesCheckedOut** Boolean  
2231 | TRUE if there currently exists a Private Working Copy for this Version Series. FALSE otherwise

2232

2233 **cmis:versionSeriesCheckedOutBy** String

2234 If IsVersionSeriesCheckedOut is TRUE: then an identifier for the user who created the Private

2235 Working Copy. **"Not set"** otherwise.

2236

2237 **cmis:versionSeriesCheckedOutId** ID

2238 If IsVersionSeriesCheckedOut is TRUE: The Identifier for the Private Working Copy. **"Not set"**

2239 otherwise.

2240

2241 **cmis:checkinComment** String

2242 Textual comment associated with the given version.

2243 *Note:* Changes made via the Versioning Services that affect the values of these properties MUST NOT

2244 constitute modifications to the Document objects in the Version Series (e.g. MUST NOT affect the

2245 cmis:lastModificationDate, etc.)

### 2246 **2.1.9.6 Document Creation and Initial Versioning State**

2247 A repository MAY create new Document objects in a **"Private Working Copy"** state when they are

2248 created via the *createDocument* or *createDocumentFromSource* services. This state is logically

2249 equivalent to having a Version Series that contains exactly one object (the PWC) and 0 other documents.

2250 The repository MAY also create new Document objects in a **"Major Version"** state. This state is logically

2251 equivalent to having a Version Series that contains exactly one Major Version and 0 other documents.

2252 The repository MAY also create new Document objects in a **"Non-Major Version"** state. This state is

2253 logically equivalent to having a Version Series that contains exactly one Non-Major Version and 0 other

2254 documents.

2255 If the repository does not support versioning the repository MUST ignore the value of the versioningState

2256 parameter.

### 2257 **2.1.9.7 Version Specific/Independent membership in Folders**

2258 Repositories MAY treat membership of a Document object in a folder collection as **"version-specific"** or

2259 **"version-independent"**.

2260 Repositories MUST indicate whether they support version-specific membership in a folder via the

2261 **"VersionSpecificFiling"** optional capability flag.

2262 If the repository is treating folder collection membership as **"version-independent"**, then:

- 2263 • Moving or Filing a Document Object into a folder MUST result in ALL Documents in the Version
- 2264 Series being moved/into the folder.
- 2265 • The Repository MAY return only the latest-version OR latest major-version Document object in a
- 2266 version series in the response to Navigation service requests (getChildren, getDescendants), and
- 2267 NEED NOT return other Document Objects filed in the folder that are in the Version Series.

2268 If the repository is treating folder collection membership as **"version-specific"**, then moving or Filing a

2269 Document Object into a folder MUST NOT result in other Documents in the Version Series being

2270 moved/into the folder.

### 2271 **2.1.9.8 Version Specific/Independent membership in Relationships**

2272 A relationship object MAY have either a version-specific or version-independent binding to its source

2273 and/or target objects. This behavior MAY vary between repositories and between individual relationship

2274 types defined for a Repository.

2275 If a relationship object has a version-independent binding to its source/target object, then:

2276 | • The getObjectRelationships service invoked on a Document Object MUST return the relationship  
2277 | if Relationship was source/target is set to ANY Document Object in the Version Series.

2278 | If a relationship object has a version-specific binding to its source/target object, then:

2279 | • The getObjectRelationships service invoked on a Document Object MUST return the relationship  
2280 | if Relationship was source/target is set to the ID of the Document Object on which the service was  
2281 | invoked.

### 2282 | **2.1.9.9 Versioning visibility in Query Services**

2283 | Repositories MAY include non-latest-versions of Document Objects in results to the Discovery Services  
2284 | (*query*).

2285 | Repositories MUST indicate whether they support querying for non-latest-versions via the  
2286 | `"_AllVersionsSearchable"` optional capability flag.

2287 | If `"_AllVersionsSearchable"` is TRUE then the Repository MUST include in the query results ANY  
2288 | Document Object in the Version Series that matches the query criteria. (subject to other query constraints  
2289 | such as security.)

2290 | Additionally, repositories MAY include Private Working Copy objects in results in results to the Discovery  
2291 | Services (*query*).

2292 | Repositories MUST indicate whether they support querying for Private Working Copy objects via the  
2293 | `"_PWCSearchable"` optional capability flag.

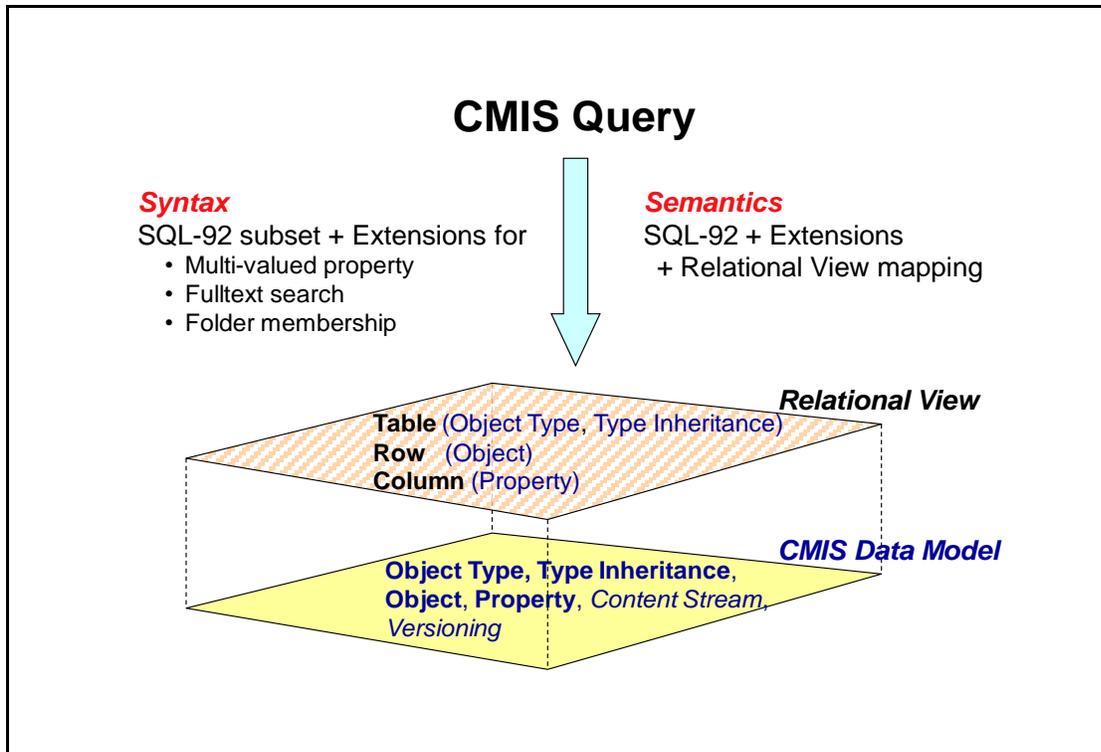
2294 | If `"_PWCSearchable"` is TRUE then the Repository MUST include in the query results ANY Private  
2295 | Working Copy Document Objects that matches the query criteria (subject to other query constraints such  
2296 | as security.)

2297 | If `"_PWCSearchable"` is FALSE then the Repository MUST NOT include in the query results ANY Private  
2298 | Working Copy Document Objects that match the query criteria (subject to other query constraints such as  
2299 | security.)

### 2300 | **2.1.10 Query**

2301 | CMIS provides a type-based query service for discovering objects that match specified criteria, by  
2302 | defining a read-only projection of the CMIS data model into a *Relational View*.

2303 | Through this relational view, queries may be performed via a simplified SQL SELECT statement. This  
2304 | query language is based on a subset of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database  
2305 | Language SQL), with a few extensions to enhance its filtering capability for the CMIS data model, such as  
2306 | existential quantification for multi-valued property, full-text search, and folder membership. Other  
2307 | statements of the SQL language are not adopted by CMIS. The semantics of this query language is  
2308 | defined by the SQL-92 standard, plus the extensions, in conjunction with the model mapping defined by  
2309 | CMIS's relational view.



2310

### 2311 2.1.10.1 Relational View Projection of the CMIS Data Model

2312 The relational view of a CMIS repository consists of a collection of virtual tables that are defined on top of  
2313 the CMIS data model. This relational view is used for query purposes only.

2314 In this relational view a **Virtual Table** is implicitly defined for each *queryable* Object-Type  
2315 defined in the repository. (Non-queryable Object-Types are NOT exposed through this Relational View.)

2316 In each **Virtual Table**, a **Virtual Column** is implicitly defined for each property defined in the Object-Type  
2317 Definition AND for all properties defined on ANY ancestor-type of the Object-Type but NOT defined in the  
2318 Object-Type definition. Virtual Columns for properties defined on ancestor-types of the Object-type but  
2319 NOT defined in the Object-Type definition MUST contain the SQL NULL value. Virtual Columns for  
2320 properties whose value is `"not set"` MUST contain the SQL NULL value.

2321 An object-type's `queryName` attribute is used as the table name for the corresponding virtual table, and a  
2322 property's `queryName` attribute is used as the column name for the corresponding table column. Please  
2323 see the restrictions on `queryName` in the appropriate data model section.

2324 The Virtual Column for a multi-valued property MUST contain a single list value that includes all values of  
2325 the property.

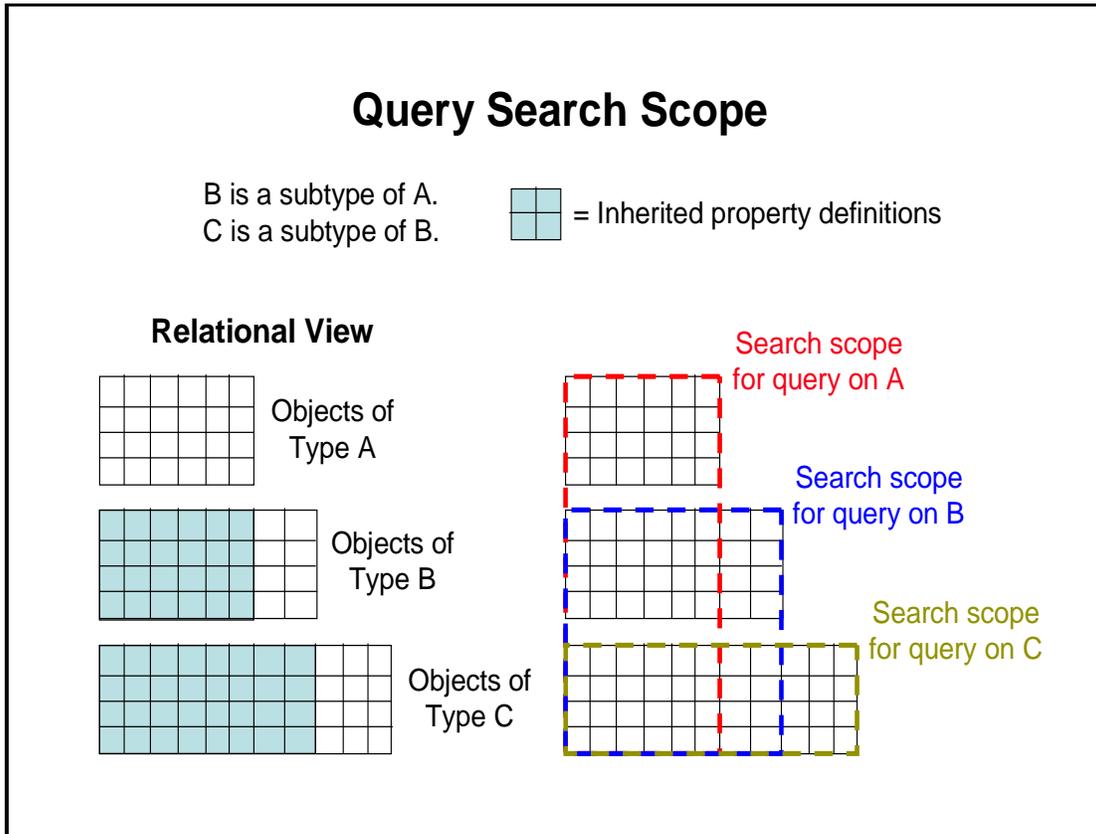
#### 2326 2.1.10.1.1 Object-Type Hierarchy in the Relational View Projection

2327 The Relational View projection of the CMIS Data Model ensures that the Virtual Table for a particular  
2328 Object-type is a complete super-set of the Virtual Table for any and all of its ancestor types.

2329 Additionally, an Object-Type definition's `includedInSupertypeQuery` specifies whether objects of that  
2330 Object-Type MUST be included in the Virtual Table for any of its ancestor types. If the  
2331 `includedInSupertypeQuery` attribute of the Object-Type is FALSE, then objects of that Object-Type  
2332 MUST NOT be included in the Virtual Table for any of its ancestor types.

2333 Thus the Virtual Table for an Object-type includes a row not only for each Object of that type, but all  
 2334 | Objects of any of that Object-types' Descendant Types for which the `includedInSupertypeQuery`  
 2335 attribute is TRUE.

2336 But since the Virtual Table will include only columns for properties defined in the Object-Type underlying  
 2337 the Virtual Table, a row that is a query result representing an Object of a Descendant Type can only  
 2338 include those columns for properties defined on the Object-Type underlying the Virtual Table.



2339

2340 **2.1.10.1.2 Content Streams**

2341 Content-streams are NOT exposed through this relational view.

2342 **2.1.10.1.3 Result Set**

2343 When a query is submitted, a set of pseudo CMIS objects will be returned. These pseudo objects are  
 2344 comprised of the properties specified in the select clause of the query statement.

2345 For each property in each object in the result set, the Repository MUST include the property definition ID  
 2346 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  
 2347 used).

2348 If the select clause of the query statement contains properties from a single type reference then the  
 2349 repository MAY represent these pseudo-objects with additional object information.

2350 **2.1.10.2 Query Language Definition**

2351 This query languages is based on a subset of the SQL-92 grammar. CMIS-specific language extensions  
 2352 to SQL-92 are called out explicitly.

2353 The basic structure of a CMIS query is a SQL statement that MUST include the following clauses:

2354 | • **SELECT [virtual columns]:** This clause identifies the set of virtual columns that will be included  
2355 | in the query results for each row.

2356 | • **FROM [Virtual Table Names]:** This clause identifies which Virtual Table(s) the query will run  
2357 | against.

2358 | Additionally, a CMIS query MAY include the following clauses:

2359 | • **WHERE [conditions]:** This clause identifies the constraints that rows MUST satisfy to be  
2360 | considered a result for the query.

2361 | • **ORDER BY [sort specification]:** This clause identifies the order in which the result rows MUST  
2362 | be sorted in the result row set.

### 2363 | 2.1.10.2.1 BNF Grammar

2364 | This BNF grammar is a "subset" of the SQL-92 grammar (ISO/IEC 9075: 1992 – Database Language  
2365 | SQL), except for some production alternatives. Specifically, except for these extensions, the following  
2366 | production rules are derived from the SQL-92 grammar. The non-terminals used in this grammar are also  
2367 | borrowed from the SQL-92 grammar without altering their semantics. Accordingly, the non-terminal  
2368 | <column name> is used for single-valued properties only so that the semantics of SQL can be preserved  
2369 | and borrowed. This approach not only facilitates comparison of the two query languages, and simplifies  
2370 | the translation of a CMIS query to a SQL query for a RDBMS-based implementation, but also allows  
2371 | future expansion of this query language to cover a larger subset of SQL with minimum conflict. The CMIS  
2372 | extensions are introduced primarily to support multi-valued properties and full-text search, and to test  
2373 | folder membership. Multi-valued properties are handled separately from single-valued properties, using  
2374 | separate non-terminals and separate production rules to prevent the extensions from corrupting SQL-92  
2375 | semantics.  
2376 |

2377  
2378 <CMIS 1.0 query statement> ::= <simple table> [ <order by clause> ]  
2379 <simple table> ::= SELECT <select list> <from clause> [ <where clause> ]  
2380 <select list> ::= "\*"\*  
2381 | <select sublist> [ { " " " <select sublist> }... ]  
2382 <select sublist> ::= <value expression> [ [ AS ] <column name> ]  
2383 | <qualifier> "\*"\*  
2384 | <multi-valued-column reference>  
2385 <value expression> ::= <column reference> | <numeric value function>  
2386 <column reference> ::= [ <qualifier> " " " ] <column name>  
2387 <multi-valued-column reference> ::= [ <qualifier> " " " ] <multi-valued-column name>  
2388 <numeric value function> ::= SCORE()  
2389 <qualifier> ::= <table name> | <correlation name>  
2390 <from clause> ::= FROM <table reference>  
2391 <table reference> ::= <table name> [ [ AS ] <correlation name> ]  
2392 | <joined table>  
2393 <joined table> ::= "(" <joined table> ")"  
2394 | <table reference> [ <join type> ] JOIN <table reference> <join specification>  
2395 <join type> ::= INNER | LEFT [ OUTER ]  
2396 <join specification> ::= ON <column reference> "=" <column reference>  
2397 <where clause> ::= WHERE <search condition>  
2398 <search condition> ::= <boolean term> | <search condition> OR <boolean term>  
2399 <boolean term> ::= <boolean factor> | <boolean term> AND <boolean factor>  
2400 <boolean factor> ::= [ NOT ] <boolean test>  
2401 <boolean test> ::= <predicate> | "(" <search condition> ")"  
2402 <predicate> ::= <comparison predicate> | <in predicate> | <like predicate> | <null predicate>  
2403 | <quantified comparison predicate> | <quantified in predicate>  
2404 | <text search predicate> | <folder predicate>  
2405 <comparison predicate> ::= <value expression> <comp op> <literal>  
2406 <comp op> ::= "=" | "<" | "<" | ">" | "<=" | ">=" | "<" | ">" | "<=" | ">="  
2407 <literal> ::= <signed numeric literal> | <character string literal> | <datetime literal> | <boolean literal>  
2408 <in predicate> ::= <column reference> [ NOT ] IN "(" <in value list> ")"  
2409 <in value list> ::= <literal> [ { " " " <literal> }... ]  
2410 <like predicate> ::= <column reference> [ NOT ] LIKE <character string literal>  
2411 <null predicate> ::= { <column reference> | <multi-valued-column reference> } IS [ NOT ] NULL  
2412 <quantified comparison predicate> ::= <literal> "=" ANY <multi-valued-column reference>  
2413 <quantified in predicate> ::= ANY <multi-valued-column reference> [ NOT ] IN "(" <in value list> ")"  
2414 <text search predicate> ::= CONTAINS "("  
2415 | [ <qualifier> " ," ] <quote> <text search expression> <quote> ")"  
2416 <folder predicate> ::= { IN\_FOLDER | IN\_TREE } "(" [ <qualifier> " " " ] <folder id> ")"  
2417 <order by clause> ::= ORDER BY <sort specification> [ { " " " <sort specification> }... ]  
2418 <sort specification> ::= <column reference> [ ASC | DESC ]  
2419 <correlation name> ::= <identifier>

2420 <table name> ::= <identifier> !! This MUST be the name of an object-type.  
2421 <column name> ::= <identifier> !! This MUST be the name of a single-valued property,  
2422 or an alias for a scalar output value.  
2423 <multi-valued-column name> ::= <identifier> !! This MUST be the name of a multi-valued property.  
2424 <folder id> ::= <character string literal> !! This MUST be the object identity of a folder object.  
2425 <identifier> ::= !! As defined by queryName attribute.  
2426 <signed numeric literal> ::= !! As defined by SQL-92 grammar.  
2427 <character string literal> ::= !! As defined by SQL-92 grammar. (i.e. enclosed in single-quotes)

2428

2429 | !! This is [an independent sub-grammar for full-text search criteria. It is isolatable from the query](#)  
2430 | [statement grammar. \(See 2.1.10.3 Escaping\)](#)

2431 <text search expression> ::= <conjunct> [ {<space> OR <space> <conjunct>} ... ]

2432 <conjunct> ::= <term> [ {<space> <term>} ... ]

2433 <term> ::= [-] <simple term>

2434 <simple term> ::= <word> | <phrase>

2435 <word> ::= <non space char> [ {<non space char>} ... ]

2436 <phrase> ::= <quote> <word> [ {<space> <word>} ... ] <quote>

2437 <space> ::= <space char> [ {<space char>} ... ]

2438 <non space char> ::= <char> - <space char>

2439 <space char> ::= ' '

2440 <char> ::= !! Any character

2441

2442 <datetime literal> ::= TIMESTAMP <quote> <datetime string> <quote>

2443 <datetime string> ::= YYYY-MM-DDThh:mm:ss.sss[Z | +hh:mm | -hh:mm]

2444 <boolean literal> ::= TRUE | FALSE | true | false

2445 | <quote> ::= ''''!! Single-quote only, consistent with SQL-92 string literal

### 2446 **2.1.10.2.2 SELECT Clause**

2447 The SELECT clause MUST contain exactly one of the following:

- 2448 | • A comma separated list of one or more column names.
  - 2449 | ○ If an explicit column list is provided: A repository MUST include in its result row set all of the
  - 2450 | columns specified in the SELECT clause.
- 2451 | • \*: If this token is specified, then the repository MUST return columns for ALL single-valued
- 2452 | properties defined in the Object-Types whose Virtual Tables are listed in the FROM clause, and
- 2453 | SHOULD also return all multi-valued properties.

2454 | All column names MUST be valid “queryName” values for properties that are defined as “queryable” in  
2455 | the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

### 2456 **2.1.10.2.3 FROM Clause**

2457 The FROM clause identifies which Virtual Table(s) the query will be run against, as described in the  
2458 previous section.

2459 The FROM clause MUST contain only the queryNames of Object-Types whose *queryable* attribute value  
2460 is TRUE.

2461 **2.1.10.2.3.1 Join Support**

2462 CMIS repositories MAY support the use of SQL JOIN queries, and MUST indicate their support level  
2463 using the ~~Optional Capability attribute "capabilityJoin".~~ Optional Capability attribute "capabilityJoin".

- 2464 • If the Repository's value for the capabilityJoin attribute is **none**, then no JOIN clauses can be  
2465 used in queries.
- 2466 • If the Repository's value for the capabilityJoin attribute is **inneronly**, then only inner JOIN  
2467 clauses can be used in queries.
- 2468 • If the Repository's value for the capabilityJoin attribute is **innerandouter**, then inner and/or outer  
2469 JOIN clauses can be used in queries.

2470 Only explicit joins using the "JOIN" keyword is supported. Queries MUST NOT include implicit joins as  
2471 part of the WHERE clause of a CMIS query.

2472 CMIS queries MUST only support join operations using the "equality" predicate on single-valued  
2473 properties.

2474 **2.1.10.2.4 WHERE Clause**

2475 This clause identifies the constraints that rows MUST satisfy to be considered a result for the query.

2476 All column names MUST be valid "queryName" or their aliased values for properties that are defined as  
2477 "queryable" in the Object-Type(s) whose Virtual Tables are listed in the FROM clause.

2478 Properties are defined to not support a "null" value, therefore the <null predicate> MUST be interpreted  
2479 as testing the not set or set state of the specified property.

2480 **2.1.10.2.4.1 Comparisons permitted in the WHERE clause.**

2481 SQL's simple comparison predicate, IN predicate, and LIKE predicate are supported, for single-valued  
2482 properties only (so that SQL's semantics is preserved). Boolean conjunction (AND), disjunction (OR), and  
2483 negation (NOT) of predicates are also supported.

2484 Repositories SHOULD support the comparisons for the property types as described in the list below.  
2485 Repositories MAY support additional comparisons and operators. Any additional operators not specified  
2486 are repository-specific:

2487  
2488 <Property Type>

2489 Supported Operators: <List of Operators supported on Type>

2490 Supported Literal: <Supported type of Literal in comparison>

2491

2492 String (Single)

2493 Supported Operators: =, <>, [NOT] LIKE

2494 Supported Literal: String

2495

2496 String (IN)

2497 Supported Operators: [NOT] IN

2498 Supported Literal: List of Strings

2499

2500 Decimal

2501 Supported Operators: =, <>, <, <=, >, >=

2502 Supported Literal: Decimal

2503

2504 Decimal (IN)  
2505 Supported Operators: [NOT] IN  
2506 Supported Literal: List of Decimal  
2507  
2508 Integer  
2509 Supported Operators: =, <>, <, <=, >, >=  
2510 Supported Literal: Integer  
2511  
2512 Integer (IN)  
2513 Supported Operators: [NOT] IN  
2514 Supported Literal: List of Integer  
2515  
2516 Boolean  
2517 Supported Operators: =  
2518 Supported Literal: <boolean literal>  
2519  
2520 DateTime  
2521 Supported Operators: =, <>, <\*, <=\*, >\*, >=\*  
2522 Supported Literal: <datetime literal>  
2523 \* - comparison is based on chronological before or after date.  
2524  
2525 DateTime (IN)  
2526 Supported Operators: [NOT] IN  
2527 Supported Literal: List of <datetime literal>~~'s'~~  
2528  
2529 ID  
2530 Supported Operators: =, <>  
2531 Supported Literal: String  
2532  
2533 ID (IN)  
2534 Supported Operators: [NOT] IN  
2535 Supported Literal: List of strings  
2536  
2537 URI  
2538 Supported Operators: =, <>  
2539 Supported Literal: String  
2540  
2541 URI (IN)  
2542 Supported Operators: [NOT] IN  
2543 Supported Literal: List of strings  
2544  
2545 URI

2546 Supported Operators: [NOT] LIKE

2547 Supported Literal: String

2548

2549 Operations on the SCORE() output MUST be treated the same as decimal operations.

2550

2551 When using properties in a join statement, comparison MUST be allowed on properties of the same types  
2552 as defined by the table above. Repositories MAY extend this behavior.

2553

2554 The ANY operation argument MUST be one of the properties found in the table above which supports  
2555 equality operations

#### 2556 **2.1.10.2.4.2 Multi-valued property support (SQL-92 Extension)**

2557 The CMIS query language includes several new non-terminals to expose semantics for querying multi-  
2558 valued properties, in a way that does not alter the semantics of existing SQL-92 production rules.

##### 2559 **2.1.10.2.4.2.1 Multi-valued column references**

2560 **BNF grammar structure:** <Multi-valued-column reference>, <multi-valued-column name>

- 2561 | • These are non-terminals defined for multi-valued properties whereas SQL-92's <column reference>  
2562 | and <column name> are retained for single-valued properties only. This is to preserve the single-  
2563 | value semantics of a regular `"column"` in the SQL-92 grammar.

##### 2564 **2.1.10.2.4.2.2 <Quantified comparison predicate>**

2565 The SQL-92 production rule for <quantified comparison predicate> is extended to accept a multi-valued  
2566 property in place of a <table subquery>. This operation is restricted to equality tests only.

2567

2568 <Table subquery> is not supported in CMIS-SQL.

2569

2570 The SQL-92 <quantifier> is restricted to ANY only.

2571

2572 The SQL-92 <row value constructor> is restricted to a literal only.

2573 **Example:**

```
2574 SELECT      Y.CLAIM_NUM, X.PROPERTY_ADDRESS, Y.DAMAGE_ESTIMATES  
2575 | FROM (POLICY AS X JOIN CLAIMS AS Y ON X.POLICY_NUM = Y.POLICY_NUM )  
2576 | WHERE      ( 100000 = ANY Y.DAMAGE_ESTIMATES )
```

2577 *(Note: DAMAGE\_ESTIMATES is a multi-valued Integer property.)*

##### 2578 **2.1.10.2.4.2.3 IN/ANY Predicate**

2579 **BNF grammar structure:** <Quantified in predicate>

2580

2581 CMIS-SQL exposes a new IN predicate defined for a multi-valued property. It is modeled after the SQL-  
2582 92 IN predicate, but since the entire predicate is different semantically, it has its own production rule in  
2583 the BNF grammar below.

2584

2585 The quantifier is restricted to ANY. The predicate MUST be evaluated to TRUE if at least one of the  
2586 | property's values is (or, is not, if NOT is specified) among the given list of literal values. Otherwise the  
2587 | predicate is evaluated to FALSE.

2588

2589 The ANY operation argument MUST be one of the properties found in the comparison list above which  
2590 supports IN operations.

2591 **Example:**

```
2592     SELECT      *
2593     FROM CAR_REVIEW
2594     WHERE      (MAKE = "buick") OR
2595                ( ANY FEATURES IN ("NAVIGATION SYSTEM", "SATELLITE RADIO",
2596                "MP3", "NAVIGATION SYSTEM", "SATELLITE RADIO", "MP3") ) (Note: FEATURES is a multi-
2597                valued String property.)
```

2598 **2.1.10.2.4.3 CONTAINS() predicate function (CMIS-SQL Extension)**

2599 **BNF grammar structure::** CONTAINS ( [ <qualifier> ,] " <text search expression> ")

2600 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided  
2601 by a Repository (~~See previous section.~~)(See previous section.)

2602 **Inputs:**

2603 **<Qualifier>**

2604 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the  
2605 FROM clause for the query.

2606 

- If specified, then the predicate SHOULD only be applied to objects in the specified Virtual  
2607 Table, but a repository MAY ignore the value of the parameter.

2608 

- If not specified, applies to the single virtual table. If the query is a join, a server SHOULD  
2609 throw an exception if the qualifier is not specified.

2610 **<Text Search Expression>**

2611 The <text search expression> parameter MUST be a character string , specifying the full-text  
2612 search criteria.

2613  
2614 The Text Search Expression may be a set of terms or phrases with an optional "!" to signal  
2615 negation. A phrase is defined as a word or group of words. A group of words must be  
2616 surrounded by quotes to be considered a single phrase.

2617  
2618 Terms separated by whitespace are AND'ed together.

2619 Terms separated by "OR" are OR'ed together

2620 Implicit "AND" has higher precedence than "OR"

2621 Within a word or phrase, each (single-)quote must also be escaped by a preceding backslash  
2622 "\"

2623 **Return value:**

2624 The predicate returns a Boolean value.

2625 The predicate MUST return TRUE if the object is considered by the repository as "relevant" with  
2626 respect to the given <text search expression> parameter.

2627 The predicate MUST return FALSE if the object is considered by the repository as not "relevant"  
2628 with respect to the given <text search expression> parameter.

2629 **Constraints:**

2630 At most one CONTAINS() function MUST be included in a single query statement. The repository  
2631 MUST throw an exception if more than one CONTAINS() function is found.

2632  
2633 The return value of the CONTAINS() function MAY only be included conjunctively (ANDed) with the  
2634 aggregate of all other predicates, if there is any, in the WHERE clause.

2635 **2.1.10.2.4.4 SCORE() predicate function**

2636 **BNF grammar structure:** SCORE ( )

2637 **Usage:** This is a predicate function that encapsulates the full-text search capability that MAY be provided  
2638 | by a Repository (~~See previous section.~~)([See previous section.](#))

2639 **Inputs:** No inputs MUST be provided for this predicate function.

2640 **Return value:**

2641 The SCORE() predicate function returns a decimal value in the interval [0,1] .

2642 A repository MUST return the value 0 if the object is considered by the repository as having  
2643 absolutely no relevance with respect to the CONTAINS() function specified in the query.

2644 A repository MUST return the value 1 if the object is considered by the repository as having  
2645 absolutely complete relevance with respect to the CONTAINS() function specified in the query.

2646 **Constraints:**

2647 The SCORE() function MUST only be used in queries that also include a CONTAINS() predicate  
2648 function

2649 The SCORE() function MUST only be used in the SELECT clause of a query. It MUST NOT be  
2650 used in the WHERE clause or in the ORDER BY clauses.

2651 An alias column name defined for the SCORE() function call in the SELECT clause (i.e.,  
2652 "SELECT SCORE() AS column\_name ...") may be used in the ORDER BY clause.

2653 If SCORE() is included in the SELECT clause and an alias column name is not provided, then a  
2654 query name of SEARCH\_SCORE is used for the query output, and the property definition ID is  
2655 repository-specific.

2656 **2.1.10.2.4.5 IN\_FOLDER() predicate function**

2657 **BNF grammar structure:** IN\_FOLDER( [ <qualifier>, ] <folder id> )

2658 **Usage:** This is a predicate function that tests whether or not a candidate object is a child-object of the  
2659 folder object identified by the given <folder id>.

2660 **Inputs:**

2661 **<qualifier>**

2662 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the  
2663 FROM clause for the query.

- 2664 | • If specified, then the predicate SHOULD only be applied to objects in the specified Virtual Table,  
2665 | but a repository MAY ignore the value of the parameter.  
2666 | • If not specified, applies to the single virtual table. If the query is a join, a server SHOULD throw an  
2667 | exception if the qualifier is not specified.

2668 **<folder id>**

2669 The value of this parameter MUST be the ID of a folder object in the repository.

2670 **Return value:**

2671 The predicate function MUST return TRUE if the object is a child-object of the folder specified by  
2672 <folder id>.

2673 The predicate function MUST return FALSE if the object is a NOT a child-object of the folder  
2674 specified by <folder id>.

2675 **2.1.10.2.4.6 IN\_TREE() predicate function**

2676 **BNF grammar structure:** IN\_TREE( [ <qualifier>, ] <folder id> )

2677 **Usage:** This is a predicate function that tests whether or not a candidate object is a descendant-object of  
2678 the folder object identified by the given <folder id>.

2679 **Inputs:**

2680 **<qualifier>**

2681 The value of this optional parameter MUST be the name of one of the Virtual Tables listed in the  
2682 FROM clause for the query.

2683 | 

- If specified, then the predicate SHOULD only be applied to objects in the specified Virtual  
2684 | Table, but a repository MAY ignore the value of the parameter.

2685 | 

- If not specified, applies to the single virtual table. If the query is a join, a server SHOULD  
2686 | throw an exception if the qualifier is not specified.

2687 **<folder id>**

2688 The value of this parameter MUST be the ID of a folder object in the repository.

2689 **Return value:**

2690 The predicate function MUST return TRUE if the object is a descendant-object of the folder  
2691 specified by <folder id>.

2692 The predicate function MUST return FALSE if the object is a NOT a descendant -object of the  
2693 folder specified by <folder id>.

2694 **2.1.10.2.5 ORDER BY Clause**

2695 This clause MUST contain a comma separated list of one or more column names.

2696 | All column names referenced in this clause MUST be valid `"queryName"` or their aliased values for  
2697 | properties defined as *orderable* in the Object-type(s) whose Virtual Tables are listed in the  
2698 | FROM clause.

2699 Only columns in the SELECT clause MAY be in the ORDER BY clause.

2700 Collation rules for the ORDER BY clause are repository specific.

2701 **2.1.10.3 Escaping**

2702 ~~Repositories MUST support the escaping of characters using a backslash (\) in the query statement. The~~  
2703 ~~backslash character (\) will be used to escape characters within quoted strings in the query as follows:~~

2704 | 1. ~~\ will represent a single-quote (') character~~

2705 | 2. ~~\\ will represent a backslash (\) character~~

2706 ~~Character escaping for character strings differs from SQL-92's escaping. A repository MUST support the~~  
2707 ~~escaping of certain literal characters in a character string, or in a text expression, using a backslash~~  
2708 ~~character (\) in the following manner. For a <character string literal>, which MUST BE a string enclosed~~  
2709 ~~in single-quotes according to the SQL-92 grammar, any occurrence of the single-quote character (') and~~  
2710 ~~the escape character (\) in the string MUST BE escaped. This applies to <folder id>, which is a <character~~  
2711 ~~string literal>. Furthermore, when a <character string literal> is used in a LIKE predicate, any occurrence~~  
2712 ~~of the percent character (%) and the underscore character (\_) in the string as a literal MUST BE escaped~~  
2713 ~~also. Therefore, within a quoted string in a query:~~

2714 | 

- ~~The double character \ represents a literal single-quote (') character.~~

2715 | 

- ~~The double character \\ represents a literal backslash (\) character.~~

2716 | 3. ~~Within a LIKE string, the double characters \% and \\_ will represent the a literal characters~~  
2717 | ~~%percent (%) character and , a literal underscore ( \_ ) character respectively.~~

2718 | 4. ~~All other instances of a \backslash (\) character are errors.~~

2719 Using double single-quotes (") as a SQL-92 way to escape a literal single-quote (') character SHOULD  
2720 BE supported as an allowable alternative to the double character \.

2721  
2722 For a <text search expression>, a second-level character escaping is required so that the <text search  
2723 expression> sub-grammar is *isolatable* from the query statement-level grammar. When a text search  
2724 expression is composed for a query according to the <text search expression> sub-grammar, any  
2725 occurrence of the following three characters in the expression as a literal character MUST BE escaped:  
2726 hyphen (-), single-quote ('), and the escape character (\). Then, before this expression is enclosed in  
2727 single-quotes and inserted into a CONTAINS() predicate, the query statement-level escaping rules  
2728 described in the above MUST BE applied. This two-level character escaping allows a query statement  
2729 parser, using statement-level escaping rules, to correctly extract a <text search expression> as a  
2730 character string literal independent of the <text search expression> sub-grammar. This extracted <text  
2731 search expression> can then be correctly interpreted by a full-text search parser independent of the  
2732 query-statement grammar, using second-level escaping rules. Since the <text search expression> sub-  
2733 grammar is isolated from the SQL-92 grammar, double single-quotes is not a valid way to escape a literal  
2734 single-quote character for second-level character escaping.

2735  
2736 An <identifier> in a query statement MUST conform to the SQL-92 identifier syntax, and MUST NOT  
2737 require character escaping.

2738  
2739 **Example 1:**

2740 A query statement that contains a full-text search for the literal string "John'sPresentation-Version2" may  
2741 be composed as:

2742 SELECT ... FROM ... WHERE ... CONTAINS('John\\\'sPresentation\\-Version2') ...

2743 A query parser extracts from this statement the text search expression "John\'sPresentation-Version2" as  
2744 a character string literal, and passes it to a text-search parser, which interprets it as a single-word full-text  
2745 search criteria: *John'sPresentation-Version2*.

2746  
2747 **Example 2:**

2748 A query statement that contains a full-text search for the phrase "Content Management" may be  
2749 composed as:

2750 SELECT ... FROM ... WHERE ... CONTAINS('\Content Management') ...

2751 A query parser extracts from this statement the text search expression "Content Management" as a  
2752 character string literal, and passes it to a text-search parser, which interprets it as a full-text search  
2753 criteria consisting of a single phrase: *Content Management*. There is no second-level escaping.

2754 **2.1.11 Change Log**

2755 CMIS provides a "change log" mechanism to allow applications to easily discover the set of changes  
2756 that have occurred to objects stored in the repository since a previous point in time. This change log can  
2757 then be used by applications such as search services that maintain an external index of the repository to  
2758 efficiently determine how to synchronize their index to the current state of the repository (rather than  
2759 having to query for all objects currently in the repository).

2760 Entries recorded in the change log are referred to below as "change events".

2761 Note that change events in the change log MUST be returned in ascending order from the time when the  
2762 change event occurred.

2763 **2.1.11.1 Completeness of the Change Log**

2764 The Change Log mechanism exposed by a repository MAY be able to return an entry for every change  
2765 ever made to content in the repository, or may only be able to return an entry for all changes made since  
2766 a particular point in time. This "completeness" level of the change log is indicated via the optional  
2767 ~~changesIncomplete~~ value found on the getRepositoryInfo serviceoptional changesIncomplete  
2768 value found on the getRepositoryInfo service response

2769 However, repositories MUST ensure that if an application requests the entire contents of the repository's  
2770 change log, that the contents of the change log includes ALL changes made to any object in the

2771 repository *after* the first change listed in the change log. (I.e. repositories MAY truncate events from the  
2772 change log on a **"first-in first-out"** basis, but not in any other order.)  
2773 A Repository MAY record events such as filing/unfiling/moving of Documents as change events on the  
2774 Documents, their parent Folder(s), or both the Documents and the parent Folders.

## 2775 2.1.11.2 Change Log Token

2776 The primary index into the change log of a repository is the **"change log token"**. The change log token is  
2777 an opaque string that uniquely identifies a particular change in the change log.

### 2778 2.1.11.2.1 **"Latest Change Token"** repository information

2779 Repositories that support the changeLogToken event MUST expose the latest change log token (i.e. the  
2780 change log token corresponding to the most recent change to any object in the repository) as a property  
2781 returned by the getRepositoryInfo service.

2782 This will enable applications to begin **"subscribing"** to the change log for a repository by discovering  
2783 what change log token they should use on a going-forward basis to discover change events to the  
2784 repository.

## 2785 2.1.11.3 Change Event

2786 A change event represents a single action that occurred to an object in the repository that affected the  
2787 persisted state of the object.

2788 A Repository that supports the change log capability MUST expose at least the following information for  
2789 each change object:

- 2790 • **ID ObjectID:** The ObjectID of the object to which the change occurred
- 2791 • **Enum ChangeType:** An enumeration that indicates the type of the change. Valid values are:
  - 2792 ○ `created`: The object was created.
  - 2793 ○ `updated`: The object was updated.
  - 2794 ○ `deleted`: The object was deleted
  - 2795 ○ `security`: The access control or security policy for the object were changed.
- 2796 • **<Properties> properties:** Additionally, for events of changeType **"updated"**, the repository MAY  
2797 optionally include the new values of properties on the object (if any).

2798 Repositories MUST indicate whether they include properties for **"updated"** change events via the  
2799 ~~optional enumCapabilityChanges~~ **capability** ~~optional enumCapabilityChanges~~ **capability**.

2800

## 2801 2.2 Services

2802 The Services section of the CMIS specification defines a set of services that are described in a  
2803 protocol/binding-agnostic fashion.

2804 Every protocol binding of the CMIS specification MUST implement all of the methods described in this  
2805 section or explain why the service is not implemented.

2806 However, the details of how each service & method is implemented will be described in those protocol  
2807 binding specifications.

### 2808 2.2.1 Common Service Elements

2809 The following elements are common across many of the CMIS services.

### 2810 2.2.1.1 Paging

2811 All of the methods that allow for the retrieval of a collection of CMIS objects support paging of their result  
2812 sets except where explicitly stated otherwise. The following pattern is used:

#### 2813 Input Parameters:

- 2814 • **(optional) Integer maxItems:** This is the maximum number of items to return in a response. The  
2815 repository MUST NOT exceed this maximum. Default is repository-specific.
- 2816 • **(optional) Integer skipCount:** This is the number of potential results that the repository MUST  
2817 skip/page over before returning any results. Defaults to 0.

#### 2818 Output Parameters:

- 2819 • **Boolean hasMoreItems:** TRUE if the Repository contains additional items after those contained in  
2820 the response. FALSE otherwise. If TRUE, a request with a larger skipCount or larger maxItems is  
2821 expected to return additional results (unless the contents of the repository has changed).
- 2822 • **Integer numItems:** If the repository knows the total number of items in a result set, the repository  
2823 SHOULD include the number here. If the repository does not know the number of items in a result  
2824 set, this parameter SHOULD not be set. The value in the parameter MAY NOT be accurate the next  
2825 time the client retrieves the result set or the next page in the result set.

2826 If the caller of a method does not specify a value for maxItems, then the Repository MAY select an  
2827 appropriate number of items to return, and MUST use the hasMoreItems output parameter to indicate if  
2828 any additional results were not returned.

2829 Repositories MAY return a smaller number of items than the specified value for maxItems.

2830 Each binding will express the above in context and may have different mechanisms for communicating  
2831 hasMoreItems and numItems.

### 2832 2.2.1.2 Retrieving additional information on objects in CMIS service calls

2833 Several CMIS services that return object information have the ability to return dependent object  
2834 information as part of their response, such as the Allowable Actions for an object, rendition information,  
2835 etc.

2836 The CMIS service methods that support returning a result set of objects will include the ability to return  
2837 the following object information:

- 2838 • Properties (retrieves a subset instead of additional information)
- 2839 • Relationships
- 2840 • Renditions
- 2841 • ACLs
- 2842 • AllowableActions

2843

2844 This section describes the input parameter & output pattern for those services. All input parameters are  
2845 optional.

#### 2846 2.2.1.2.1 Properties

2847 **Description:** All of the methods that allow for the retrieval of properties for CMIS Objects have a  
2848 **"Property Filter"** as an optional parameter, which allows the caller to specify a subset of properties for  
2849 Objects that MUST be returned by the repository in the output of the method.

#### 2850 Optional Input Parameter:

- 2851 • **String filter:** Value indicating which properties for Objects MUST be returned. Values are:  
2852 • **Not set:** The set of properties to be returned MUST be determined by the repository.  
2853 • **A comma-delimited list of property definition Query Names: The properties listed MUST be**  
2854 **returned.**

2855 |  “\*\*\*” : All properties MUST be returned for all objects.

2856 | ~~Repositories SHOULD return only the properties specified in the property filter if they exist on the object's~~  
2857 | ~~type definition.~~

2858 |  
2859 | If a property is requested by a filter, a property element MUST be returned for that property. A repository  
2860 | MAY return additional properties. If a returned property is in a "not set" state, a value element MUST NOT  
2861 | be returned for that property.

2862 |  
2863 | If a property filter specifies a property that is '~~not set~~not set', it MUST be represented as a property  
2864 | element without a value element.

## 2865 | 2.2.1.2.2 Relationships

2866 | **Description:** Used to retrieve the relationships in which the object(s) are participating.

2867 | **Optional Input Parameter:**

2868 | • **Enum includeRelationships:** Value indicating what relationships in which the objects returned  
2869 | participate MUST be returned, if any. Values are:

2870 |       none : No relationships MUST be returned. (Default).

2871 |       source : Only relationships in which the objects returned are the source MUST be  
2872 |       returned.

2873 |       target : Only relationships in which the objects returned are the target MUST be  
2874 |       returned.

2875 |       both : Relationships in which the objects returned are the source or the target MUST be  
2876 |       returned.

2877 | **Output Parameter for each object:**

2878 | • **<Array> Relationships:** A collection of the relationship objects.

## 2879 | 2.2.1.2.3 Policies

2880 | **Description:** Used to retrieve the policies currently applied to the object(s).

2881 | **Optional Input Parameter:**

2882 | • **Boolean includePolicyIds:** If TRUE, then the Repository MUST return the Ids of the policies  
2883 | applied to the object. Defaults to FALSE.

2884 | **Output Parameter or each object:**

2885 | • **<Array> Policies:** A collection of the policy objects.

## 2886 | 2.2.1.2.4 Renditions

2887 | **Description:** Used to retrieve the renditions of the object(s).

2888 | **Optional Input Parameter:**

2889 | • **String renditionFilter:** The Repository MUST return the set of renditions whose kind matches this  
2890 | filter. See section below for the filter grammar.

2891 |       o Defaults to “cmis:none”.

2892 | **Output Parameter for each object:**

2893 | • **<Array> Renditions:** The set of renditions.

### 2894 | 2.2.1.2.4.1 Rendition Filter Grammar

2895 | The Rendition Filter grammar is defined as follows:

2896 <renditionInclusion> ::= <none> | <wildcard> | <termlist>  
 2897 <termlist> ::= <term> | <term> ',' <termlist>  
 2898 <term> ::= <kind> | <mimetype>  
 2899 <kind> ::= <text>  
 2900 <mimetype> ::= <type> '/' <subtype>  
 2901 <type> ::= <text>  
 2902 <subtype> ::= <text> | <wildcard>  
 2903 <text> ::= ~~/\*!!~~ any char except whitespace~~\*/~~  
 2904 <wildcard> ::= '\*'  
 2905 <none> ::= 'cmis:none'

2906 An inclusion pattern allows:

- 2907 • **Wildcard** : include all associated Renditions
- 2908 • **Comma-separated list of Rendition kinds or mimetypes** : include only those Renditions that
- 2909 match one of the specified kinds or mimetypes
- 2910 • **cmis:none**: (Default) exclude all associated Renditions

2911 Examples:

- 2912 • \* (include all Renditions)
- 2913 • cmis:thumbnail (include only Thumbnails)
- 2914 • Image/\* (include all image Renditions)
- 2915 • application/pdf, application/x-shockwave-flash (include web ready Renditions)
- 2916 • cmis:none (exclude all Renditions)

### 2917 2.2.1.2.5 ACLs

2918 **Description:** Used to retrieve the ACLs for the object(s) described in the service response.

2919 **Optional Input Parameter:**

- 2920 • **Boolean includeACL:** If TRUE, then the Repository MUST return the ACLs for each object in the
- 2921 result set. Defaults to FALSE.

2922 **Output Parameter for each object:**

- 2923 • **<Array> ACLs:** The list of access control entries of the ACL for the object.

### 2924 2.2.1.2.6 Allowable Actions

2925 **Description:** Used to retrieve the allowable actions for the object(s) described in the service response.

2926 **Optional Input Parameter:**

- 2927 • **Boolean includeAllowableActions:** If TRUE, then the Repository MUST return the available
- 2928 actions for each object in the result set. Defaults to FALSE.

2929 **Output Parameter for each object:**

- 2930 • **<Array> AllowableActions:** See `cmisAllowableActionsType` in The list of allowable actions for
- 2931 the `CMIS-schema` object.

### 2932 2.2.1.3 Change Tokens

2933 The CMIS base object-type definitions include an opaque string `"_ChangeToken"` property that a

2934 Repository MAY use for optimistic locking and/or concurrency checking to ensure that user updates do

2935 not conflict.

2936 If a Repository provides values for the ChangeToken property for an Object, then all invocations of the  
2937 `update` methods on that object (updateProperties, setContentStream, deleteContentStream) MUST  
2938 provide the value of the changeToken property as an input parameter, and the Repository MUST throw  
2939 an updateConflictException if the value specified for the changeToken does NOT match the  
2940 changeToken value for the object being updated.

## 2941 2.2.1.4 Exceptions

2942 The following sections list the complete set of exceptions that MAY be returned by a repository in  
2943 response to a CMIS service method call.

### 2944 2.2.1.4.1 General Exceptions

2945 The following exceptions MAY be returned by a repository in response to ANY CMIS service method call.

2946 The `Cause` field indicates the circumstances under which a repository SHOULD return a particular  
2947 exception.

#### 2948 `invalidArgument`

2949 Cause: One or more of the input parameters to the service method is missing or invalid.

2950

#### 2951 `objectNotFound`

2952 Cause: The service call has specified an object that does not exist in the Repository.

2953

#### 2954 `notSupported`

2955 Cause: The service method invoked requires ~~an optional capability~~ an optional capability not  
2956 supported by the repository.

2957

#### 2958 `permissionDenied`

2959 Cause: The caller of the service method does not have sufficient permissions to perform the  
2960 operation.

2961

#### 2962 `runtime`

2963 Cause: Any other cause not expressible by another CMIS exception.

### 2964 2.2.1.4.2 Specific Exceptions

2965 The following exceptions MAY be returned by a repository in response to one or more CMIS service  
2966 methods calls.

2967 For each exception, the general intent is listed as well as a list of the methods which MAY cause the  
2968 exception to be thrown.

#### 2969 `constraint`

2970 Intent: The operation violates a Repository- or Object-level constraint defined in the CMIS  
2971 domain model.

2972 Methods:

- 2973 • **Navigation Services:**

- 2974 ○ getObjectParents

- 2975 • **Object Services:**

- 2976 ○ createDocument

- 2977 ○ createDocumentFromSource

- 2978 ○ createFolder

- 2979 ○ createRelationship
- 2980 ○ createPolicy
- 2981 ○ updateProperties
- 2982 ○ moveObject
- 2983 ○ deleteObject
- 2984 ○ setContentStream
- 2985 ○ deleteContentStream
- 2986 ● **Multi-filing Services:**
- 2987 ○ addObjectToFolder
- 2988 ● **Versioning Services:**
- 2989 ○ checkOut
- 2990 ○ cancelCheckOut
- 2991 ○ checkIn
- 2992 ● **Policy Services:**
- 2993 ○ applyPolicy
- 2994 ○ removePolicy
- 2995 ● **Change Log Services:**
- 2996 ○ getContentChanges

2997

2998 **contentAlreadyExists**

2999 Intent: The operation attempts to set the content stream for a Document that already has a  
 3000 content stream without explicitly specifying the `overwriteFlag` parameter.

3001 Methods:

- 3002 ● **Object Services:**
- 3003 ○ setContentStream

3004

3005 **filterNotValid**

3006 Intent: The property filter or rendition filter input to the operation is not valid.

3007 Methods:

- 3008 ● **Navigation Services:**
- 3009 ○ getDescendants
- 3010 ○ getChildren
- 3011 ○ getFolderParent
- 3012 ○ getObjectParents
- 3013 ○ getCheckedOutDocs
- 3014 ● **Object Services:**
- 3015 ○ getProperties
- 3016 ○ getRenditions
- 3017 ○ getObject
- 3018 ○ getObjectByPath
- 3019 ● **Versioning Services:**
- 3020 ○ getPropertiesOfLatestVersion



- 3064 ○ getContentStream
- 3065 ○ setContentStream
- 3066 • **Versioning Services:**
- 3067 ○ checkIn

3068

3069 **updateConflict**

3070 Intent: The operation is attempting to update an object that is no longer current (as  
3071 determined by the repository).

3072 Methods:

- 3073 • **Object Services:**
- 3074 ○ updateProperties
- 3075 ○ moveObject
- 3076 ○ deleteObject
- 3077 ○ deleteTree
- 3078 ○ setContentStream
- 3079 ○ deleteContentStream
- 3080 • **Versioning Services:**
- 3081 ○ checkOut
- 3082 ○ cancelCheckOut
- 3083 ○ checkIn

3084

3085 **versioning**

3086 Intent: The operation is attempting to perform an action on ~~a non-current version~~ a non-  
3087 current version of a Document that cannot be performed on a non-current version.

3088 Methods:

- 3089 • **Object Services:**
- 3090 ○ updateProperties
- 3091 ○ moveObject
- 3092 ○ setContentStream
- 3093 ○ deleteContentStream
- 3094 • **Versioning Services:**
- 3095 ○ checkOut
- 3096 ○ cancelCheckOut
- 3097 ○ checkIn

3098 **2.2.1.5 ACLs**

3099 Those services which allow for the setting of ACLs may take the optional macro cmis:user which allows  
3100 the caller to indicate the operation applies to the current authenticated user.

3101 **2.2.2 Repository Services**

3102 The Repository Services (getRepositoryInfo, getRepositoryInfo, getTypeChildren, getTypeDescendants,  
3103 getTypeDefinition) are used to discover information about the repository, including information about the  
3104 repository and the object-types defined for the repository.

### 3105 **2.2.2.1 getRepositories**

3106 **Description:** Returns a list of CMIS repositories available from this CMIS service endpoint.

#### 3107 **2.2.2.1.1 Inputs**

3108 None.

#### 3109 **2.2.2.1.2 Outputs**

3110 A list of repository information, with (at least) the following information for each entry:

- 3111 • **ID repositoryId:** The identifier for the Repository.
- 3112 • **String repositoryName:** A display name for the Repository.

#### 3113 **2.2.2.1.3 Exceptions Thrown & Conditions**

3114 See section 2.2.1.4.1 General Exceptions

### 3115 **2.2.2.2 getRepositoryInfo**

3116 **Description:** Returns information about the CMIS repository, the [optional capabilities](#) optional capabilities  
3117 it supports and its Access Control information if applicable.

#### 3118 **2.2.2.2.1 Inputs**

3119 **Required:**

- 3120 • **ID repositoryId:** The identifier for the Repository.

#### 3121 **2.2.2.2.2 Outputs**

- 3122 • **ID repositoryId:** The identifier for the Repository.
  - 3123 ○ **Note:** This MUST be the same identifier as the input to the method.
- 3124 • **String repositoryName:** A display name for the Repository.
- 3125 • **String repositoryDescription:** A display description for the Repository.
- 3126 • **String vendorName:** A display name for the vendor of the Repository's underlying application.
- 3127 • **String productName:** A display name for the Repository's underlying application.
- 3128 • **String productVersion:** A display name for the version number of the Repository's  
3129 underlying application.
- 3130 • **ID rootFolderId:** The ID of the Root Folder Object for the Repository.
- 3131 • **<List of capabilities>:** The set of values for the repository-optional capabilities specified in  
3132 [section 2.1.1.4section](#) 2.1.1.1 Optional Capabilities
- 3133 • **String latestChangeLogToken:** The change log token corresponding to the most recent  
3134 change event for any object in the repository.
- 3135 • **String cmisVersionSupported:** A decimal that indicates what version of the CMIS  
3136 specification this repository supports as specified in 2.1.1.2 Implementation Information.
- 3137 • **URI thinClientURI:** A optional repository-specific URI pointing to the repository's web  
3138 interface.
- 3139 • **Boolean changesIncomplete:** Indicates whether or not the repository's change log can return  
3140 all changes ever made to any object in the repository or only changes made after a particular  
3141 point in time. Applicable when the repository's optional capability `capabilityChanges` is not  
3142 none.

- 3143 |           o If FALSE, then the change log can return all changes ever made to every object.
- 3144 |           o If TRUE, then the change log includes all changes made since a particular point in time,
- 3145 |           but not all changes ever made.
- 3146 |       • **<List of enum values>** `changesOnType`: Indicates whether changes are available for base
- 3147 |       types in the repository. Valid values are from `enumBaseObjectTypeIds`. See section 2.1.11
- 3148 |       [Change Log-Change Log](#).
- 3149 |           o **cmis:document**
- 3150 |           o **cmis:folder**
- 3151 |           o **cmis:policy**
- 3152 |           o **cmis:relationship**
- 3153 |       • **Enum** `supportedPermissions`: specifies which types of permissions are supported.
- 3154 |           o **basic**: indicates that the CMIS Basic permissions are supported.
- 3155 |           o **repository**: Indicates that repository specific permissions are supported.
- 3156 |           o **both**: indicates that both CMIS basic permissions and repository specific permissions are
- 3157 |           supported.
- 3158 |       • **Enum** `propagation`: The list of allowed values for `applyACL`, which control how non-direct
- 3159 |       ACEs are handled by the repository:
  - 3160 |           o **objectonly**: indicates that the repository is able to apply ACEs without changing the
  - 3161 |           ACLs of other objects – i.e. ACEs are applied, potentially **“breaking”** the **“sharing”**
  - 3162 |           dependency for non-direct ACEs.
  - 3163 |           o **propagate**: indicates that the repository is able to apply ACEs to a given object and
  - 3164 |           propagate this change to all inheriting objects – i.e. ACEs are applied with the (intended)
  - 3165 |           side effect to inheriting objects.
  - 3166 |           o **repositorydetermined**: indicates that the repository uses its own mechanisms to
  - 3167 |           handle non-direct ACEs when applying ACLs.
- 3168 |       • **<Array> Permission** `permissions`: The list of repository-specific permissions the repository
- 3169 |       supports for managing ACEs (see section 2.8 Access Control).
- 3170 |       • **<Array> PermissionMapping** `mapping`: The list of mappings for the CMIS Basic permissions to
- 3171 |       allowable actions (see section 2.8 Access Control).
- 3172 |       • **String** `principalAnonymous`: If set, this field holds the principal who is used for anonymous
- 3173 |       access. This principal can then be passed to the ACL services to specify what permissions
- 3174 |       anonymous users should have.
- 3175 |       • **String** `principalAnyone`: If set, this field holds the principal who is used to indicate any
- 3176 |       authenticated user. This principal can then be passed to the ACL services to specify what
- 3177 |       permissions any authenticated user should have.

3178 | The `cmisRepositoryInfoType` schema describes the markup that will be included in all CMIS protocol  
3179 | bindings to implement this service.

### 3180 | **2.2.2.2.3 Exceptions Thrown & Conditions**

3181 | See section 2.2.1.4.1 General Exceptions

### 3182 | **2.2.2.3 getTypeChildren**

3183 | **Description:** Returns the list of ~~Object-Types~~[Object-Types](#) defined for the Repository that are children of  
3184 | the specified Type.

#### 3185 | **2.2.2.3.1 Inputs**

3186 | **Required:**

- 3187 | • **String repositoryId:** The identifier for the Repository.

3188 | **Optional:**

- 3189 | • **String typeld:** The typeld of an Object-Type specified in the Repository.
- 3190 |     ○ If specified, then the Repository MUST return all of child types of the specified type.
- 3191 |     ○ If not specified, then the Repository MUST return all Base Object-Types.
- 3192 | • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
- 3193 | definitions for each Object-Type returned.
- 3194 |     ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.
- 3195 | • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3196 | • **Integer skipCount:** See section 2.2.1.1 Paging.

3197 | **2.2.2.3.2 Outputs**

3198 | **<Array> Object-Types:** The list of child ~~Object-Types~~Object-Types defined for the given typeld.

3199 | **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3200 | **Optional:**

3201 | **Integer numItems:** See section 2.2.1.1 Paging.

3202 | **2.2.2.3.3 Exceptions Thrown & Conditions**

3203 | See section 2.2.1.4.1 General Exceptions

3204 | **2.2.2.4 getTypeDescendants**

3205 | **Description:** Returns the set of descendant ~~Object-Types~~Object-Types defined for the Repository under

3206 | the specified Type.

3207 | **Notes:**

- 3208 | • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3209 | • The order in which results are returned is repository-specific.

3210 | **2.2.2.4.1 Inputs**

3211 | **Required:**

- 3212 | • **String repositoryId:** The identifier for the Repository.

3213 | **Optional:**

- 3214 | • **String typeld:** The typeld of an Object-Type specified in the Repository.
- 3215 |     ○ If specified, then the Repository MUST return all descendant types for the specified type.
- 3216 |     ○ If not specified, then the Repository MUST return all types and MUST ignore the value of
- 3217 | the depth parameter
- 3218 | • **Integer depth:** The number of levels of depth in the type hierarchy from which to return results.
- 3219 | Valid values are:
- 3220 |     ○ **1:** Return only types that are children of the type.
- 3221 |     ○ **<Integer value greater than 1>:** Return only types that are children of the type and
- 3222 | descendants up to <value> levels deep.
- 3223 |     ○ **-1:** Return ALL descendant types at all depth levels in the CMIS hierarchy.
- 3224 |     ○ The default value is repository specific and SHOULD be at least 2 or -1.
- 3225 | • **Boolean includePropertyDefinitions:** If TRUE, then the Repository MUST return the property
- 3226 | definitions for each Object-Type returned.

3227 |                   ○ If FALSE (default), the Repository MUST return only the attributes for each Object-Type.

#### 3228 | **2.2.2.4.2 Outputs**

3229 |     **<Array> Object-Types:** The hierarchy of ~~Object-Types~~Object-Types defined for the Repository.

#### 3230 | **2.2.2.4.3 Exceptions Thrown & Conditions**

3231 | See section 2.2.1.4.1 General Exceptions

3232 |     • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with  
3233 |         an invalid depth.

#### 3234 | **2.2.2.5 getTypeDefinition**

3235 | **Description:** Gets the definition of the specified Object-Type.Inputs

##### 3236 | **2.2.2.5.1 Inputs**

3237 | **Required:**

- 3238 |     • **String repositoryId:** The identifier for the Repository.
- 3239 |     • **String typeId:** The typeId of an Object-Type specified in the Repository.

##### 3240 | **2.2.2.5.2 Outputs**

- 3241 |     • Object-type including all property definitions. See section 2.1.3.3 (Object-Type Property  
3242 |         Definitions) for further details.

#### 3243 | **2.2.2.5.3 Exceptions Thrown & Conditions**

3244 | See section 2.2.1.4.1 General Exceptions

### 3245 | **2.2.3 Navigation Services**

3246 | The Navigation Services (getDescendants, getChildren, getFolderParent, getObjectParents,  
3247 | getCheckedoutDocs), are used to traverse the folder hierarchy in a CMIS **Repository**, and to locate  
3248 | Documents that are checked out.

#### 3249 | **2.2.3.1 getChildren**

3250 | **Description:** Gets the list of child objects contained in the specified folder.

3251 | **Notes:**

- 3252 |     • If the Repository supports the optional `“VersionSpecificFiling”` capability, then the repository  
3253 |         MUST return the document versions filed in the specified folder.
- 3254 |         ○ Otherwise, the latest version of the documents MUST be returned.

##### 3255 | **2.2.3.1.1 Inputs**

3256 | **Required:**

- 3257 |     • **ID repositoryId:** The identifier for the Repository.
- 3258 |     • **ID folderId:** The identifier for the folder.

3259 | **Optional:**

- 3260 |     • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3261 |     • **Integer skipCount:** See section 2.2.1.1 Paging.

- 3262 • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and  
3263 the ascending modifier `"ASC"` or the descending modifier `"DESC"` for each query name. A  
3264 repository's handling of the orderBy input is repository-specific.
- 3265 • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the  
3266 matched object if they exist on the matched object type definition and in the filter.
- 3267 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3268 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3269 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3270 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each  
3271 child object for use in constructing that object's path.

### 3272 2.2.3.1.2 Outputs

- 3273 • **<Array> ObjectResults:** A list of the child objects for the specified folder. Each object result  
3274 MUST include the following elements if they are requested:
  - 3275 ○ **<Array> Properties:** The list of properties for the object.
  - 3276 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3277 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3278 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
  - 3279 ○ **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.
- 3280 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

3281 **Optional:**

- 3282 • **Integer numItems:** See section 2.2.1.1 Paging.

### 3283 2.2.3.1.3 Exceptions Thrown & Conditions

- 3284 • See section 2.2.1.4.1 General Exceptions
- 3285 • `filterNotValid:` The Repository MUST throw this exception if `filter` property `or rendition` filter  
3286 input parameter is not valid.
- 3287 • `invalidArgument:` if the specified folder is not a folder

### 3288 2.2.3.2 getDescendants

3289 **Description:** Gets the set of `descendant-objects` `descendant objects` contained in the specified folder or  
3290 any of its child-folders.

3291 **Notes:**

- 3292 • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3293 • The order in which results are returned is repository-specific..
- 3294 • If the Repository supports the optional capability `capabilityVersionSpecificFiling`, then  
3295 the repository MUST return the document versions filed in the specified folder or its descendant  
3296 folders. Otherwise, the latest version of the documents MUST be returned.
- 3297 • If the Repository supports the optional capability `capabilityMutlifiling` and the same  
3298 document is encountered multiple times in the hierarchy, then the repository MUST return that  
3299 document each time is encountered.

### 3300 2.2.3.2.1 Inputs

3301 **Required:**

- 3302 • **ID repositoryId:** The identifier for the Repository.

- 3303 | • **ID folderId:** The identifier for the folder.
- 3304 | **Optional:**
- 3305 | • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results.
- 3306 | Valid values are:
- 3307 |     o **1:** Return only objects that are children of the folder.
- 3308 |     o **<Integer value greater than 1>:** Return only objects that are children of the folder and
- 3309 |     descendants up to <value> levels deep.
- 3310 |     o **-1:** Return ALL descendant objects at all depth levels in the CMIS hierarchy.
- 3311 |     o The default value is repository specific and SHOULD be at least 2 or -1
- 3312 | • **String filter:** See section 2.2.1.2.1 Properties.
- 3313 | • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3314 | • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3315 | • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3316 | • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each
- 3317 | child object for use in constructing that object's path.

### 3318 | 2.2.3.2.2 Outputs

- 3319 | • **<Array> ObjectResults:** A list of the descendant objects for the specified folder. Each object
- 3320 | result MUST include the following elements if they are requested:
- 3321 |     o **<Array> Properties:** The list of properties for the object.
- 3322 |     o **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3323 |     o **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3324 |     o **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3325 |     o **String PathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

### 3326 | 2.2.3.2.3 Exceptions Thrown & Conditions

3327 | See section 2.2.1.4.1 General Exceptions

- 3328 | • **filterNotValid:** The Repository MUST throw this exception if the ise property or rendition filter
- 3329 | input parameter is not valid.
- 3330 | • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with
- 3331 | "depth = 0"
- 3332 | • **invalidArgument:** if the specified folder is not a folder

### 3333 | 2.2.3.3 getFolderTree

3334 | **Description:** Gets the set of descendant folder objects contained in the specified folder.

3335 |

3336 | **Notes:**

- 3337 | • This method does NOT support paging as defined in the 2.2.1.1 Paging section.
- 3338 | • The order in which results are returned is repository-specific..

### 3339 | 2.2.3.3.1 Inputs

3340 | **Required:**

- 3341 | • **ID repositoryId:** The identifier for the Repository.
- 3342 | • **ID folderId:** The identifier for the folder.

3343 **Optional:**

- 3344 • **Integer depth:** The number of levels of depth in the folder hierarchy from which to return results.  
3345 Valid values are:
  - 3346 ○ **1:** Return only folders that are children of the folder.
  - 3347 ○ **<Integer value greater than 1>:** Return only folders that are children of the folder and  
3348 descendant folders up to <value> levels deep.
  - 3349 ○ **-1:** Return ALL descendant folders at all depth levels in the CMIS hierarchy.
  - 3350 ○ The default value is repository specific and SHOULD be at least 2 or -1
- 3351 • **String filter:** See section 2.2.1.2.1 Properties.
- 3352 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3353 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3354 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3355 • **Boolean includePathSegment:** Defaults to FALSE. If TRUE, returns a PathSegment for each  
3356 child object for use in constructing that object's path.

3357 **2.2.3.3.2 Outputs**

- 3358 • **<Array> ObjectResults:** A list of the descendant folders for the specified folder. Each object  
3359 result MUST include the following elements if they are requested:
  - 3360 ○ **<Array> Properties:** The list of properties for the object.
  - 3361 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3362 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3363 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
  - 3364 ○ **String pathSegment:** If includePathSegment was TRUE. See section 2.1.5.3 Paths.

3365 **2.2.3.3.3 Exceptions Thrown & Conditions**

- 3366 • See section 2.2.1.4.1 General Exceptions
- 3367 • **filterNotValid:** The Repository MUST throw this exception if **thi**se property **or rendition** filter  
3368 input parameter is not valid.
- 3369 • **invalidArgument:** The Repository MUST throw this exception if the service is invoked with  
3370 an invalid depth
- 3371 • **invalidArgument:** if the specified folder is not a folder

3372  
3373

3374 **2.2.3.4 getFolderParent**

3375 **Description:** Gets the parent folder object for the specified folder object.

3376 **2.2.3.4.1 Inputs**

3377 **Required:**

- 3378 • **ID repositoryId:** The identifier for the Repository.
- 3379 • **ID folderId:** The identifier for the folder.

3380 **Optional:**

- 3381 • **String filter:** See section 2.2.1.2.1 Properties.

## 3382 2.2.3.4.2 Outputs

- 3383 • **Object:** The parent folder object of the specified folder.

## 3384 2.2.3.4.3 Exceptions Thrown & Conditions

- 3385 • See section 2.2.1.4.1 General Exceptions
- 3386 • `filterNotValid`: The Repository MUST throw this exception if this property filter input  
3387 parameter is not valid.
- 3388 • `invalidArgument`: The Repository MUST throw this exception if the `folderId` input is the root  
3389 folder.

## 3390 2.2.3.5 getObjectParents

3391 **Description:** Gets the parent folder(s) for the specified non-folder, fileable object.

### 3392 2.2.3.5.1 Inputs

#### 3393 Required:

- 3394 • **ID repositoryId:** The identifier for the Repository.
- 3395 • **ID objectId:** The identifier for the object.

#### 3396 Optional:

- 3397 • **String filter:** See section 2.2.1.2.1 Properties
- 3398 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3399 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3400 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3401 • **Boolean includeRelativePathSegment:** See section 2.1.5.3 Paths.

### 3402 2.2.3.5.2 Outputs

- 3403 • **<Array> ObjectResults:** A list of the parent folder(s) of the specified objects. Empty for unfiled  
3404 objects or for the root folder. Each object result MUST include the following elements if they are  
3405 requested:
  - 3406 ○ **<Array> Properties:** The list of properties for the object.
  - 3407 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3408 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3409 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
  - 3410 ○ **String relativePathSegment:** If `includeRelativePathSegment` was TRUE. See section  
3411 2.1.5.3 Paths.

### 3412 2.2.3.5.3 Exceptions Thrown & Conditions

- 3413 • See section 2.2.1.4.1 General Exceptions
- 3414 • `constraint`: The Repository MUST throw this exception if this method is invoked on an object  
3415 who Object-Type Definition specifies that it is not fileable.
- 3416 • `filterNotValid`: The Repository MUST throw this exception if this property or rendition filter  
3417 input parameter is not valid.

## 3418 2.2.3.6 getCheckedOutDocs

3419 **Description:** Gets the list of documents that are checked out that the user has access to.

### 3420 2.2.3.6.1 Inputs

#### 3421 Required:

- 3422 • **ID repositoryId:** The identifier for the Repository.

#### 3423 Optional:

- 3424 • **ID folderId:** The identifier for a folder in the repository from which documents should be returned.
  - 3425 ○ If specified, the Repository MUST only return checked out documents that are child-
  - 3426 objects of the specified folder.
  - 3427 ○ If not specified, the Repository MUST return checked out documents from anywhere in
  - 3428 the repository hierarchy.
- 3429 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 3430 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 3431 • **String orderBy:** The orderBy parameter MUST be a comma-separated list of query names and
- 3432 the ascending modifier `"ASC"` or the descending modifier `"DESC"` for each query name. A
- 3433 repository's handling of the orderBy input is repository-specific.
- 3434 • **String filter:** See section 2.2.1.2.1 Properties.
- 3435 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3436 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3437 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

### 3438 2.2.3.6.2 Outputs

- 3439 • **<Array> ObjectResults:** A list of checked out documents. Each object result MUST include the
- 3440 following elements if they are requested:
  - 3441 ○ **<Array> Properties:** The list of properties for the object.
  - 3442 ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 3443 ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 3444 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 3445 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

#### 3446 Optional:

- 3447 • **Integer numItems:** See section 2.2.1.1 Paging.

3448

### 3449 2.2.3.6.3 Exceptions Thrown & Conditions

- 3450 • See section 2.2.1.4.1 General Exceptions
- 3451 • `filterNotValid`: The Repository MUST throw this exception if `thi`se property `or rendition` filter
- 3452 input parameter is not valid.

## 3453 2.2.4 Object Services

3454 CMIS provides ID-based CRUD (Create, Retrieve, Update, Delete), operations on objects in a Repository.

### 3455 2.2.4.1 createDocument

3456 **Description:** Creates a document object of the specified type (given by the `cmis:objectTypeId` property)

3457 in the (optionally) specified location.

### 3458 2.2.4.1.1 Inputs

#### 3459 Required:

3460 **◦ ID repositoryId:** The identifier for the Repository.

3461 **◦ <Array> properties:** The property values that MUST be applied to the newly-created Document  
3462 Object.

#### 3463 Optional:

3464 • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-  
3465 created Document Object.

3466 ◦ This parameter MUST be specified if the Repository does NOT support the optional  
3467 `"unfiling"` capability.

3468 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the newly-  
3469 created Document Object. The method of passing the contentStream to the server and the  
3470 encoding mechanism will be specified by each specific binding. MUST be required if the type  
3471 requires it.

3472 • **Enum versioningState:** An enumeration specifying what the versioning state of the newly-created  
3473 object MUST be. If the repository does not support versioning, the repository MUST ignore the  
3474 versioningState parameter. Valid values are:

3475 ◦ `none`: The document MUST be created as a non-versionable document.

3476 ◦ `checkedout`: The document MUST be created in the checked-out state. The checked-  
3477 out document MAY be visible to other users.

3478 ◦ `major (default)`: The document MUST be created as a major version

3479 ◦ `minor`: The document MUST be created as a minor version.

3480 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document  
3481 object.

3482 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document  
3483 object, either using the ACL from folderId if specified, or being applied if no folderId is specified.

3484 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3485 Document object, either using the ACL from folderId if specified, or being ignored if no folderId is  
3486 specified.

### 3487 2.2.4.1.2 Outputs

3488 **ID objectId:** The ID of the newly-created document.

### 3489 2.2.4.1.3 Exceptions Thrown & Conditions

3490 • See section 2.2.1.4.1 General Exceptions

3491 • `constraint`: The Repository MUST throw this exception if ANY of the following conditions are  
3492 met:

3493 ◦ The `cmis:objectId` property value is not an Object-Type whose baseType is  
3494 `"Document"`.

3495 ◦ The `cmis:objectId` property value is NOT in the list of AllowedChildObjectTypes of  
3496 the parent-folder specified by folderId.

3497 ◦ The value of any of the properties violates the min/max/required/length constraints  
3498 specified in the property definition in the Object-Type.

3499 ◦ The `"contentStreamAllowed"` attribute of the Object-Type definition specified by the  
3500 `cmis:objectId` property value is set to `"required"` and no contentStream input  
3501 parameter is provided.

- 3502 | ○ The `"_versionable"` attribute of the Object-Type definition specified by the  
3503 | `cmis:objectTypeId` property value is set to FALSE and a value for the `versioningState`  
3504 | input parameter is provided that is something other than `"_none"`.
- 3505 | ○ The `"_versionable"` attribute of the Object-Type definition specified by the  
3506 | `cmis:objectTypeId` property value is set to TRUE and the value for the `versioningState`  
3507 | input parameter is provided that is `"_none"`.
- 3508 | ○ The `"_controllablePolicy"` attribute of the Object-Type definition specified by the  
3509 | `cmis:objectTypeId` property value is set to FALSE and at least one policy is provided.
- 3510 | ○ The `"_controllableACL"` attribute of the Object-Type definition specified by the  
3511 | `cmis:objectTypeId` property value is set to FALSE and at least one ACE is provided.
- 3512 | ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3513 | repository.
- 3514 | • `nameConstraintViolation`: See section 2.2.1.4.2\_Specific Exceptions. If the repository  
3515 | detects a violation with the given `cmis:name` property value, the repository MAY throw this  
3516 | exception or chose a name which does not conflict.
- 3517 | • `storage`: See section 2.2.1.4.2\_Specific Exceptions.
  - 3518 | ○ The `"_controllableACL"` `streamNotSupported`: The Repository MUST throw this  
3519 | exception if the `"_contentStreamAllowed"` attribute of the Object-Type definition specified  
3520 | by the `cmis:objectTypeId` property value is set to `"_not allowed"` ~~FALSE and at least one~~  
3521 | ACE is provided.
  - 3522 | ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3523 | repository.
- 3524 | • ~~`nameConstraintViolation`: See section 2.2.1.4.2\_Specific Exceptions. If the repository~~  
3525 | ~~detects a violation with the given `cmis:name` property value, the repository MAY throw this~~  
3526 | ~~exception or chose a name which does not conflict.~~
- 3527 | • ~~`storage`: See section 2.2.1.4.2\_Specific Exceptions.~~
- 3528 | • ~~`streamNotSupported`: The Repository MUST throw this exception if the~~  
3529 | ~~`"_contentStreamAllowed"` attribute of the Object-Type definition specified by the `cmis:objectTypeId`~~  
3530 | ~~property value is set to `"_not allowed"` and a `contentStream` input parameter is provided.~~

## 3531 2.2.4.2 createDocumentFromSource

3532 **Description:** Creates a document object as a copy of the given source document in the (optionally)  
3533 specified location.

### 3534 2.2.4.2.1 Inputs

#### 3535 Required:

- 3536 | • **ID repositoryId:** The identifier for the Repository.
- 3537 | • **ID sourceId:** The identifier for the source document.

#### 3538 Optional:

- 3539 | • **<Array> properties:** The property values that MUST be applied to the Object. This list of  
3540 | properties SHOULD only contain properties whose values differ from the source document.
- 3541 | • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-  
3542 | created Document Object.
  - 3543 | ○ This parameter MUST be specified if the Repository does NOT support the optional  
3544 | `"_unfiling"` capability.
- 3545 | • **Enum versioningState:** An enumeration specifying what the `versioning` state of the newly-  
3546 | created object MUST be. If the repository does not support versioning, the repository MUST  
3547 | ignore the `versioningState` parameter. Valid values are:

- 3548           o none: The document MUST be created as a non-versionable document.
- 3549           o checkedout: The document MUST be created in the checked-out state.
- 3550           o major (default): The document MUST be created as a major version
- 3551           o minor: The document MUST be created as a minor version.
- 3552       • <Array> policies: A list of policy IDs that MUST be applied to the newly-created Document
- 3553       object.
- 3554       • <Array> ACE addACEs: A list of ACEs that MUST be added to the newly-created Document
- 3555       object, either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3556       • <Array> ACE removeACEs: A list of ACEs that MUST be removed from the newly-created
- 3557       Document object, either using the ACL from folderId if specified, or being ignored if no folderId is
- 3558       specified.

#### 3559 2.2.4.2.2 Outputs

3560       **ID objectId:** The ID of the newly-created document.

#### 3561 2.2.4.2.3 Exceptions Thrown & Conditions

- 3562       • See section 2.2.1.4.1 General Exceptions
- 3563       • constraint: The Repository MUST throw this exception if ANY of the following conditions are
- 3564       met:
  - 3565           o The sourceId is not an Object whose baseType is `"Document"`.
  - 3566           o The source document's cmis:objectId property value is NOT in the list of
  - 3567           AllowedChildObjectTypes of the parent-folder specified by folderId.
  - 3568           o The `"versionable"` attribute of the Object-Type definition specified by the
  - 3569           cmis:objectId property value is set to FALSE and a value for the versioningState
  - 3570           input parameter is provided that is something other than `"none"`.
  - 3571           o The `"versionable"` attribute of the Object-Type definition specified by the
  - 3572           cmis:objectId property value is set to TRUE and the value for the versioningState
  - 3573           input parameter is provided that is `"none"`.
  - 3574           o The `"controllablePolicy"` attribute of the Object-Type definition specified by the
  - 3575           cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3576           o The `"controllableACL"` attribute of the Object-Type definition specified by the
  - 3577           cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3578           o At least one of the permissions is used in an ACE provided which is not supported by the
  - 3579           repository.
- 3580       • nameConstraintViolation: See section 2.2.1.4.2 Specific Exceptions. If the repository
- 3581       detects a violation with the given cmis:name property value, the repository MAY throw this
- 3582       exception or chose a name which does not conflict.
- 3583       • storage: See section 2.2.1.4.2 Specific Exceptions.
- 3584       • streamNotSupported: The Repository MUST throw this exception if the
- 3585       `"contentStreamAllowed"` attribute of the Object-Type definition specified by the
- 3586       cmis:objectId property value is set to `"not allowed"` and a contentStream input parameter is
- 3587       provided.

#### 3588 2.2.4.3 createFolder

3589       **Description:** Creates a folder object of the specified type in the specified location.

### 3590 2.2.4.3.1 Inputs

#### 3591 Required:

- 3592 • **ID repositoryId:** The identifier for the Repository.
- 3593 • **<Array> properties:** The property values that MUST be applied to the newly-created Folder  
3594 Object.
- 3595 • **ID folderId:** The identifier for the folder that MUST be the parent folder for the newly-created  
3596 Folder Object.

#### 3597 Optional:

- 3598 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Folder object.
- 3599 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Folder object,  
3600 either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3601 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3602 Folder object, either using the ACL from folderId if specified, or being ignored if no folderId is  
3603 specified.

### 3604 2.2.4.3.2 Outputs

- 3605 • **ID objectId:** The ID of the newly-created folder.

### 3606 2.2.4.3.3 Exceptions Thrown & Conditions

- 3607 • See section 2.2.1.4.1 General Exceptions
- 3608 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are  
3609 met:
  - 3610 ○ The cmis:objectId property value is not an Object-Type whose baseType is  
3611 `"Folder"`.
  - 3612 ○ The value of any of the properties violates the min/max/required/length constraints  
3613 specified in the property definition in the Object-Type.
  - 3614 ○ The cmis:objectId property value is NOT in the list of AllowedChildObjectTypes of  
3615 the parent-folder specified by folderId.
  - 3616 ○ The `"controllablePolicy"` attribute of the Object-Type definition specified by the  
3617 cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3618 ○ The `"controllableACL"` attribute of the Object-Type definition specified by the  
3619 cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3620 ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3621 repository.
- 3622 • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository  
3623 detects a violation with the given cmis:name property value, the repository MAY throw this  
3624 exception or chose a name which does not conflict.
- 3625 • **storage:** See section 2.2.1.4.2 Specific Exceptions.

### 3626 2.2.4.4 createRelationship

3627 **Description:** Creates a relationship object of the specified type

#### 3628 2.2.4.4.1 Inputs

##### 3629 Required:

- 3630 • **ID repositoryId:** The identifier for the Repository.

- 3631 | • **<Array> properties:** The property values that MUST be applied to the newly-created Relationship  
3632 | Object.

3633 | **Optional:**

- 3634 | • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Relationship  
3635 | object.
- 3636 | • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Relationship  
3637 | object, either using the ACL from folderId if specified, or being applied if no folderId is specified.  
3638 | **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3639 | Relationship object, either using the ACL from folderId if specified, or being ignored if no folderId  
3640 | is specified.

3641 | **2.2.4.4.2 Outputs**

- 3642 | • **ID objectId:** The ID of the newly-created relationship.

3643 | **2.2.4.4.3 Exceptions Thrown & Conditions**

- 3644 | • See section 2.2.1.4.1 General Exceptions
- 3645 | • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are  
3646 | met:
- 3647 | ○ The cmis:objectId property value is not an Object-Type whose baseType is  
3648 | `"Relationship"`.
  - 3649 | ○ The value of any of the properties violates the min/max/required/length constraints  
3650 | specified in the property definition in the Object-Type.
  - 3651 | ○ The sourceObjectId's Object-Type is not in the list of `"allowedSourceTypes"` specified by  
3652 | the Object-Type definition specified by cmis:objectId property value.
  - 3653 | ○ The targetObjectId's Object-Type is not in the list of `"allowedTargetTypes"` specified by  
3654 | the Object-Type definition specified by cmis:objectId property value.
  - 3655 | ○ The `"controllablePolicy"` attribute of the Object-Type definition specified by the  
3656 | cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3657 | ○ The `"controllableACL"` attribute of the Object-Type definition specified by the  
3658 | cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3659 | ○ At least one of the permissions is used in an ACE provided which is not supported by the  
3660 | repository.
- 3661 | • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository  
3662 | detects a violation with the given cmis:name property value, the repository MAY throw this  
3663 | exception or chose a name which does not conflict.
- 3664 | • **storage:** See section 2.2.1.4.2 Specific Exceptions.

3665 | **2.2.4.5 createPolicy**

3666 | **Description:** Creates a policy object of the specified type

3667 | **2.2.4.5.1 Inputs**

3668 | **Required:**

- 3669 | • **ID repositoryId:** The identifier for the Repository.
- 3670 | • **<Array> properties:** The property values that MUST be applied to the newly-created Policy  
3671 | Object.

3672 | **Optional:**

- 3673 | • **ID folderId:** If specified, the identifier for the folder that MUST be the parent folder for the newly-  
3674 | created Policy Object.
- 3675 |     • This parameter MUST be specified if the Repository does NOT support the optional  
3676 |     "unfiling" capability.
- 3677 | • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Policy object.
- 3678 | • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Policy object,  
3679 | either using the ACL from folderId if specified, or being applied if no folderId is specified.
- 3680 | • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
3681 | Policy object, either using the ACL from folderId if specified, or being ignored if no folderId is  
3682 | specified.

### 3683 | 2.2.4.5.2 Outputs

- 3684 | • **ID objectId:** The ID of the newly-created Policy Object.

### 3685 | 2.2.4.5.3 Exceptions Thrown & Conditions

- 3686 | • See section 2.2.1.4.1 General Exceptions
- 3687 | • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are  
3688 | met:
  - 3689 |     • The cmis:objectId property value is not an Object-Type whose baseType is  
3690 |     "Policy".
  - 3691 |     • The value of any of the properties violates the min/max/required/length constraints  
3692 |     specified in the property definition in the Object-Type.
  - 3693 |     • The cmis:objectId property value is NOT in the list of AllowedChildObjectIds of  
3694 |     the parent-folder specified by folderId.
  - 3695 |     • The "controllablePolicy" attribute of the Object-Type definition specified by the  
3696 |     cmis:objectId property value is set to FALSE and at least one policy is provided.
  - 3697 |     • The "controllableACL" attribute of the Object-Type definition specified by the  
3698 |     cmis:objectId property value is set to FALSE and at least one ACE is provided.
  - 3699 |     • At least one of the permissions is used in an ACE provided which is not supported by the  
3700 |     repository.
- 3701 | • **nameConstraintViolation:** See section 2.2.1.4.2 Specific Exceptions. If the repository  
3702 | detects a violation with the given cmis:name property value, the repository MAY throw this  
3703 | exception or chose a name which does not conflict.
- 3704 | • **storage:** See section 2.2.1.4.2 Specific Exceptions.

### 3705 | 2.2.4.6 getAllowableActions

3706 | **Description:** Gets the list of allowable actions for an Object (see section.2.2.1.2.6 Allowable Actions).

#### 3707 | 2.2.4.6.1 Inputs

3708 | **Required:**

- 3709 | • **ID repositoryId:** The identifier for the Repository.
- 3710 | • **ID objectId:** The identifier for the object

#### 3711 | 2.2.4.6.2 Outputs

- 3712 | • **<Array> AllowableActions:** see section 2.2.1.2.6 Allowable Actions.

3713 **2.2.4.6.3 Exceptions Thrown & Conditions**

3714 See section 2.2.1.4.1 General Exceptions

3715 **2.2.4.7 getObject**

3716 **Description:** Gets the specified information for the Object.

3717 **2.2.4.7.1 Inputs**

3718 **Required:**

- 3719 • **ID repositoryId:** The identifier for the Repository.
- 3720 • **ID objectId:** The identifier for the object

3721 **Optional:**

- 3722 • **String filter:** See section 2.2.1.2.1 Properties.
- 3723 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 3724 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 3725 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 3726 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 3727 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3728 **2.2.4.7.2 Outputs**

- 3729 • **<Array> Properties:** The list of properties for the object.
- 3730 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 3731 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 3732 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 3733 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 3734 • **<Array> AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

3735 **2.2.4.7.3 Exceptions Thrown & Conditions**

3736 See section 2.2.1.4.1 General Exceptions

3737 **filterNotValid:** The Repository MUST throw this exception if the se property or rendition filter  
3738 input parameter is not valid.

3739 **2.2.4.8 getProperties**

3740 **Description:** Gets the list of properties for an Object.

3741 **2.2.4.8.1 Inputs**

3742 **Required:**

- 3743 • **ID repositoryId:** The identifier for the Repository.
- 3744 • **ID objectId:** The identifier for the object

3745 **Optional:**

- 3746 • **String filter:** See section 2.2.1.2.1 Properties.

3747 **2.2.4.8.2 Outputs**

3748 **<Array> Properties:** The list of properties for the object.

3749 **2.2.4.8.3 Exceptions Thrown & Conditions**

3750 See section 2.2.1.4.1 General Exceptions

3751 `filterNotValid`: The Repository MUST throw this exception if this property filter input parameter  
3752 is not valid.

3753 **2.2.4.9 getObjectByPath**

3754 **Description:** Gets the specified object.

3755 **2.2.4.9.1 Inputs**

3756 **Required:**

- 3757 • **ID repositoryId**: The identifier for the Repository.
- 3758 • **String path**: The path to the object. See section 2.1.5.3 Paths.

3759 **Optional:**

- 3760 • **String filter**: See section 2.2.1.2.1 Properties.
- 3761 • **Boolean includeAllowableActions**: See section 2.2.1.2.6 Allowable Actions.
- 3762 • **Enum includeRelationships**: See section 2.2.1.2.2 Relationships.
- 3763 • **String renditionFilter**: See section 2.2.1.2.4 Renditions.
- 3764 • **Boolean includePolicyIds**: See section 2.2.1.2.2 Relationships.
- 3765 • **Boolean includeACL**: See section 2.2.1.2.5 ACLs.

3766 **2.2.4.9.2 Outputs**

- 3767 • **<Array> Properties**: The list of properties for the object.
- 3768 • **AllowableActions**: See section 2.2.1.2.6 Allowable Actions.

3769 **2.2.4.9.3 Exceptions Thrown & Conditions**

- 3770 • See section 2.2.1.4.1 General Exceptions

3771 • `filterNotValid`: The Repository MUST throw this exception if this property or rendition filter  
3772 input parameter is not valid.

3773 **2.2.4.10 getContentStream**

3774 **Description:** Gets the content stream for the specified Document object, or gets a rendition stream for a  
3775 specified rendition of a document or folder object.

3776 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,  
3777 in a manner appropriate to that protocol.

3778 **2.2.4.10.1 Inputs**

3779 **Required:**

- 3780 • **ID repositoryId**: The identifier for the Repository.
- 3781 • **ID objectId**: The identifier for the object

3782 **Optional:**

- 3783 • **ID streamId**: The identifier for the rendition stream, when used to get a rendition stream. For  
3784 Documents, if not provided then this method returns the content stream. For Folders, it MUST be  
3785 provided.

3786 **2.2.4.10.2 Outputs**

- 3787 |
  - **<Stream> ContentStream:** The specified content stream or rendition stream for the object.

3788 **2.2.4.10.3 Exceptions Thrown & Conditions**

- 3789 |
  - See section 2.2.1.4.1 General Exceptions

3790 | `constraint:` The Repository MUST throw this exception if the object specified by `objectId` does  
3791 | NOT have a content stream or rendition stream.

3792 **2.2.4.11 getRenditions**

3793 **Description:** Gets the list of associated Renditions for the specified object. Only rendition attributes are  
3794 | returned, not rendition stream.

3795 **Notes:** Each CMIS protocol binding MAY provide a way for fetching a sub-range within a content stream,  
3796 | in a manner appropriate to that protocol.

3797 **2.2.4.11.1 Inputs**

3798 **Required:**

- 3799 |
  - **ID repositoryId:** The identifier for the Repository.
  - **ID objectId:** The identifier for the object

3801 **Optional:**

- 3802 |
  - **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.

3805 **2.2.4.11.2 Outputs**

- 3806 |
  - **<Array> Renditions:** The set of renditions available on this object

3807 **2.2.4.11.3 Exceptions Thrown & Conditions**

- 3808 |
  - See section 2.2.1.4.1 General Exceptions
  - `notSupported:` The service method requires functionality that is not supported by the  
3809 | repository
  - `filterNotValid:` The rendition filter specified is not valid

3812 **2.2.4.12 updateProperties**

3813 **Description:** Updates properties of the specified object.

3814 **Notes:**

- 3815 |
  - A Repository MAY automatically create new Document versions as part of an update properties  
3816 | 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  
3817 | include all updatable properties, or only those properties whose values are different than the  
3818 | original value of the object.

3820 **2.2.4.12.1 Inputs**

3821 **Required:**

- 3822 |
  - **ID repositoryId:** The identifier for the Repository.

3823 | **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:objectId property value of the given object is NOT in the list of AllowedChildObjectTypes of the parent-folder specified by targetFolderId.

- 3863 | • `nameConstraintViolation`: See section 2.2.1.4.2 Specific Exceptions. The repository MAY  
3864 | throw this exception or chose a name which does not conflict.
- 3865 | • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 3866 | • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.
- 3867 | • `versioning`: The repository MAY throw this exception if the object is a non-current Document  
3868 | Version.

## 3869 | 2.2.4.14 deleteObject

3870 | **Description:** Deletes the specified object.

### 3871 | 2.2.4.14.1 Inputs

#### 3872 | **Required:**

- 3873 | • **ID repositoryId:** The identifier for the Repository.
- 3874 | • **ID objectId:** The identifier of the object to be deleted.

#### 3875 | **Optional:**

- 3876 | • **Boolean allVersions:** If TRUE (default), then delete all versions of the document. If FALSE,  
3877 | delete only the document object specified. The Repository MUST ignore the value of this  
3878 | parameter when this service is invoke on a non-document object or non-versionable document  
3879 | object.

3880 |

### 3881 | 2.2.4.14.2 Exceptions Thrown & Conditions

- 3882 | • See section 2.2.1.4.1 General Exceptions
- 3883 | • `constraint`: The Repository MUST throw this exception if the method is invoked on a Folder  
3884 | object that contains one or more objects.
- 3885 | • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.

## 3886 | 2.2.4.15 deleteTree

3887 | **Description:** Deletes the specified folder object and all of its child- and descendant-objects.

#### 3888 | **Notes:**

- 3889 | • A Repository MAY attempt to delete child- and descendant-objects of the specified folder in any  
3890 | order.
- 3891 | • Any child- or descendant-object that the Repository cannot delete MUST persist in a valid state in  
3892 | the CMIS domain model.
- 3893 | • This is not atomic.
- 3894 | • However, if `deletesinglefiled` is chosen and some objects fail to delete, then single-filed objects  
3895 | are either deleted or kept, never just unfiled. This is so that a user can call this command again to  
3896 | recover from the error by using the same tree.

### 3897 | 2.2.4.15.1 Inputs

#### 3898 | **Required:**

- 3899 | • **ID repositoryId:** The identifier for the Repository.
- 3900 | • **ID folderId:** The identifier of the folder to be deleted.

#### 3901 | **Optional:**

- 3902 | • **Boolean allVersions:** If TRUE (default), then delete all versions of the document. If FALSE, delete  
3903 | only the document object specified. The Repository MUST ignore the value of this parameter when  
3904 | this service is invoke on a non-document object or non-versionable document object.
- 3905 | • **Enum unfileObjects:** An enumeration specifying how the repository MUST process file-able  
3906 | child- or descendant-objects. Valid values are:
  - 3907 | ○ **unfile:** Unfile all fileable objects.
  - 3908 | ○ **deletesinglefiled:** Delete all fileable non-folder objects whose only parent-folders are in  
3909 | the current folder tree. Unfile all other fileable non-folder objects from the current folder tree.
  - 3910 | ○ **delete (default):** Delete all fileable objects.
- 3911 | • **boolean continueOnFailure:** If TRUE, then the repository SHOULD continue attempting to perform  
3912 | this operation even if deletion of a child- or descendant-object in the specified folder cannot be  
3913 | deleted.
  - 3914 | ○ If FALSE (**default**), then the repository SHOULD abort this method when it fails to delete a  
3915 | single child- or descendant-object.

### 3916 | 2.2.4.15.2 Outputs

- 3917 | • **<Array> ID failedToDelete:** A list of identifiers of objects in the folder tree that were not deleted.

### 3918 | 2.2.4.15.3 Exceptions Thrown & Conditions

- 3919 | • See section 2.2.1.4.1 General Exceptions
- 3920 | • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

### 3921 | 2.2.4.16 setContentStream

3922 | **Description:** Sets the content stream for the specified Document object.

3923 | **Notes:** A Repository MAY automatically create new Document versions as part of this service method.  
3924 | Therefore, the objectId output NEED NOT be identical to the objectId input.

#### 3925 | 2.2.4.16.1 Inputs

3926 | **Required:**

- 3927 | • **ID repositoryId:** The identifier for the Repository.
- 3928 | • **ID objectId:** The identifier for the Document object.
- 3929 | • **<contentStream> contentStream:** The Content Stream

3930 | **Optional:**

- 3931 | • **Boolean overwriteFlag:** If TRUE (**default**), then the Repository MUST replace the existing  
3932 | content stream for the object (if any) with the input contentStream.
  - 3933 | ○ If FALSE, then the Repository MUST only set the input contentStream for the object if the  
3934 | object currently does not have a content-stream.
- 3935 | • **String changeToken:** See section 2.2.1.3 Change Tokens.

#### 3936 | 2.2.4.16.2 Outputs

- 3937 | • **ID objectId:** The ID of the document.
- 3938 | • **String changeToken:** See section 2.2.1.3 Change Tokens.

#### 3939 | 2.2.4.16.3 Exceptions Thrown & Conditions

- 3940 | • See section 2.2.1.4.1 General Exceptions

- 3941 • `contentAlreadyExists`: The Repository MUST throw this exception if the input parameter  
3942 `overwriteFlag` is FALSE and the Object already has a content-stream.
- 3943 • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 3944 • `streamNotSupported`: The Repository MUST throw this exception if the  
3945 `"_contentStreamAllowed"` attribute of the Object-Type definition specified by the  
3946 `cmis:objectTypeId` property value of the given document is set to `"_notallowed"`.
- 3947 • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.
- 3948 • `versioning`: The repository MAY throw this exception if the object is a non-current Document  
3949 Version.

## 3950 2.2.4.17 deleteContentStream

3951 **Description:** Deletes the content stream for the specified Document object.

3952 **Notes:** A Repository MAY automatically create new Document versions as part of this service method.  
3953 Therefore, the `objectId` output NEED NOT be identical to the `objectId` input.

### 3954 2.2.4.17.1 Inputs

3955 **Required:**

- 3956 • **ID repositoryId**: The identifier for the Repository.
- 3957 • **ID objectId**: The identifier for the Document object.

3958 **Optional:**

- 3959 • **String changeToken**: See section 2.2.1.3 Change Tokens.

### 3960 2.2.4.17.2 Outputs

- 3961 • **ID objectId**: The ID of the Document object.
- 3962 • **String changeToken**: See section 2.2.1.3 Change Tokens.

### 3963 2.2.4.17.3 Exceptions Thrown & Conditions

- 3964 • See section 2.2.1.4.1 General Exceptions
- 3965 • `constraint`: The Repository MUST throw this exception if the Object's Object-Type definition  
3966 `"_contentStreamAllowed"` attribute is set to `"_required"`.
- 3967 • `storage`: See section 2.2.1.4.2 Specific Exceptions.
- 3968 • `updateConflict`: See section 2.2.1.4.2 Specific Exceptions.
- 3969 • `versioning`: The repository MAY throw this exception if the object is a non-current Document  
3970 Version.

## 3971 2.2.5 Multi-filing Services

3972 The Multi-filing services (*addObjectToFolder*, *removeObjectFromFolder*) are supported only if the  
3973 repository supports the multifiling or unfiling ~~optional capabilities~~ optional capabilities. The Multi-filing  
3974 Services are used to file/un-file objects into/from folders.

3975 This service is NOT used to create or delete objects in the repository.

### 3976 2.2.5.1 addObjectToFolder

3977 **Description:** Adds an existing fileable non-folder object to a folder.

3978 **2.2.5.1.1 Inputs**

3979 **Required:**

- 3980 • **ID repositoryId:** The identifier for the Repository.
- 3981 • **ID objectId:** The identifier of the object.
- 3982 • **ID folderId:** The folder into which the object is to be filed.

3983 **Optional:**

- 3984 • **1. Boolean allVersions:** Add all versions of the object to the folder if the repository supports  
3985 version-specific filing. Defaults to TRUE.

3986 **2.2.5.1.2 Exceptions Thrown & Conditions**

- 3987 • See section 2.2.1.4.1 General Exceptions.
- 3988 • **constraint:** The Repository MUST throw this exception if the `cmis:objectId` property value  
3989 of the given object is NOT in the list of `AllowedChildObjectTypes` of the parent-folder specified  
3990 by `folderId`.

3991 **2.2.5.2 removeObjectFromFolder**

3992 **Description:** Removes an existing fileable non-folder object from a folder.

3993 **2.2.5.2.1 Inputs**

3994 **Required:**

- 3995 • **ID repositoryId:** The identifier for the Repository.
- 3996 • **ID objectId:** The identifier of the object.

3997 **Optional:**

- 3998 • **ID folderId:** The folder from which the object is to be removed.
  - 3999 ○ If no value is specified, then the Repository MUST remove the object from all folders in which  
4000 it is currently filed.

4001 **2.2.5.2.2 Exceptions Thrown & Conditions**

- 4002 • See section 2.2.1.4.1 General Exceptions

4003 **2.2.6 Discovery Services**

4004 The Discovery Services (*query*) are used to search for query-able objects within the Repository.

4005 **2.2.6.1 query**

4006 **Description:** Executes a CMIS query statement against the contents of the Repository.

4007 **2.2.6.1.1 Inputs**

4008 **Required:**

- 4009 • **ID repositoryId:** The identifier for the Repository.
- 4010 • **String statement:** CMIS query to be executed. (See section 2.1.10 Query.)

4011 **Optional:**

- 4012 • **Boolean searchAllVersions:**
  - 4013 ○ If TRUE, then the Repository MUST include latest and non-latest versions of document  
4014 objects in the query search scope.

- 4015 |           ○ If FALSE (**default**), then the Repository MUST only include latest versions of documents
- 4016 |           in the query search scope.
- 4017 |           ○ If the Repository does not support the optional `capabilityAllVersionsSearchable`
- 4018 |           capability, then this parameter value MUST be set to FALSE.
- 4019 |       • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4020 |           ○ Note: For query statements where the SELECT clause contains properties from only one
- 4021 |           virtual table reference (i.e. referenced object-type), any value for this enum may be used.
- 4022 |           If the SELECT clause contains properties from more than one table, then the value of this
- 4023 |           parameter MUST be `"none"`.
- 4024 |       • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4025 |           ○ If the SELECT clause contains properties from more than one table, then the value of this
- 4026 |           parameter MUST not be set.
- 4027 |       • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4028 |           ○ Note: For query statements where the SELECT clause contains properties from only one
- 4029 |           virtual table reference (i.e. referenced object-type), any value for this parameter may be
- 4030 |           used. If the SELECT clause contains properties from more than one table, then the value
- 4031 |           of this parameter MUST be `"FALSE"`.
- 4032 |       • **Integer maxItems:** See section 2.2.1.1 Paging.
- 4033 |       • **Integer skipCount:** See section 2.2.1.1 Paging.

#### 4034 | 2.2.6.1.2 Outputs

- 4035 |       • **<Array> Object QueryResults:** The set of results for the query. (See section 2.1.10 Query.).
- 4036 |       Each object result MUST include the following elements if they are requested:
  - 4037 |           ○ **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
  - 4038 |           ○ **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
  - 4039 |           ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4040 |       • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

#### 4041 | Optional:

- 4042 |       • **Integer numItems:** See section 2.2.1.1 Paging.

#### 4044 | 2.2.6.1.3 Exceptions Thrown & Conditions

- 4045 |       • See section 2.2.1.4.1 General Exceptions
- 4046 |       • If the select clause includes properties from more than a single type reference, then the
- 4047 |       repository SHOULD throw an exception if includeRelationships is something other than `"none"`
- 4048 |       or includeAllowableActions is specified as TRUE.

#### 4049 | 2.2.6.2 getContentChanges

4050 | **Description:** Gets a list of content changes. This service is intended to be used by search crawlers or  
 4051 | other applications that need to efficiently understand what has changed in the repository.

#### 4052 | Notes:

- 4053 |       • The content stream is NOT returned for any change event.
- 4054 |       • The definition of the authority needed to call this service is repository specific.
- 4055 |       • The latest change log token for a repository can be acquired via the `getRepositoryInfo` service.

### 4056 2.2.6.2.1 Inputs

#### 4057 Required:

- 4058 • **ID repositoryId:** The identifier for the Repository.

#### 4059 Optional:

- 4060 • **String changeLogToken:**

- 4061 ○ If specified, then the Repository MUST return the change event corresponding to the  
4062 value of the specified change log token as the first result in the output.
- 4063 ○ If not specified, then the Repository MUST return the first change event recorded in the  
4064 change log.

- 4065 • **Boolean includeProperties:**

- 4066 ○ If TRUE, then the Repository MUST include the updated property values for `"_updated"`  
4067 change events if the repository supports returning property values as specified by  
4068 `capabilityChanges`.
- 4069 ○ If FALSE (default), then the Repository MUST NOT include the updated property values  
4070 for `"_updated"` change events. The single exception to this is that the `objectId` MUST  
4071 always be included.

- 4072 • **Boolean includePolicyIds:**

4073 If TRUE, then the Repository MUST include the IDs of Policies applied to the object referenced in  
4074 each change event, if the change event modified the set of policies applied to the object.

4075 If FALSE (default), then the Repository will not include policy information.

- 4076 • **String filter:** See section 2.2.1.2.1 Properties. The service will only return the properties in the  
4077 matched object if they exist on the matched object type definition and in the filter.
- 4078 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4079 • **Integer maxItems:** See section 2.2.1.1 Paging.

### 4080 2.2.6.2.2 Outputs

- 4081 • **<Array> changeEvents:** A collection of CMIS objects that MUST include the information **as**  
4082 ~~specified in 2.1.11.3~~ as specified in 2.1.11.3. Each result MUST include the following elements if  
4083 they are requested:

- 4084 ○ **<Array> policyIDs:** The IDs of Policies applied to the object referenced in the change  
4085 event.

- 4086 ○ **<Array> ACLs:** The ACLs applied to the object reference in the change event.

- 4087 • **String latestChangeLogToken:** The change log token corresponding to the last change event in  
4088 `changeEvents`.
- 4089 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

#### 4090 Optional:

- 4091 • **Integer numItems:** See section 2.2.1.1 Paging.

### 4092 2.2.6.2.3 Exceptions Thrown & Conditions

- 4093 • See section 2.2.1.4.1 General Exceptions
- 4094 • **constraint:** The Repository MUST throw this exception if the event corresponding to the  
4095 change log token provided as an input parameter is no longer available in the change log. (E.g.  
4096 because the change log was truncated).

## 4097 2.2.7 Versioning Services

4098 The Versioning services (checkOut, cancelCheckOut, getPropertiesOfLatestVersion, getAllVersions,  
4099 deleteAllVersions) are used to navigate or update a Document Version Series.

### 4100 2.2.7.1 checkOut

4101 **Description:** Create a private working copy of the document.

#### 4102 2.2.7.1.1 Inputs

4103 **Required:**

- 4104 • **ID repositoryId:** The identifier for the Repository.
- 4105 • **ID objectId:** The identifier of the document version.

#### 4106 2.2.7.1.2 Outputs

- 4107 • **ID objectId:** The identifier for the `"Private Working Copy"` document.
- 4108 • **Boolean contentCopied:** TRUE if the content-stream of the Private Working Copy is a copy of  
4109 the contentStream of the Document that was checked out.
- 4110 • **Boolean contentCopied:** FALSE if the content-stream of the Private Working Copy is `"not set"`.

#### 4111 2.2.7.1.3 Exceptions Thrown & Conditions

- 4112 • See section 2.2.1.4.1 General Exceptions
- 4113 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type  
4114 definition's `versionable` attribute is FALSE.
- 4115 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 4116 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 4117 • **versioning:** The repository MAY throw this exception if the object is a non-current Document  
4118 Version.

### 4119 2.2.7.2 cancelCheckOut

4120 **Description:** Reverses the effect of a check-out. Removes the private working copy of the checked-out  
4121 document, allowing other documents in the version series to be checked out again. If the private working  
4122 copy has been created by createDocument, cancelCheckOut MUST delete the created document.

#### 4123 2.2.7.2.1 Inputs

4124 **Required:**

- 4125 • **ID repositoryId:** The identifier for the Repository.
- 4126 • **ID objectId:** The identifier of the Private Working Copy.

#### 4127 2.2.7.2.2 Exceptions Thrown & Conditions

- 4128 • See section 2.2.1.4.1 General Exceptions
- 4129 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type  
4130 definition's `versionable` attribute is FALSE.
- 4131 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.
- 4132 • **versioning:** The repository MAY throw this exception if the object is a non-current Document  
4133 Version.

### 4134 2.2.7.3 checkIn

4135 **Description:** Checks-in the Private Working Copy document.

4136 **Notes:**

- 4137 • For repositories that do NOT support the **optional** `"capabilityPWCUpdatable"`  
4138 `capability, optional "capabilityPWCUpdatable" capability`, the *properties* and *contentStream*  
4139 input parameters MUST be provided on the checkIn method for updates to happen as part of  
4140 checkIn.
- 4141 • Each CMIS protocol bindings MUST specify whether the checkin service MUST always include all  
4142 updatable properties, or only those properties whose values are different than the original value  
4143 of the object.

### 4144 2.2.7.3.1 Inputs

4145 **Required:**

- 4146 • **ID repositoryId:** The identifier for the Repository.
- 4147 • **ID objectId:** The identifier of the document.

4148 **Optional:**

- 4149 • **Boolean major:** TRUE (**default**) if the checked-in Document Object MUST be a major version.  
4150 ○ FALSE if the checked-in Document Object MUST NOT be a major version.
- 4151 • **<Array> properties:** The property values that MUST be applied to the checked-in Document  
4152 Object.
- 4153 • **<contentStream> contentStream:** The Content Stream that MUST be stored for the checked-in  
4154 Document Object. The method of passing the contentStream to the server and the encoding  
4155 mechanism will be specified by each specific binding.
- 4156 • **String checkInComment:** See section 2.1.9.5 Versioning Properties on Document Objects.
- 4157 • **<Array> policies:** A list of policy IDs that MUST be applied to the newly-created Document  
4158 object.
- 4159 • **<Array> ACE addACEs:** A list of ACEs that MUST be added to the newly-created Document  
4160 object.
- 4161 • **<Array> ACE removeACEs:** A list of ACEs that MUST be removed from the newly-created  
4162 Document object.

### 4163 2.2.7.3.2 Outputs

4164 **ID objectId:** The ID of the checked-in document.

### 4165 2.2.7.3.3 Exceptions Thrown & Conditions

- 4166 • See section 2.2.1.4.1 General Exceptions
- 4167 • **constraint:** The Repository MUST throw this exception if the Document's Object-Type  
4168 definition's *versionable* attribute is FALSE.
- 4169 • **storage:** See section 2.2.1.4.2 Specific Exceptions.
- 4170 • **streamNotSupported:** The Repository MUST throw this exception if the  
4171 `"contentStreamAllowed"` attribute of the Object-Type definition specified by the  
4172 `cmis:objectId` property value is set to `"not allowed"` and a `contentStream` input parameter is  
4173 provided.
- 4174 • **updateConflict:** See section 2.2.1.4.2 Specific Exceptions.

4175 **2.2.7.4 getObjectOfLatestVersion**

4176 **Description:** Get a the latest Document object in the Version Series.

4177 **2.2.7.4.1 Inputs**

4178 **Required:**

- 4179 • **ID repositoryId:** The identifier for the Repository.
- 4180 • **ID objectId:** The identifier for the Version Series.

4181 **Optional:**

- 4182 • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major  
4183 version object in the Version Series.
  - 4184 ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-  
4185 major) version object in the Version Series.
- 4186 • **String filter:** See section 2.2.1.2.1 Properties.
- 4187 • **Enum includeRelationships:** See section 2.2.1.2.2 Relationships.
- 4188 • **Boolean includePolicyIds:** See section 2.2.1.2.3 Policies.
- 4189 • **String renditionFilter:** See section 2.2.1.2.4 Renditions.
- 4190 • **Boolean includeACL:** See section 2.2.1.2.5 ACLs.
- 4191 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4192 **2.2.7.4.2 Outputs**

- 4193 • **<Array> Properties:** The list of properties for the object.
- 4194 • **<Array> Relationships:** See section 2.2.1.2.2 Relationships.
- 4195 • **<Array> Policy Ids:** See section 2.2.1.2.3 Policies.
- 4196 • **<Array> Renditions:** See section 2.2.1.2.4 Renditions.
- 4197 • **<Array> ACLs:** See section 2.2.1.2.5 ACLs.
- 4198 • **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.

4199 **2.2.7.4.3 Exceptions Thrown & Conditions**

- 4200 • See section 2.2.1.4.1 General Exceptions
- 4201 • **filterNotValid:** The Repository MUST throw this exception if this property or rendition filter  
4202 input parameter is not valid.
- 4203 • **objectNotFound:** The Repository MUST throw this exception if the input parameter major is  
4204 TRUE and the Version Series contains no major versions.

4205 **2.2.7.5 getPropertiesOfLatestVersion**

4206 **Description:** Get a subset of the properties for the latest Document Object in the Version Series.

4207 **2.2.7.5.1 Inputs**

4208 **Required:**

- 4209 • **ID repositoryId:** The identifier for the Repository.
- 4210 • **ID objectId:** The identifier for the Version Series.

4211 **Optional:**

- 4212 | • **Boolean major:** If TRUE, then the Repository MUST return the properties for the latest major version  
4213 | object in the Version Series.
- 4214 |     ○ If FALSE (**default**), the Repository MUST return the properties for the latest (major or non-  
4215 |     major) version object in the Version Series.
- 4216 | • **String filter:** See section 2.2.1.2.1 Properties.

## 4217 | 2.2.7.5.2 Outputs

4218 |     **<Array> Properties:** The list of properties for the object.

## 4219 | 2.2.7.5.3 Exceptions Thrown & Conditions

- 4220 |     • See section 2.2.1.4.1 General Exceptions
- 4221 |     • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
4222 |     parameter is not valid.
- 4223 |     • **objectNotFound:** The Repository MUST throw this exception if the input parameter major is  
4224 |     TRUE and the Version Series contains no major versions.

## 4225 | 2.2.7.6 getAllVersions

4226 | **Description:** Returns the list of all Document Objects in the specified Version Series, sorted by  
4227 | cmis:creationDate descending.

4228 | **Notes:**

- 4229 |     • The result set for this operation MUST include the Private Working Copy, subject to caller's  
4230 |     access privileges.

## 4231 | 2.2.7.6.1 Inputs

4232 | **Required:**

- 4233 |     • **ID repositoryId:** The identifier for the Repository.
- 4234 |     • **ID objectId:** The identifier for the Version Series.

4235 | **Optional:**

- 4236 |     • **String filter:** See section 2.2.1.2.1 Properties.
- 4237 |     • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

## 4238 | 2.2.7.6.2 Outputs

- 4239 |     • **<Array> ObjectResults:** A list of Document Objects in the specified Version Series. Each object  
4240 |     result MUST include the following elements if they are requested:
- 4241 |         ○ **<Array> Properties:** The list of properties for the object.
- 4242 |         ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4243 |

## 4244 | 2.2.7.6.3 Exceptions Thrown & Conditions

- 4245 |     • See section 2.2.1.4.1 General Exceptions
- 4246 |     • **filterNotValid:** The Repository MUST throw this exception if this property filter input  
4247 |     parameter is not valid.

## 4248 2.2.8 Relationship Services

4249 The Relationship Services (*getObjectRelationships*) are used to retrieve the dependent Relationship  
4250 objects associated with an independent object.

### 4251 2.2.8.1 getObjectRelationships

4252 **Description:** Gets all or a subset of relationships associated with an independent object.

#### 4253 2.2.8.1.1 Inputs

4254 **Required:**

- 4255 • **ID repositoryId:** The identifier for the Repository.
- 4256 • **ID objectId:** The identifier of the object.

4257

4258 **Optional:**

- 4259 • **Boolean includeSubRelationshipTypes:** If TRUE, then the Repository MUST return all  
4260 relationships whose Object-Types are descendant-types of the given object's cmis:objectTypeId  
4261 property value as well as relationships of the specified type.
  - 4262 ○ Default is FALSE
  - 4263 ○ If FALSE, then the Repository MUST only return relationships whose Object-Type is  
4264 equivalent to the given object's cmis:objectTypeId property value.
- 4265 • **Enum relationshipDirection:** An enumeration specifying whether the Repository MUST  
4266 return relationships where the specified Object is the source of the relationship, the target of the  
4267 relationship, or both. Valid values are:
  - 4268 ○ **source: (default)** The Repository MUST return only relationship objects where the specified  
4269 object is the source object.
  - 4270 ○ **target:** The Repository MUST return only relationship objects where the specified object is  
4271 the target object.
  - 4272 ○ **either:** The Repository MUST return relationship objects where the specified object is  
4273 either the source or the target object.
- 4274 • **ID typeId:** If specified, then the Repository MUST return only relationships whose Object-Type is  
4275 of the type specified
  - 4276 ○ If not specified, then the repository MUST return Relationship objects of all types.
- 4277 • **Integer maxItems:** See section 2.2.1.1 Paging.
- 4278 • **Integer skipCount:** See section 2.2.1.1 Paging.
- 4279 • **String filter:** See section 2.2.1.2.1 Properties.
- 4280 • **Boolean includeAllowableActions:** See section 2.2.1.2.6 Allowable Actions.

#### 4281 2.2.8.1.2 Outputs

- 4282 • **<Array> Objects:** A list of the relationship objects. Each object result MUST include the following  
4283 elements if they are requested:
  - 4284 ○ **<Array> Properties:** The list of properties for the object.
  - 4285 ○ **AllowableActions:** See section 2.2.1.2.6 Allowable Actions.
- 4286 • **Boolean hasMoreItems:** See section 2.2.1.1 Paging.

4287 **Optional:**

- 4288 • **Integer numItems:** See section 2.2.1.1 Paging.

4289

### 4290 2.2.8.1.3 Exceptions Thrown & Conditions

- 4291 • See section 2.2.1.4.1 General Exceptions
- 4292 • `filterNotValid`: The Repository MUST throw this exception if this property filter input
- 4293 parameter is not valid.

## 4294 2.2.9 Policy Services

4295 The Policy Services (*applyPolicy*, *removePolicy*, *getAppliedPolicies*) are used to apply or remove a policy  
4296 object to a *controllablePolicy* object.

### 4297 2.2.9.1 applyPolicy

4298 **Description:** Applies a specified policy to an object.

#### 4299 2.2.9.1.1 Inputs

4300 **Required:**

- 4301 • **ID repositoryId**: The identifier for the Repository.
- 4302 • **ID policyId**: The identifier for the Policy to be applied.
- 4303 • **ID objectId**: The identifier of the object.

#### 4304 2.2.9.1.2 Exceptions Thrown & Conditions

4305 See section 2.2.1.4.1 General Exceptions

4306 `constraint`: The Repository MUST throw this exception if the specified object's Object-Type  
4307 `definition`'s attribute for *controllablePolicy* is FALSE.

### 4308 2.2.9.2 removePolicy

4309 **Description:** Removes a specified policy from an object.

#### 4310 2.2.9.2.1 Inputs

4311 **Required:**

- 4312 • **ID repositoryId**: The identifier for the Repository.
- 4313 • **ID policyId**: The identifier for the Policy to be removed.
- 4314 • **ID objectId**: The identifier of the object.

#### 4315 2.2.9.2.2 Exceptions Thrown & Conditions

- 4316 • See section 2.2.1.4.1 General Exceptions
- 4317 • `constraint`: The Repository MUST throw this exception if the specified object's Object-Type
- 4318 `definition`'s attribute for *controllablePolicy* is FALSE.

### 4319 2.2.9.3 getAppliedPolicies

4320 **Description:** Gets the list of policies currently applied to the specified object.

#### 4321 2.2.9.3.1 Inputs

4322 **Required:**

- 4323 • **ID repositoryId**: The identifier for the Repository.
- 4324 • **ID objectId**: The identifier of the object.

4325 **Optional:**  
4326 **String filter:** See section 2.2.1.2.1 Properties.

### 4327 **2.2.9.3.2 Outputs**

4328 **<Array> Objects:** A list of Policy Objects.

### 4329 **2.2.9.3.3 Exceptions Thrown & Conditions**

- 4330 • See section 2.2.1.4.1 General Exceptions
- 4331 • `filterNotValid`: The Repository MUST throw this exception if this property filter input  
4332 parameter is not valid.

## 4333 **2.2.10 ACL Services**

### 4334 **2.2.10.1 getACL**

4335 **Description:** Get the ACL currently applied to the specified document or folder object.

#### 4336 **2.2.10.1.1 Inputs**

4337 **Required:**

- 4338 • **ID repositoryId:** The identifier for the repository.
- 4339 • **ID objectId:** The identifier for the object

4340 **Optional:**

- 4341 • **Boolean onlyBasicPermissions:** See section 2.8 Access Control. The repository SHOULD  
4342 make a best effort to fully express the native security applied to the object
  - 4343 ○ **TRUE:** (default value if not provided) indicates that the client requests that the returned  
4344 ACL be expressed using only the CMIS Basic permissions.
  - 4345 ○ **FALSE:** indicates that the server may respond using either solely CMIS Basic  
4346 permissions, or repository specific permissions or some combination of both.

#### 4347 **2.2.10.1.2 Outputs**

- 4348 • **<Array> AccessControlEntryType:** The list of access control entries of the ACL for the object.

4349 **Optional:**

- 4350 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –  
4351 i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.

### 4352 **2.2.10.1.3 Exceptions Thrown & Conditions**

- 4353 • See section 2.2.1.4.1 General Exceptions

#### 4354 **2.2.10.1.4 Notes**

4355 This service MUST be supported by a repository, if `getRepository` returns `capabilityACL=discover` or  
4356 `=manage`.

4357 How an ACL for the object is computed is up to the repository. A client MUST NOT assume that the ACEs  
4358 from the ACL as returned by this service can be applied via `applyACL`.

### 4359 **2.2.10.2 applyACL**

4360 **Description:** Adds or removes the given ACEs to or from the ACL of document or folder object.

### 4361 2.2.10.2.1 Inputs

#### 4362 Required:

- 4363 • **ID repositoryId:** The identifier for the repository.
- 4364 • **ID objectId:** The identifier for the object

#### 4365 Optional:

- 4366 • **<Array> AccessControlEntryType addACEs:** The ACEs to be added.
- 4367 • **<Array> AccessControlEntryType removeACEs:** The ACEs to be removed.
- 4368 • **Enum ACLPropagation:** Specifies how ACEs should be handled:
  - 4369 ○ **objectonly:** ACEs must be applied without changing the ACLs of other objects.
  - 4370 ○ **propagate:** ACEs must be applied by propagate the changes to all **"inheriting"**
  - 4371 **objects.**
  - 4372 ○ **repositorydetermined: Default value.** Indicates that the client leaves the behavior to
  - 4373 **the repository.**

### 4374 2.2.10.2.2 Outputs

- 4375 • **<Array> AccessControlEntryType:** The list of access control entries of the resulting ACL for the
- 4376 **object**

#### 4377 Optional:

- 4378 • **Boolean exact:** An indicator that the ACL returned fully describes the permission for this object –
- 4379 **i.e. there are no other security constraints applied to this object. Not provided defaults to FALSE.**
- 4380 • ~~**String changeToken:** See section 2.2.1.3 Change Tokens.~~

### 4381 2.2.10.2.3 Exceptions Thrown & Conditions

- 4382 • See section 2.2.1.4.1 General Exceptions
- 4383 • **constraint:** The Repository MUST throw this exception if ANY of the following conditions are
- 4384 **met:**
  - 4385 ○ The specified object's Object-Type definition's attribute for *controllableACL* is FALSE.
  - 4386 ○ The value for *ACLPropagation* does not match the values as returned via
  - 4387 *getACLCapabilities*.
  - 4388 ○ At least one of the specified values for *permission* in ANY of the ACEs does not match
  - 4389 ANY of the *permissionNames* as returned by *getACLCapability* and is not a CMIS Basic
  - 4390 *permission*

### 4391 2.2.10.2.4 Notes

4392 This service MUST be supported by a repository, if *getRepository* returns *capabilityACL=manage*.

4393 How ACEs are added or removed to or from the object is up to the repository – with respect to the

4394 *ACLPropagation* provided by the client. For **"shared"** ACEs (e.g. via inheritance), the repository MAY

4395 merge the ACEs provided with the ACEs of the ACL already applied to the object (i.e. the ACEs provided

4396 MAY not be completely added or removed from the effective ACL for the object).

4397

4398

---

## 4399 3 Restful AtomPub Binding

### 4400 3.1 Overview

4401 This binding is based upon the Atom (RFC4287) and Atom Publishing Protocol (RFC5023).  
4402 Implementations of CMIS MUST be compliant with RFC4287 and RFC5023.

4403  
4404 In this binding, the client interacts with the repository by acquiring the service document. The client will  
4405 request the service document by the URI provided by the vendor. The client will then choose a CMIS  
4406 collection, and then start accessing the repository by following the references in the returned documents.

4407  
4408 This binding consists of a service document specifying at least CMIS service collections, atom collections,  
4409 feeds and entry documents. CMIS extends the Atom and AtomPub documents utilizing the Atom and  
4410 AtomPub extension mechanism. CMIS also leverages link tags to specify additional resources related to  
4411 the requested resource.

4412  
4413 When requesting a resource, optional parameters may be specified to change default behavior via query  
4414 parameters.

### 4415 3.1.1 Namespaces

4416 This specification uses the following namespaces and prefixes when referring to xml or xml schema  
4417 elements in the text or examples:

- 4418 ~~• CMIS-Core: <http://docs.oasis-open.org/ns/cmisis/core/200908/>~~
- 4419 • [CMIS-Core: http://docs.oasis-open.org/ns/cmisis/core/200908/](http://docs.oasis-open.org/ns/cmisis/core/200908/)
  - 4420 ○ Prefix: cmis
- 4421 ~~• CMIS-RestAtom: <http://docs.oasis-open.org/ns/cmisis/restatom/200908/>~~
- 4422 • [CMIS-RestAtom: http://docs.oasis-open.org/ns/cmisis/restatom/200908/](http://docs.oasis-open.org/ns/cmisis/restatom/200908/)
  - 4423 ○ Prefix: cmisra
- 4424 ~~• Atom : <http://www.w3.org/2005/Atom>~~
- 4425 • [Atom : http://www.w3.org/2005/Atom](http://www.w3.org/2005/Atom)
  - 4426 ○ Prefix: atom
- 4427 ~~• AtomPub: <http://www.w3.org/2007/app>~~
- 4428 • [AtomPub: http://www.w3.org/2007/app](http://www.w3.org/2007/app)
  - 4429 ○ Prefix: app

### 4431 3.1.2 Authentication

4432 Authentication SHOULD be handled by the transport protocol. Please see AtomPub (RFC5023) section  
4433 14.

### 4435 3.1.3 Response Formats

4436 The client can specify, in HTTP the Accept header, which formats are acceptable to the client. With this  
4437 mechanism the client can chose which response format the CMIS implementation should respond with.

4438 The CMIS compliant implementation MUST support the appropriate Media Types specified in this  
4439 document.

### 4440 **3.1.4 Optional Arguments**

4441 The binding supports adding optional parameters to CMIS resources to modify the default behavior.  
4442 CMIS implementations MUST support arguments being specified as HTTP query string parameters.

4443 Names and valid values for HTTP query string parameters are as described in the appropriate CMIS  
4444 Service descriptions [see CMIS Domain Model]. Valid values of enumeration types are also represented  
4445 in the CMIS Core XML Schema

### 4446 **3.1.5 Errors and Exceptions**

4447 Exceptions MUST be mapped to the appropriate HTTP status code.

4448 Repositories SHOULD provide sufficient information in the body of the HTTP response for a user to  
4449 determine corrective action.

4450 See Section 3.2.4 HTTP Status Codes for more information.

### 4451 **3.1.6 Renditions**

4452 Each Rendition included in a CMIS AtomPub response is represented as an Atom link with relationship  
4453 alternate.

4454

4455 The following attributes SHOULD be included on the link element:

- 4456 • href: URI to the rendition content stream
- 4457 • type: The Media Type of the Rendition
- 4458 • cmisra:renditionKind: The Rendition Kind for the Rendition

4459

4460 The following attributes MAY be included

- 4461 • title: The Filename (or name property if object) of Rendition
- 4462 • length: The length of the rendition

### 4463 **3.1.7 Content Streams**

4464 The content stream for a document SHOULD be referenced by the content src attribute as well as the  
4465 edit-media link relation.

4466 A CMIS Repository MAY use different URIs for both content src attribute and the edit-media link relation  
4467 for the same content stream.

4468 The following attributes SHOULD be included on the link element:

- 4469 • href: URI to the content stream
- 4470 • type: The Media Type of the content stream

### 4471 **3.1.8 Paging of Feeds**

4472 For paging, please see the AtomPub RFC. CMIS leverages first, next, previous, and last link relations to  
4473 express paging.

4474 If the repository can include the number of items (numItems in CMIS Domain Model) in a feed, then the  
4475 repository SHOULD include the cmisra:numItems extension element in the feed.

### 4476 **3.1.9 Services not Exposed**

4477 The following services are not exposed in this binding:

- 4478 • getRenditions: This is exposed as part of getObject
- 4479 • getProperties: This is exposed as part of getObject
- 4480 • createDocumentFromSource: This is not exposed in this binding except as the client saving the
- 4481 resource and resubmitting it without the cmis:objectId.
- 4482 • Setting ACL on Create or Checkin operations
  - 4483 ○ This is currently not possible with the REST binding. The Create or Checkin operation
  - 4484 must be performed first. Then the dependent resource, ACL, must be retrieved and
  - 4485 updated.
- 4486 • setContentStream: This does not return the new object id and change token as specified by the
- 4487 domain model. This is not possible without introducing a new HTTP header.
- 4488 • deleteContentStream: This does not return the new object id and change token as specified by
- 4489 the domain model. This is not possible without introducing a new HTTP header.
- 4490 • checkOut: This does not return whether or not content was copied. This is not possible without
- 4491 introducing a new HTTP header.

### 4492 3.1.9.1 removePolicy

4493 This service is exposed from the domain model in the RESTful Atom Binding. However, it is not as  
 4494 straightforward. To remove a policy from an object, one must do:

- 4495 • Get the object.
- 4496 • Fetch the policies collection of the object.
- 4497 • Walk through the feed and find the policy object where cmis:objectId == policy id to remove.
- 4498 • Get the self lin of this policy object.
- 4499 • Perform a DELETE on this URL.

4500  
 4501 This is also the only case in the RESTful Atom Binding where an URI in a collection (policies) is specific  
 4502 to that collection.

## 4503 3.2 HTTP

### 4504 3.2.1 Entity Tag

4505 CMIS changeTokens are represented as Entity Tags and follow HTTP's use of Entity Tags. CMIS server  
 4506 implementations SHOULD support Entity Tags. ChangeTokens are also provided as properties and  
 4507 SHOULD be provided when the object is included inside an atom entry or feed.

4508  
 4509 On updates, perform the following checks (HTTP & CMIS levels):

- 4510  
 4511 1. If If-Match header is sent by client, ETag value is pulled from HTTP header If-Match per  
 4512 RFC2616. The supplied ETag is compared against the ETag on the server. If the match fails,  
 4513 then status code 412 is used.
- 4514  
 4515 2. If cmis:changeToken property is supplied by the client, compare the supplied and the  
 4516 cmis:changeToken on the server. If the comparison fails, then return status code 409 per CMIS.
- 4517  
 4518 3. If ETag and cmis:changeToken are both specified, the HTTP If-Match check should be performed  
 4519 first.

4520 **3.2.2 HTTP Range**

4521 Repositories MAY support HTTP Range requests on Content Streams.

4522 **3.2.3 HTTP OPTIONS Method**

4523 The repository MAY support the HTTP OPTIONS method on all the resources defined in this  
4524 specification. If the repository supports OPTIONS, then the repository MUST at least return the HTTP  
4525 methods specified for that resource in the Allow header.

4526 **3.2.4 HTTP Status Codes**

4527 Please see the HTTP specification for more information on the HTTP status codes. These are provided  
4528 as guidance from the HTTP specification. If any conflict arises, the HTTP specification is authoritative.

4529 **3.2.4.1 General CMIS Exceptions**

4530 The following listing defines the HTTP status codes that repositories MUST return for the various common  
4531 exceptions defined in CMIS Domain Model.

4532	CMIS Services Exception	HTTP Status Code
4533	invalidArgument	400
4534	objectNotFound	404
4535	permissionDenied	403
4536	notSupported	405
4537	runtime	500
4538	constraint	409
4539	filterNotValid	400
4540	streamNotSupported	403
4541	storage	500
4542	contentAlreadyExists	409
4543	versioning	409
4544	updateConflict	409
4545	nameConstraintViolation	409

4546

4547 **3.2.4.2 Notable HTTP Status Codes**

- 4548 • 415 Unsupported Media Type
  - 4549 | ○ When a document is POST<sup>ed</sup> to a collection that does not support the media type of the  
4550 | document, this status code MUST be returned
- 4551 • 422 Unprocessable Entity (Defined in RFC4918 Section 11.2)
  - 4552 | ○ When a request has been POST<sup>ed</sup> but cannot be processed, this status code MUST be  
4553 | returned

4554

4555 Please see RFC2616 Section 10 for more information.

4556

4557 **3.3 Media Types**

4558 CMIS introduces new media types for:

- 4559 • a CMIS Query document (application/cmismessaging+xml)
- 4560 • a CMIS AllowableActions document (application/cmismessaging+xml)
- 4561 • an Atom Document (Entry or Feed) with any CMIS Markup (application/cmismessaging+xml)
- 4562 • an Atom Feed Document with CMIS Hierarchy extensions (application/cmismessaging+xml)
- 4563 • a CMIS ACL Document (application/cmismessaging+xml)

4564

4565 In addition to those media types specified by CMIS, CMIS also leverages these media types:

- 4566 • AtomPub Service (application/atomsvc+xml)
- 4567 • Atom Entry (application/atom+xml;type=entry)
- 4568 • Atom Feed (application/atom+xml;type=feed)

### 4569 3.3.1 CMIS Atom

4570 Media Type: application/cmismessaging+xml

4571 Starting tag: atom:feed or atom:entry

4572 Type Parameters:

- 4573 • type – the semantics of the type parameter MUST be the same as the media type parameter for
- 4574 atom documents.

4575

4576 This allows clients to differentiate between repositories that require atom media type with CMIS  
 4577 extensions (application/cmismessaging+xml) for creation and repositories that allow generic atom media type  
 4578 without CMIS extensions (application/atom+xml).

4579

4580 This is only used for CMIS repositories to advertise what media types are accepted for adding to a  
 4581 collection (e.g., creating resources in a collection). As such CMIS does not require specifying whether an  
 4582 atom feed has CMIS markup. It is included to be consistent with the Atom media type.

4583

4584 All feeds and entries from a CMIS repository MUST utilize the atom media type for exposing Atom  
 4585 resources. Please see the individual resources for more information on the media type. This provides  
 4586 the interoperability with Atom clients.

4587

4588 Example:

4589

```

4590 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4591 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmismessaging/200908/"
4592 xmlns:cmism="http://docs.oasis-open.org/ns/cmismessaging/200908/"
4593 xmlns:atom="http://www.w3.org/2005/Atom"
4594 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4595 open.org/ns/cmismessaging/200908/">
4596   <atom:author>
4597     <atom:name>Al Brown</atom:name>
4598   </atom:author>
4599   <atom:id>urn:uuid:efe0542e-8933-4b3e-93f2-4d1caa3fc2d9</atom:id>
4600   <atom:title type="text">CMIS Example Document</atom:title>
4601   <atom:updated>2010-01-25T10:20:58.318-08:00</atom:updated>
4602   <atom:content type="text">some text</atom:content>
4603   <cmisra:object>
4604     <cmis:properties>
4605       <cmis:propertyId localName="rep-cmismessaging:objectTypeId"
4606 propertyDefinitionId="cmismessaging:objectTypeId">
4607         <cmis:value>invoice</cmis:value>

```

```
4608         </cmis:propertyId>
4609         <cmis:propertyString localName="rep-cmis:name"
4610 propertyDefinitionId="cmis:name">
4611             <cmis:value>CMIS Example Document</cmis:value>
4612         </cmis:propertyString>
4613     </cmis:properties>
4614 </cmisra:object>
4615 </atom:entry>
4616
```

### 4617 **3.3.2 CMIS Query**

4618 Media Type: application/cmismquery+xml

4619 Starting tag: cmis:query

4620

4621 This document contains the representation of a query to be executed in a CMIS repository.

4622

4623 Example:

```
4624 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4625 <cmis:query xmlns:cmis="http://docs.oasis-open.org/ns/cmism/core/200908/"
4626 xmlns:cmism="http://docs.oasis-open.org/ns/cmism/messaging/200908/"
4627 xmlns:atom="http://www.w3.org/2005/Atom"
4628 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4629 open.org/ns/cmism/restatom/200908/">
4630     <cmis:statement>SELECT * FROM cmis:document</cmis:statement>
4631     <cmis:searchAllVersions>true</cmis:searchAllVersions>
4632     <cmis:includeAllowableActions>false</cmis:includeAllowableActions>
4633     <cmis:includeRelationships>none</cmis:includeRelationships>
4634     <cmis:renditionFilter>*</cmis:renditionFilter>
4635     <cmis:maxItems>50</cmis:maxItems>
4636     <cmis:skipCount>0</cmis:skipCount>
4637 </cmis:query>
```

4638

4639 Please also see the example documents included with the schema.

4640

### 4641 **3.3.3 CMIS Allowable Actions**

4642 Media Type: application/cmismallowableactions+xml

4643 Starting tag: cmis:allowableActions

4644

4645 This document contains the representation of the allowable actions the user may perform on the  
4646 referenced object.

4647

4648 Example:

```
4649 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4650 <cmis:allowableActions xmlns:cmis="http://docs.oasis-
4651 open.org/ns/cmism/core/200908/" xmlns:cmism="http://docs.oasis-
4652 open.org/ns/cmism/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
4653 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4654 open.org/ns/cmism/restatom/200908/">
4655     <cmis:canDeleteObject>true</cmis:canDeleteObject>
4656     <cmis:canUpdateProperties>true</cmis:canUpdateProperties>
4657     <cmis:canGetProperties>true</cmis:canGetProperties>
4658     <cmis:canGetObjectRelationships>true</cmis:canGetObjectRelationships>
4659     <cmis:canGetObjectParents>true</cmis:canGetObjectParents>
```

```

4660 <cmis:canMoveObject>true</cmis:canMoveObject>
4661 <cmis:canDeleteContentStream>true</cmis:canDeleteContentStream>
4662 <cmis:canCheckOut>true</cmis:canCheckOut>
4663 <cmis:canCancelCheckOut>true</cmis:canCancelCheckOut>
4664 <cmis:canCheckIn>true</cmis:canCheckIn>
4665 <cmis:canSetContentStream>true</cmis:canSetContentStream>
4666 <cmis:canGetAllVersions>true</cmis:canGetAllVersions>
4667 <cmis:canAddObjectToFolder>true</cmis:canAddObjectToFolder>
4668 <cmis:canRemoveObjectFromFolder>true</cmis:canRemoveObjectFromFolder>
4669 <cmis:canGetContentStream>true</cmis:canGetContentStream>
4670 <cmis:canApplyPolicy>true</cmis:canApplyPolicy>
4671 <cmis:canGetAppliedPolicies>true</cmis:canGetAppliedPolicies>
4672 <cmis:canRemovePolicy>true</cmis:canRemovePolicy>
4673 <cmis:canCreateDocument>true</cmis:canCreateDocument>
4674 </cmis:allowableActions>

```

4675

4676 Please also see the example documents included with the schema.

4677

### 4678 3.3.4 CMIS Tree

4679 Media Type: application/cmistree+xml

4680 Starting tag: atom:feed

4681

4682 This document is an atom feed (application/atom+xml;type=feed) with CMIS markup to nest a hierarchy.

4683

4684 Please see Section 3.5.2.1 for more information.

4685

4686 Example:

```

4687 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4688 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
4689 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
4690 xmlns:atom="http://www.w3.org/2005/Atom"
4691 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4692 open.org/ns/cmisis/restatom/200908/"
4693 <atom:title type="text">Feed for folder1</atom:title>
4694 <atom:author>
4695 <atom:name>Al Brown</atom:name>
4696 <atom:uri>http://www.ibm.com/</atom:uri>
4697 <atom:email>albertcbrown@us.ibm.com</atom:email>
4698 </atom:author>
4699 <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4700 <atom:id>urn:uuid:4a80905c-f774-4a9e-a57d-bf0dae5a796e</atom:id>
4701 <atom:link type="application/atom+xml;type=feed" rel="self"
4702 href="http://cmisexample.oasis-open.org/repl/cf3c076e-36e9-4ace-8fed-
4703 41e0d92dfc71/3"/>
4704 <atom:link type="application/atomsvc+xml" rel="service"
4705 href="http://cmisexample.oasis-open.org/repl//service"/>
4706 <atom:link type="application/atom+xml;type=entry" rel="via"
4707 href="http://cmisexample.oasis-open.org/repl/cf3c076e-36e9-4ace-8fed-
4708 41e0d92dfc71"/>
4709 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
4710 open.org/ns/cmisis/link/200908/foldertree" href="http://cmisexample.oasis-
4711 open.org/repl/cf3c076e-36e9-4ace-8fed-41e0d92dfc71/foldertree"/>
4712 <atom:link type="application/atom+xml;type=feed" rel="down"
4713 href="http://cmisexample.oasis-open.org/repl/cf3c076e-36e9-4ace-8fed-
4714 41e0d92dfc71/children"/>

```

```

4715     <atom:link type="application/atom+xml;type=entry" rel="up"
4716 href="http://cmisexample.oasis-open.org/rep1/bb11830c-7d1e-4b0f-9ff2-
4717 af4857c49200"/>
4718     <atom:entry>
4719         <atom:author>
4720             <atom:name>Al Brown</atom:name>
4721             <atom:uri>http://www.ibm.com/</atom:uri>
4722             <atom:email>albertcbrown@us.ibm.com</atom:email>
4723         </atom:author>
4724         <atom:content src="http://cmisexample.oasis-open.org/rep1/63a9c18c-
4725 5e31-4590-8462-86d181e345a4"/>
4726         <atom:id>urn:uuid:63a9c18c-5e31-4590-8462-86d181e345a4</atom:id>
4727         <atom:title type="text">CMIS Example Folder as Customer
4728 type</atom:title>
4729         <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4730         <atom:link rel="self" href="http://cmisexample.oasis-
4731 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4"/>
4732         <atom:link rel="edit" href="http://cmisexample.oasis-
4733 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4"/>
4734         <atom:link type="application/cmisacl+xml;type=allowableActions"
4735 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions"
4736 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4737 86d181e345a4/allowableactions"/>
4738         <atom:link type="application/atom+xml;type=entry" rel="describedby"
4739 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4740 86d181e345a4/type"/>
4741         <atom:link type="application/atomsvc+xml" rel="service"
4742 href="http://cmisexample.oasis-open.org/rep1//service"/>
4743         <atom:published>2010-01-25T10:20:58.536-08:00</atom:published>
4744         <atom:summary type="html">HTML summary of Entry 63a9c18c-5e31-4590-
4745 8462-86d181e345a4</atom:summary>
4746         <atom:link type="application/atom+xml;type=entry" rel="up"
4747 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4748 86d181e345a4/up"/>
4749         <atom:link type="application/atom+xml;type=feed" rel="down"
4750 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4751 86d181e345a4/children"/>
4752         <atom:link type="application/cmistree+xml" rel="down"
4753 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4754 86d181e345a4/tree"/>
4755         <atom:link type="application/atom+xml;type=feed"
4756 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/foldertree"
4757 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4758 86d181e345a4/foldertree"/>
4759         <atom:link type="application/atom+xml;type=feed"
4760 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/relationships"
4761 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4762 86d181e345a4/relationships"/>
4763         <atom:link type="application/atom+xml;type=feed"
4764 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/policies"
4765 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4766 86d181e345a4/policies"/>
4767         <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
4768 open.org/ns/cmisacl/link/200908/acl" href="http://cmisexample.oasis-
4769 open.org/rep1/63a9c18c-5e31-4590-8462-86d181e345a4/acl"/>
4770         <cmisra:object>
4771             <cmis:properties>
4772                 <cmis:propertyId localName="rep-cmis:objectId"
4773 propertyDefinitionId="cmis:objectId">
4774                     <cmis:value>63a9c18c-5e31-4590-8462-
4775 86d181e345a4</cmis:value>
4776                 </cmis:propertyId>
4777             </cmis:properties>
4778         </cmisra:object>

```

```

4779     <cmisra:pathSegment>customer</cmisra:pathSegment>
4780     <cmisra:children>
4781         <atom:feed>
4782             <atom:title type="text">CMIS Example Folder as Customer
4783 type</atom:title>
4784             <atom:author>
4785                 <atom:name>Al Brown</atom:name>
4786                 <atom:uri>http://www.ibm.com/</atom:uri>
4787                 <atom:email>albertcbrown@us.ibm.com</atom:email>
4788             </atom:author>
4789             <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4790             <atom:id>urn:uuid:51b5c0cd-e473-4492-82b3-
4791 666fbf913cf0</atom:id>
4792             <atom:link type="application/atom+xml;type=feed" rel="self"
4793 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4794 86d181e345a4/3"/>
4795             <atom:link type="application/atomsvc+xml" rel="service"
4796 href="http://cmisexample.oasis-open.org/rep1//service"/>
4797             <atom:link type="application/atom+xml;type=entry" rel="via"
4798 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4799 86d181e345a4"/>
4800             <atom:link type="application/atom+xml;type=feed"
4801 rel="http://docs.oasis-open.org/ns/cmisa/link/200908/foldertree"
4802 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4803 86d181e345a4/foldertree"/>
4804             <atom:link type="application/atom+xml;type=feed" rel="down"
4805 href="http://cmisexample.oasis-open.org/rep1/63a9c18c-5e31-4590-8462-
4806 86d181e345a4/children"/>
4807             <atom:link type="application/atom+xml;type=entry" rel="up"
4808 href="http://cmisexample.oasis-open.org/rep1/cf3c076e-36e9-4ace-8fed-
4809 41e0d92dfc71"/>
4810         <atom:entry>
4811             <atom:author>
4812                 <atom:name>Al Brown</atom:name>
4813                 <atom:uri>http://www.ibm.com/</atom:uri>
4814                 <atom:email>albertcbrown@us.ibm.com</atom:email>
4815             </atom:author>
4816             <atom:content src="http://cmisexample.oasis-
4817 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4818             <atom:id>urn:uuid:20cb7e68-0a7e-46ea-87e0-
4819 09fb8d85286e</atom:id>
4820             <atom:title type="text">CMIS Example Doc as Invoice
4821 type</atom:title>
4822             <atom:updated>2010-01-25T10:20:58.536-08:00</atom:updated>
4823             <atom:link rel="self" href="http://cmisexample.oasis-
4824 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4825             <atom:link rel="edit" href="http://cmisexample.oasis-
4826 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e"/>
4827             <atom:link
4828 type="application/cmisa+xml;type=allowableActions" rel="http://docs.oasis-
4829 open.org/ns/cmisa/link/200908/allowableactions" href="http://cmisexample.oasis-
4830 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/allowableactions"/>
4831             <atom:link type="application/atom+xml;type=entry"
4832 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-
4833 46ea-87e0-09fb8d85286e/type"/>
4834             <atom:link type="application/atomsvc+xml" rel="service"
4835 href="http://cmisexample.oasis-open.org/rep1//service"/>
4836             <atom:published>2010-01-25T10:20:58.536-
4837 08:00</atom:published>
4838             <atom:summary type="html">HTML summary of Entry 20cb7e68-
4839 0a7e-46ea-87e0-09fb8d85286e</atom:summary>
4840             <atom:link rel="edit-media"
4841 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4842 09fb8d85286e/edit-media"/>

```

```

4843         <atom:link rel="alternate" href="http://cmisexample.oasis-
4844 open.org/rep1/20cb7e68-0a7e-46ea-87e0-09fb8d85286e/alternate"/>
4845         <atom:link type="application/atom+xml;type=feed" rel="up"
4846 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4847 09fb8d85286e/parents"/>
4848         <atom:link type="application/atom+xml;type=feed"
4849 rel="version-history" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-
4850 0a7e-46ea-87e0-09fb8d85286e/allversions"/>
4851         <atom:link type="application/atom+xml;type=entry"
4852 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/20cb7e68-
4853 0a7e-46ea-87e0-09fb8d85286e/latest"/>
4854         <atom:link type="application/atom+xml;type=feed"
4855 rel="http://docs.oasis-open.org/ns/cmisacl/200908/relationships"
4856 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4857 09fb8d85286e/relationships"/>
4858         <atom:link type="application/atom+xml;type=feed"
4859 rel="http://docs.oasis-open.org/ns/cmisacl/200908/policies"
4860 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4861 09fb8d85286e/policies"/>
4862         <atom:link type="application/cmisacl+xml"
4863 rel="http://docs.oasis-open.org/ns/cmisacl/200908/acl"
4864 href="http://cmisexample.oasis-open.org/rep1/20cb7e68-0a7e-46ea-87e0-
4865 09fb8d85286e/acl"/>
4866         <cmisra:object>
4867           <cmis:properties>
4868             <cmis:propertyId localName="rep-cmis:objectId"
4869 propertyDefinitionId="cmis:objectId">
4870 <cmis:value>20cb7e68-0a7e-46ea-87e0-09fb8d85286e</cmis:value>
4871             </cmis:propertyId>
4872           </cmis:properties>
4873         </cmisra:object>
4874         <cmisra:pathSegment>invoicel.pdf</cmisra:pathSegment>
4875       </atom:entry>
4876     </atom:feed>
4877   </cmisra:children>
4878 </atom:entry>
4879 <atom:entry>
4880   <atom:author>
4881     <atom:name>Al Brown</atom:name>
4882     <atom:uri>http://www.ibm.com/</atom:uri>
4883     <atom:email>albertcbrown@us.ibm.com</atom:email>
4884   </atom:author>
4885   <atom:content src="http://cmisexample.oasis-open.org/rep1/1de1d476-
4886 11fb-47bf-b136-8a8d0b4b030a"/>
4887   <atom:id>urn:uuid:1de1d476-11fb-47bf-b136-8a8d0b4b030a</atom:id>
4888   <atom:title type="text">CMIS Example Doc as Invoice type</atom:title>
4889   <atom:updated>2010-01-25T10:20:58.568-08:00</atom:updated>
4890   <atom:link rel="self" href="http://cmisexample.oasis-
4891 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a"/>
4892   <atom:link rel="edit" href="http://cmisexample.oasis-
4893 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a"/>
4894   <atom:link type="application/cmisacl+xml;type=allowableActions"
4895 rel="http://docs.oasis-open.org/ns/cmisacl/200908/allowableactions"
4896 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4897 8a8d0b4b030a/allowableactions"/>
4898   <atom:link type="application/atom+xml;type=entry" rel="describedby"
4899 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4900 8a8d0b4b030a/type"/>
4901   <atom:link type="application/atomsvc+xml" rel="service"
4902 href="http://cmisexample.oasis-open.org/rep1//service"/>
4903   <atom:published>2010-01-25T10:20:58.568-08:00</atom:published>
4904   <atom:summary type="html">HTML summary of Entry 1de1d476-11fb-47bf-
4905 b136-8a8d0b4b030a</atom:summary>

```

```

4906     <atom:link rel="edit-media" href="http://cmisexample.oasis-
4907 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/edit-media"/>
4908     <atom:link rel="alternate" href="http://cmisexample.oasis-
4909 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/alternate"/>
4910     <atom:link type="application/atom+xml;type=feed" rel="up"
4911 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4912 8a8d0b4b030a/parents"/>
4913     <atom:link type="application/atom+xml;type=feed" rel="version-history"
4914 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4915 8a8d0b4b030a/allversions"/>
4916     <atom:link type="application/atom+xml;type=entry" rel="current-
4917 version" href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4918 8a8d0b4b030a/latest"/>
4919     <atom:link type="application/atom+xml;type=feed"
4920 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
4921 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4922 8a8d0b4b030a/relationships"/>
4923     <atom:link type="application/atom+xml;type=feed"
4924 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
4925 href="http://cmisexample.oasis-open.org/rep1/1de1d476-11fb-47bf-b136-
4926 8a8d0b4b030a/policies"/>
4927     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
4928 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
4929 open.org/rep1/1de1d476-11fb-47bf-b136-8a8d0b4b030a/acl"/>
4930     <cmisra:object>
4931       <cmis:properties>
4932         <cmis:propertyId localName="rep-cmis:objectId"
4933 propertyDefinitionId="cmis:objectId">
4934           <cmis:value>1de1d476-11fb-47bf-b136-
4935 8a8d0b4b030a</cmis:value>
4936         </cmis:propertyId>
4937       </cmis:properties>
4938     </cmisra:object>
4939     <cmisra:pathSegment>invoice3.pdf</cmisra:pathSegment>
4940   </atom:entry>
4941 </atom:feed>

```

4942

4943 Note: This media type is used on links with relation down (see section 3.4.3.2 Hierarchy Navigation  
4944 Internet Draft Link Relations). When the individual resources are returned by the CMIS repository they  
4945 will use the atom media type (application/atom+xml)

4946

4947 Please also see the example documents included with the schema.

4948

### 4949 3.3.5 CMIS ACL

4950 Media Type: application/cmisisacl+xml

4951 Starting tag: cmis:acl

4952

4953 This document specifies an Access Control List based on the schema in CMIS Domain Model.

4954

4955 Example:

```

4956 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
4957 <cmis:acl xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
4958 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
4959 xmlns:atom="http://www.w3.org/2005/Atom"
4960 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
4961 open.org/ns/cmisis/restatom/200908/">

```

```
4962 <cmis:permission>
4963   <cmis:principal>
4964     <cmis:principalId>Al Brown</cmis:principalId>
4965   </cmis:principal>
4966   <cmis:permission>cmis:read</cmis:permission>
4967   <cmis:permission>cmis:write</cmis:permission>
4968   <cmis:permission>cmis:all</cmis:permission>
4969   <cmis:permission>publish</cmis:permission>
4970   <cmis:direct>true</cmis:direct>
4971 </cmis:permission>
4972 </cmis:acl>
```

4973

4974 Please also see the example documents included with the schema.

4975

## 4976 **3.4 Atom Extensions for CMIS**

### 4977 **3.4.1 Atom Element Extensions**

#### 4978 **3.4.1.1 AtomPub Workspace**

##### 4979 **3.4.1.1.1 cmisra:collectionType**

4980 This element is included inside the app:collection element. This specifies the cmis collection type.

##### 4981 **3.4.1.1.2 cmisra:repositoryInfo**

4982 This element is included inside the app:workspace element. This specifies information about the CMIS  
4983 repository.

##### 4984 **3.4.1.1.3 cmis:uritemplate**

4985 This element is included inside the app:workspace element. This specifies information about URI  
4986 templates

#### 4987 **3.4.1.2 Atom Feed**

##### 4988 **3.4.1.2.1 cmisra:numItems**

4989 This element is included inside the atom:feed element. This specifies the number of items in the feed.

#### 4990 **3.4.1.3 Atom Entry**

##### 4991 **3.4.1.3.1 cmisra:children**

4992 This element is included inside the atom:entry element. This includes the children of the atom entry. This  
4993 element MUST include an atom:feed element.

4994

##### 4995 **3.4.1.3.2 cmisra:object**

4996 This element is included inside the atom:entry element for CMIS Document, Folder, Relationship and  
4997 Policy objects. This specifies the CMIS object information for the atom entry.

4998

4999 **3.4.1.3.3 cmisra:pathSegment**

5000 This element is included inside the atom:entry element for CMIS Type Definitions that are filable. This  
5001 specifies the pathSegment for this object in the folder representing the feed.

5002

5003 **3.4.1.3.4 cmisra:relativePathSegment**

5004 This element is included inside the atom:entry element. This specifies the relative pathSegment for the  
5005 object in that particular folder. This MUST be used only inside an object parents feed.

5006 **3.4.1.3.5 cmisra:type**

5007 This element is included inside the atom:entry element for CMIS Type Definitions. This specifies the type  
5008 definition the atom entry represents.

5009 **3.4.1.3.6 cmisra:content**

5010 This element specifies the content of the atom:entry element. The content is base64 encoded in the  
5011 base64 element. The elements of a cmisra:content element are:

- 5012 • mediaType: This contains the media type of the content as described by RFC4288.
- 5013 • base64: This contains the base64 content of the file

5014

5015 This element MUST take precedence over atom:content on submission of an atom entry to a repository.

5016

5017 A repository MUST use the atom:content element to return back to the client the content of the document.

5018

5019 Note: This is required when the client has an XML document stored that is might not be well formed and  
5020 thus would not be able to be included inside atom:content element.

5021

5022 **3.4.2 Attributes**

5023 These attributes are in the CMIS RestAtom namespace (cmisra).

5024 **3.4.2.1 cmisra:id**

5025 This attribute is used on the atom:link element to specify the cmis id of the resource. This attribute  
5026 SHOULD be on all link relations that point to a CMIS object.

5027

5028 This attribute MAY also be on cmisra:type. The value of the attribute on cmisra:type MUST be the same as  
5029 the type definition id.

5030

5031 Example:

```
5032 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
5033 <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"  
5034 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"  
5035 xmlns:atom="http://www.w3.org/2005/Atom"  
5036 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
5037 open.org/ns/cmisis/restatom/200908/" type="application/atom+xml;type=feed"  
5038 rel="down" href="http://cmisexample.oasis-open.org/rep1//children/e170da7d-  
5039 d322-472d-b1eb-67bdb1ec18ca/1" cmisra:id="e170da7d-d322-472d-b1eb-  
5040 67bdb1ec18ca"/>
```

5041

5042 Please also see the example documents included with the schema.

5043

### 5044 **3.4.2.2 cmisra:renditionKind**

5045 This attribute is used on the atom:link element with relation alternate to specify the renditionKind of the  
5046 resource. This attribute SHOULD be on all link elements with relation alternate that are a CMIS rendition.

5047

5048 Example:

```
5049 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
5050 <atom:link xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"  
5051 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"  
5052 xmlns:atom="http://www.w3.org/2005/Atom"  
5053 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
5054 open.org/ns/cmisis/restatom/200908/" type="text/html" rel="alternate"  
5055 href="http://cmisexample.oasis-open.org/rep1//rendition/e170da7d-d322-472d-  
5056 b1eb-67bdb1ec18ca/1" cmisra:renditionKind="cmis:thumbnail"/>
```

5057

5058 Please also see the example documents included with the schema.

### 5059 **3.4.3 CMIS Link Relations**

5060 The listing below outlines the different link relation types in CMIS. This is in addition to the link relations  
5061 specified by Atom and Atom Publishing Protocol. The registry for link relations is located at  
5062 <http://www.iana.org/assignments/link-relations/link-relations.xhtml>.

5063

5064 The link element with a specified relation MUST be included if client can perform the operation. The  
5065 repository SHOULD omit the link relation if the operation is not available. The operation may not be  
5066 available due to a variety of reasons such as access control, administrative policies, or other  
5067 mechanisms.

5068

5069 Links may have the following attribute in addition to the ones specified by Atom and Atom Publishing  
5070 Protocol:

- 5071 • (CMIS) id: Specifies the CMIS ID of the resource referenced by the link. Repositories SHOULD  
5072 include this attribute for elements such as atom:link that point to CMIS resources that have an id.

5073

5074 These are the link relation types specified by CMIS:

#### 5075 **3.4.3.1 Existing Link Relations**

5076 Existing link relations should be used where appropriate by the implementation. In addition, the following  
5077 link relations are leveraged for the CMIS specification:

- 5078 • self

- 5079 ○ This link relation provides the URI to retrieve this resource again.

- 5080 ○ Service: The appropriate service that generated the atom entry or feed.

- 5081 ○ Resources: All except AllowableActions, ACL and Content Streams

- 5082 • service

- 5083 ○ The service link relation when provided on a CMIS resource MUST point to an AtomPub  
5084 service document with only one workspace element. This workspace element MUST  
5085 represent the repository containing that resource.

- 5086 ○ Media Type: application/atomsvc+xml

- 5087
  - Resources: All except AllowableActions, ACL and Content Streams
- 5088
  - describedby
    - 5089 ○ When used on a CMIS resource, this link relation MUST point to an atom entry that
    - 5090 describes the type of that resource.
    - 5091 ○ Service: getTypeDefinition on specified object
    - 5092 ○ Media Type: application/atom+xml;type=entry
    - 5093 ○ Resources: CMIS Document, CMIS Folder, CMIS Relationship, CMIS Policy objects and
    - 5094 CMIS Types
  - 5095 ● via
    - 5096 ○ When used on an Atom Feed document, this link relation MUST point to the atom entry
    - 5097 representing the CMIS resource from whom this feed is derived.
    - 5098 ○ Media Type: application/atom+xml;type=entry
    - 5099 ○ Resources: All CMIS Feeds and Collections
  - 5100 ● edit-media
    - 5101 ○ When used on a CMIS document resource, this link relation MUST point to the URI for
    - 5102 content stream of the CMIS document. This URI MUST be used to set or delete the
    - 5103 content stream. This URI MAY be used to retrieve the content stream for the document.
    - 5104 ○ Service: setContentStream (PUT) , deleteContentStream (DELETE)
    - 5105 ○ Media Type: Specific to resource
    - 5106 ○ Resources: CMIS Document
  - 5107 ● edit
    - 5108 ○ When used on a CMIS resource, this link relation MUST provide an URI that can be used
    - 5109 with the HTTP PUT method to modify the atom:entry for the CMIS resource
    - 5110 ○ Service: getObject (GET), updateProperties (PUT)
    - 5111 ○ Media Type: application/atom+xml;type=entry
    - 5112 ○ Resources: CMIS Documents, CMIS Folders, CMIS Relationships and CMIS Policies
  - 5113 ● alternate
    - 5114 ○ This is used to express Renditions on a CMIS resource. See section 3.1.6 Renditions.
    - 5115 ○ Service: getContentStream for specified rendition
    - 5116 ○ Resources: CMIS Document, CMIS Folder and CMIS Policies
  - 5117 ● first
    - 5118 ○ This is used for Paging. Please see the AtomPub specification.
    - 5119 ○ Media Type: application/atom+xml;type=feed
    - 5120 ○ Resources: All Feeds
  - 5121 ● previous
    - 5122 ○ This is used for Paging. Please see the AtomPub specification.
    - 5123 ○ Media Type: application/atom+xml;type=feed
    - 5124 ○ Resources: All Feeds
  - 5125 ● next
    - 5126 ○ This is used for Paging. Please see the AtomPub specification.
    - 5127 ○ Media Type: application/atom+xml;type=feed
    - 5128 ○ Resources: All Feeds
  - 5129 ● last
    - 5130 ○ This is used for Paging. Please see the AtomPub specification.

- 5131 ○ Media Type: application/atom+xml;type=feed
- 5132 ○ Resources: All Feeds

5133  
5134

5135 Please see <http://www.iana.org/assignments/link-relations/link-relations.xhtml> for more information on  
5136 these link relations.

### 5137 **3.4.3.2 Hierarchy Navigation Internet Draft Link Relations**

5138 CMIS leverages the following link relations:

- 5139 • up
  - 5140 ○ Service: getFolderParent, getObjectParents, getTypeDefinition, getObject
  - 5141 ○ Media Type: application/atom+xml;type=feed, application/atom+xml;type=entry
  - 5142 ○ Resources: CMIS Document, CMIS Folder, CMIS Type Definitions, CMIS Folder
  - 5143 Children, CMIS Folder Descendants, CMIS FolderTree, CMIS Type Children, CMIS Type
  - 5144 Descendants
    - 5145 ▪ This link relation is not included on CMIS Base Type Definitions or the CMIS
    - 5146 Root Folder
- 5147 • down
  - 5148 ○ Service: getChildren, getDescendants, getTypeChildren, getTypeDescendants
  - 5149 ○ Media Type:
    - 5150 ▪ For children: application/atom+xml;type=feed
    - 5151 ▪ For descendants: application/cmistree+xml
      - 5152 • The descendants feed resource when retrieved from the CMIS repository
      - 5153 will use the Atom Feed Media Type (application/atom+xml;type=feed)
  - 5154 ○ Resources: CMIS Folder, Type

### 5155 **3.4.3.3 Versioning Internet Draft Link Relations**

5156 CMIS leverages the following link relations from the Internet Draft:

- 5157 • version-history
  - 5158 ○ Service: getAllVersions
  - 5159 ○ Media Type: application/atom+xml;type=feed
  - 5160 ○ Resources: CMIS Document
- 5161 • current-version
  - 5162 ○ Service: getObjectForLatestVersion
  - 5163 ○ Media Type: application/atom+xml;type=entry
  - 5164 ○ Resources: CMIS Document
- 5165 • working-copy
  - 5166 ○ Service: getObject for private-working-copy specified by
  - 5167 cmis:versionSeriesCheckedOutId property
  - 5168 ○ Media Type: application/atom+xml;type=entry
  - 5169 ○ Resources: CMIS Document

### 5170 **3.4.3.4 CMIS Specific Link Relations**

5171 CMIS defines the following link relations:

- 5172 | ~~o <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>~~
- 5173 | o <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>
- 5174 |     ▪ This link relation MUST point to a resource containing a CMIS AllowableActions
- 5175 |     document for the CMIS resource containing this link relation.
- 5176 |     ▪ Service: getAllowableActions
- 5177 |     ▪ Media Type: application/cmisisallowableactions+xml
- 5178 |     ▪ Resources: CMIS Documents, CMIS Folders, CMIS Policies, and CMIS
- 5179 |     Relationships
- 5180 | ~~o <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>~~
- 5181 | o <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>
- 5182 |     ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS
- 5183 |     relationship resources for the CMIS resource containing this link relation.
- 5184 |     ▪ Service: getObjectRelationships
- 5185 |     ▪ Media Type: application/atom+xml;type=feed
- 5186 |     ▪ Resources: CMIS Documents, CMIS Folders, and CMIS Policies
- 5187 | ~~o <http://docs.oasis-open.org/ns/cmisis/link/200908/source>~~
- 5188 | o <http://docs.oasis-open.org/ns/cmisis/link/200908/source>
- 5189 |     ▪ When used on a CMIS Relationship resource, this link relation MUST point to an
- 5190 |     atom entry document for the CMIS Resource specified by the cmisis:sourcelid
- 5191 |     property on the relationship.
- 5192 |     ▪ Source Link on Relationship
- 5193 |     ▪ Media Type: application/atom+xml;type=entry
- 5194 |     ▪ Resources: CMIS Relationships
- 5195 | ~~o <http://docs.oasis-open.org/ns/cmisis/link/200908/target>~~
- 5196 | o <http://docs.oasis-open.org/ns/cmisis/link/200908/target>
- 5197 |     ▪ When used on a CMIS Relationship resource, this link relation MUST point to an
- 5198 |     atom entry document for the CMIS Resource specified by the cmisis:targetlid
- 5199 |     property on the relationship.
- 5200 |     ▪ Target Link on Relationship
- 5201 |     ▪ Media Type: application/atom+xml;type=entry
- 5202 |     ▪ Resources: CMIS Relationships
- 5203 | ~~o <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>~~
- 5204 | o <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>
- 5205 |     ▪ This link relation MUST point to a resource containing an Atom Feed of CMIS
- 5206 |     Policy resources for the CMIS resource containing this link relation.
- 5207 |     ▪ Service: getAppliedPolicies
- 5208 |     ▪ Media Type: application/atom+xml;type=feed
- 5209 |     ▪ Resources: CMIS Documents and CMIS Folders
- 5210 | ~~o <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>~~
- 5211 | o <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>
- 5212 |     ▪ This link relation MUST point to a resource containing a CMIS ACL document for
- 5213 |     the CMIS resource containing this link relation.
- 5214 |     ▪ Service: getACL
- 5215 |     ▪ Media Type: application/cmisisacl+xml

- 5216                   ▪ Resources: CMIS Documents, CMIS Folders, CMIS Relationships, and CMIS
- 5217                   Policies that are securable
- 5218                   ○ ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/changes>~~
- 5219                   ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/changes>
- 5220                   ▪ This link relation MUST point to an Atom Feed containing the set of changes
- 5221                   ▪ Service: getContentChanges
- 5222                   ▪ Media Type: application/atom+xml;type=feed
- 5223                   ▪ Resources: AtomPub Workspace Element in Service Document
- 5224                   ○ ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>~~
- 5225                   ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>
- 5226                   ▪ Used in AtomPub Service Document to identify the folder tree for a specified
- 5227                   folder
- 5228                   ▪ Service: getFolderTree
- 5229                   ▪ Media Type: application/atom+xml;type=feed
- 5230                   ▪ Resources: CMIS Folder, also used in AtomPub Service Document for root folder
- 5231                   ○ ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>~~
- 5232                   ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>
- 5233                   ▪ Used in AtomPub Service Document to identify the base types descendants
- 5234                   ▪ Service: getTypeDescendants
- 5235                   ▪ Media Type: application/atom+xml;type=feed
- 5236                   ▪ Resources: AtomPub Workspace Element in Service Document
- 5237                   ○ ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>~~
- 5238                   ○ <http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>
- 5239                   ▪ Used in AtomPub Service Document to identify the root folder descendants
- 5240                   ▪ Service: getDescendants for root folder
- 5241                   ▪ Media Type: application/atom+xml;type=feed
- 5242                   ▪ Resources: AtomPub Workspace Element in Service Document
- 5243

## 5244 **3.5 Atom Resources**

5245 For all Atom Resources used in this specification, the following MUST be followed:

### 5246 **3.5.1 Feeds**

5247 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis objects:

- 5248                   • atom:updated SHOULD be the latest time the folder or its contents was updated. If unknown by
- 5249                   the underlying repository, it MUST be the current time.
- 5250                   • atom:author/atom:name MUST be the CMIS property cmis:createdBy
- 5251                   • atom:title MUST be the CMIS property cmis:name
- 5252                   • The atom:link with relation self MUST be generated to return the URI of the feed. If paging or any
- 5253                   other mechanism is used to filter, sort, or change the representation of the feed, the URI MUST
- 5254                   point back a resource with the same representation.
- 5255                   • A feed SHOULD contain the element app:collection, describing the appropriate media types
- 5256                   supported for creation of new entries in the feed

- 5257 | • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom:s  
5258 specification and be a valid URI.
- 5259 | • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than  
5260 contained in the feed, then a link with the relation next MUST be included in the feed.
- 5261
- 5262 Any feed MUST be a valid Atom Feed document and conform to the guidelines below for cmis types:
- 5263 | • atom:updated SHOULD be the latest time type definition was updated. If unknown by the  
5264 underlying repository, it MUST be the current time.
- 5265 | • atom:author/atom:name is repository specific
- 5266 | • atom:title MUST be the displayName attribute of the CMIS Type Definition.
- 5267 | • The atom:link with relation self MUST be generated to return the URI of the feed
- 5268 | • atom:id SHOULD be derived from the id attribute of the CMIS Type Definition. This id MUST be  
5269 compliant with atom:s specification and be a valid URI.
- 5270 | • Feeds MAY be paged via the link relations specified in AtomPub. If more items are available than  
5271 contained in the feed, then a link with the relation next MUST be included in the feed.
- 5272

5273 If on the root type, all fields are repository specific.

5274

5275 Ordering of entries in a feed is repository-specific if orderBy argument is not specified. If orderBy  
5276 argument is specified, the order of the entries in the feed SHOULD conform to the ordering specified by  
5277 the orderBy argument.

5278

5279 Note: Please see [feedvalidator.org](http://feedvalidator.org) to validate Atom compliance.

## 5280 3.5.2 Entries

5281 At any point where an Atom document of type Entry is sent or returned, it must be a valid Atom Entry  
5282 document and conform to the guidelines below for a cmis object:

- 5283 | • atom:title MUST be the cmis:name property
- 5284 | • app:edited MUST be cmis:lastModified~~ed~~ationDate
- 5285 | • atom:updated MUST be cmis:lastModified~~ed~~ationDate
- 5286 | • atom:published MUST be cmis:created~~ed~~ionDate
- 5287 | • atom:author/atom:name MUST be cmis:createdBy
- 5288 | • All CMIS properties MUST be exposed in CMIS cmis:properties elements even if they are  
5289 duplicated in an atom element
- 5290 | • atom:id SHOULD be derived from cmis:objectId. This id MUST be compliant with atom:s  
5291 specification and be a valid ~~U~~RI.
- 5292 | • The repository SHOULD populate the atom:summary tag with text that best represents a  
5293 summary of the object. For example, an HTML table containing the properties and their values or  
5294 the description of the document if available.
- 5295

5296 For Documents that support Content Streams:

5297 The repository SHOULD use the atom:content/src attribute to point to the content stream.  
5298 The client SHOULD use cmisra:content if the content is not well-formed or would have  
5299 trouble fitting inside an atom:content element. The repository MUST use the  
5300 cmisra:content element if provided by the client over the atom:content element.

5301

5302 Other Objects (Folders, Relationships, and other Document Types that do not support Content  
5303 Streams, etc):

5304 The repository MUST comply with the atom specification and have an atom:content  
5305 element. This is repository specific. Any value in the content field MUST be ignored if the  
5306 atom entry represents a non-document object by the CMIS repository when the atom  
5307 entry is POSTed to a collection or sent to the repository via a PUT.

5308  
5309 When POSTing an Atom Document, the Atom elements MUST take precedence over the corresponding  
5310 writable CMIS property. For example, atom:title will overwrite cmis:name.

5311  
5312 At any point where an Atom document of CMIS Type is sent or returned, it must be a valid Atom Entry  
5313 document and conform to the guidelines below for a cmis type definition:

- 5314 • atom:title MUST be the cmis:displayName
- 5315 • The repository SHOULD populate the atom:summary tag with text that best represents a  
5316 summary of the object. For example, the type description if available.
- 5317 • The repository MUST comply with the atom specification and have an atom:content element. This  
5318 is repository specific. Any value in the content field MUST be ignored if the atom entry represents  
5319 a non-document object by the CMIS repository when the atom entry is POSTed to a collection or  
5320 sent to the repository via a PUT.

5321  
5322  
5323 Any atom element that is not specified is repository-specific.

### 5324 3.5.2.1 Hierarchical Atom Entries

5325 The repository SHOULD NOT provide any links to hierarchical objects if those capabilities are not  
5326 supported with the exception of getTypeDescendants which is required

5327  
5328 For atom entries that are hierarchical such as Folder Tree or Descendants, the repository MUST populate  
5329 a cmisra:children element in the atom:entry with the enclosing feed of its direct children. This pattern  
5330 continues until the depth is satisfied.

5331  
5332 The cmisra:children element MUST include an atom:feed element that contains the children entries of this  
5333 resource.

5334  
5335 If an entry does not contain cmisra:children element, then the entry MAY have children even though it is  
5336 not represented in the atom entry.

5337  
5338 For Example, here is a minimal Atom Entry with CMIS Children Extension Element:

```
5339 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
5340 <atom:entry xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"  
5341 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"  
5342 xmlns:atom="http://www.w3.org/2005/Atom"  
5343 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
5344 open.org/ns/cmisis/restatom/200908/">  
5345   <atom:author>  
5346     <atom:name>Al Brown</atom:name>  
5347   </atom:author>  
5348   <atom:content src="http://cmisexample.oasis-open.org/repl/af1d8c7f-b554-  
5349 4dfb-bfe1-1f41e4b34fef"/>  
5350   <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef</atom:id>  
5351   <atom:title type="text">CMIS Example Folder as Customer type</atom:title>
```

```

5352     <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5353     <cmisra:object>
5354         <cmis:properties>
5355             <cmis:propertyId localName="rep-cmis:objectId"
5356 propertyDefinitionId="cmis:objectId">
5357                 <cmis:value>af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef</cmis:value>
5358             </cmis:propertyId>
5359         </cmis:properties>
5360     </cmisra:object>
5361     <cmisra:pathSegment>customer</cmisra:pathSegment>
5362     <cmisra:children>
5363         <atom:feed>
5364             <atom:title type="text">CMIS Example Folder as Customer
5365 type</atom:title>
5366             <atom:author>
5367                 <atom:name>Al Brown</atom:name>
5368                 <atom:uri>http://www.ibm.com/</atom:uri>
5369                 <atom:email>albertcbrown@us.ibm.com</atom:email>
5370             </atom:author>
5371             <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5372             <atom:id>urn:uuid:ce2d65af-b246-454b-90ff-0986d9b05178</atom:id>
5373             <atom:link type="application/atom+xml;type=feed" rel="self"
5374 href="http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5375 1f41e4b34fef/3"/>
5376             <atom:link type="application/atomsvc+xml" rel="service"
5377 href="http://cmisexample.oasis-open.org/rep1//service"/>
5378             <atom:link type="application/atom+xml;type=entry" rel="via"
5379 href="http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5380 1f41e4b34fef"/>
5381             <atom:link type="application/atom+xml;type=feed"
5382 rel="http://docs.oasis-open.org/ns/cmis/link/200908/foldertree"
5383 href="http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5384 1f41e4b34fef/foldertree"/>
5385             <atom:link type="application/atom+xml;type=feed" rel="down"
5386 href="http://cmisexample.oasis-open.org/rep1/af1d8c7f-b554-4dfb-bfe1-
5387 1f41e4b34fef/children"/>
5388             <atom:link type="application/atom+xml;type=entry" rel="up"
5389 href="http://cmisexample.oasis-open.org/rep1/2eb09309-58f7-4627-b735-
5390 4d5cf4ba6554"/>
5391         <atom:entry>
5392             <atom:author>
5393                 <atom:name>Al Brown</atom:name>
5394             </atom:author>
5395             <atom:content src="http://cmisexample.oasis-
5396 open.org/rep1/af1d8c7f-b554-4dfb-bfe1-1f41e4b34fef"/>
5397             <atom:id>urn:uuid:af1d8c7f-b554-4dfb-bfe1-
5398 1f41e4b34fef</atom:id>
5399             <atom:title type="text">CMIS Example Child of
5400 Folder</atom:title>
5401             <atom:updated>2010-01-25T10:20:57.818-08:00</atom:updated>
5402             <cmisra:object>
5403                 <cmis:properties>
5404                     <cmis:propertyId localName="rep-cmis:objectId"
5405 propertyDefinitionId="cmis:objectId">
5406                         <cmis:value>af1d8c7f-b554-4dfb-bfe1-
5407 1f41e4b34fef</cmis:value>
5408                     </cmis:propertyId>
5409                 </cmis:properties>
5410             </cmisra:object>
5411             <cmisra:pathSegment>document</cmisra:pathSegment>
5412         </atom:entry>
5413     </atom:feed>
5414 </cmisra:children>
5415 </atom:entry>

```

5416

5417 Please also see the example documents included with the schema.

### 5418 **3.6 AtomPub Service Document (Repository)**

5419 The AtomPub Service Document contains the set of repositories that are available. Each repository is  
5420 mapped to a `app:workspace` element in the AtomPub Service document.

5421

5422 CMIS Services exposed:

5423 GET: `getRepositories`, `getRepositoryInfo`

5424

5425 Media Type: `application/atomsvc+xml`

5426

5427 How the client will get the initial AtomPub (APP) service document or the URI for the service document is  
5428 repository specific. Examples are via URI, or loading the service document from disk.

5429

5430 The service document will be available from Atom Entry and Atom Feed documents via a link relationship,  
5431 service. That AtomPub service document MUST contain only one workspace element which MUST be  
5432 the workspace representing the repository containing the Atom Entry or Atom Feed document.

5433

5434 A workspace element for a CMIS repository MUST have a collection element for each of following  
5435 collections: Each collection MUST also contain a `cmisra:collectionType` element with the given value:

- 5436 • Root Folder Children Collection: Root folder of the Repository
  - 5437 ○ `"/root/"` for the children collection of the root folder
  - 5438 ○ `cmisra:collectiontype="/root/"`
- 5439 • Types Children Collection: Collection containing the base types in the repository
  - 5440 ○ `"/types/"` for the children collection
  - 5441 ○ `cmisra:collectiontype="/types/"`

5442

5443 The workspace element SHOULD contain these collections if the repository supports this functionality:

- 5444 • CheckedOut collection: collection containing all checked out documents user can see
  - 5445 ○ `"/checkedout/"`
  - 5446 ○ `cmisra:collectiontype="/checkedout/"`
- 5447 • Query collection: Collection for posting queries to be executed
  - 5448 ○ `"/query/"`
  - 5449 ○ `cmisra:collectiontype="/query/"`
- 5450 • Unfiled folder: Folder for posting documents to be unfiled; read can be disabled
  - 5451 ○ `"/unfiled/"`
  - 5452 ○ `cmisra:collectiontype="/unfiled/"`

5453

5454 The repository MUST include the URI templates in the workspace elements.

5455

5456 The workspace element MUST also contain the following link element with the relation:

- 5457 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>:<http://docs.oasis-open.org/ns/cmisis/link/200908/typedescendants>: This link relation points to the types descendants  
5458 | for the base types in the repository.  
5459 |

5460

5461 | The workspace element MUST contain the following link elements of the following relations for those  
5462 | services which are supported by the repository:

- 5463 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>:<http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>: This link relation points to the folder tree of the root  
5464 | folder. See Folder Tree resource for more information.  
5465 |

- 5466 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>:<http://docs.oasis-open.org/ns/cmisis/link/200908/rootdescendants>: This link relation points to the Root Folder  
5467 | Descendants Feed for the root folder.  
5468 |

- 5469 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/changes>:<http://docs.oasis-open.org/ns/cmisis/link/200908/changes>: This link relation points to the changes feed for the  
5470 | repository.  
5471 |

5472

5473 | The workspace element may include app:collection element for the collections that represent folders in  
5474 | the repository. However, an alternative approach, especially for a repository with many folders, is to not  
5475 | enumerate those collections here, but include the app:collection element per RFC5023 in the Atom Feed  
5476 | document.

### 5477 | 3.6.1 URI Templates

5478

5479 | CMIS defines the following URI Templates:

- 5480 | • objectbyid  
5481 | • objectbypath  
5482 | • query  
5483 | • typebyid

5484

5485 | Repositories MUST provide the following URI Templates:

- 5486 | • objectbyid  
5487 | • objectbypath  
5488 | • typebyid

5489

5490 | Repositories MUST provide the URI Template query if the repository supports query.

5491

5492 | [URI Templates MUST only be used with HTTP GET.](#)

5493

5494 | Repositories MAY extend that set of templates. Those URI Template Types will be repository specific.  
5495 | Repositories MAY have more than one entry per URI Template type if the entries have different media  
5496 | types.

5497

5498 | URI Templates are simple replacement of the template parameter with the specified value. If a client  
5499 | does not want to specify a value for some of these variables, then the client MUST substitute an empty  
5500 | string for the variable.

5501

5502 For example, if the URI template that supports the variable {id} is

5503 `http://example.org/repl/getbyid/{id}`

5504

5505 | If the client wants to find the entry for an object with an id of 'obj\_4'obj\_1' then the URI would be:

5506 `http://example.org/repl/getbyid/obj_1`

5507

5508 Arguments that are substituted for URI template parameters MUST be percent escaped according to  
5509 RFC3986. Please see that RFC for more information.

5510

5511 All variables MUST be in the template.

5512

5513 Structure of URI Template:

```
5514 <xs:complexType name="cmisUriTemplateType">
5515   <xs:sequence>
5516     <xs:element name="template" type="xs:string" />
5517     <xs:element name="type" type="xs:string" />
5518     <xs:element name="mediatype" type="xs:string" />
5519     <xs:any processContents="lax" namespace="##other"
5520 minOccurs="0"
5521           maxOccurs="unbounded" />
5522   </xs:sequence>
5523 </xs:complexType>
```

5524

5525

5526 Example of URI Template element in an AtomPub Workspace Element:

```
5527 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5528 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5529 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5530 open.org/ns/cmisis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5531 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5532 open.org/ns/cmisis/restatom/200908/">
5533   <cmisra:template>http://cmisexample.oasis-
5534 open.org/repl/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
5535 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
5536 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
5537   <cmisra:type>objectbyid</cmisra:type>
5538   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5539 </cmisra:uritemplate>
```

5540

5541 Please also see the example documents included with the schema.

5542

### 5543 3.6.1.1 Object By Id

5544 This URI template provides a method for creating an URI that directly accesses an atom entry  
5545 representing documents, folders, policies or relationship objects. See section 3.10 for more information.

5546

5547 Type: objectbyid

5548 Media Type: application/atom+xml;type=entry

5549

5550 Service: getObjectById

5551

5552 Variables that are supported by the template:

- 5553 • {id}: Id of object
- 5554 • {filter}: Property Filter
- 5555 • {includeAllowableActions}
  - 5556 ○ Valid values: true, false
- 5557 • {includePolicyIds}: Include Policy Ids:
  - 5558 ○ Valid values: true, false
- 5559 • {includeRelationships}: Include relationships
  - 5560 ○ Valid values: See enumIncludeRelationships
- 5561 • {includeACL}: Include ACLs
  - 5562 ○ Valid values: true, false
- 5563 • {renditionFilter}
  - 5564 ○ Valid values: Please see renditionFilter in CMIS Domain Model

5565

5566 Example:

```
5567 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5568 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5569 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5570 open.org/ns/cmisis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5571 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5572 open.org/ns/cmisis/restatom/200908/">
5573   <cmisra:template>http://cmisexample.oasis-
5574 open.org/rep1/objectbyid/{id}?filter={filter}&includeAllowableActions={inc
5575 ludeAllowableActions}&includePolicyIds={includePolicyIds}&includeRelat
5576 ionships={includeRelationships}&includeACL={includeACL}</cmisra:template>
5577   <cmisra:type>objectbyid</cmisra:type>
5578   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5579 </cmisra:uritemplate>
5580
```

5581

5582 Please also see the example documents included with the schema.

### 5583 3.6.1.2 Object By Path

5584 This URI template provides a method for creating an URI that directly accesses an atom entry  
5585 representing documents, folders or policy objects. See section 3.10 for more information.

5586

5587 Type: objectbypath

5588 Media Type: application/atom+xml;type=entry

5589

5590 Service: getObjectByPath

5591

5592 Variables that are supported by the template:

- 5593 • {path}: Path of Object
- 5594 • {filter}: Property Filter
- 5595 • {includeAllowableActions}: Boolean for include Allowable Actions
  - 5596 ○ Valid values: true, false
- 5597 • {includePolicyIds}: Include Policy Ids:

- 5598           o Valid values: true, false
- 5599       • {includeRelationships}: Include relationships
- 5600           o Valid values: See enumIncludeRelationships
- 5601       • {includeACL}: Include ACLs
- 5602           o Valid values: true, false
- 5603       • {renditionFilter}
- 5604           o Valid values: Please see renditionFilter in CMIS Domain Model

5605  
5606

5607 Example:

```
5608 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5609 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5610 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5611 open.org/ns/cmisis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5612 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5613 open.org/ns/cmisis/restatom/200908/">
5614   <cmisra:template>http://cmisexample.oasis-
5615 open.org/repl/objectbypath?p={path}&filter={filter}&includeAllowableAc
5616 tions={includeAllowableActions}&includePolicyIds={includePolicyIds}&in
5617 cludeRelationships={includeRelationships}&includeACL={includeACL}</cmisra:
5618 template>
5619   <cmisra:type>objectbypath</cmisra:type>
5620   <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5621 </cmisra:uritemplate>
5622
```

5623

5624 Please also see the example documents included with the schema.

5625

### 5626 3.6.1.3 Query

5627 Type: query

5628 Media Type: application/atom+xml;type=feed

5629

5630 Service: query

5631

5632 Variables that are supported by the template:

- 5633       • {q}: CMIS Query Statement
- 5634       • {searchAllVersions}: Boolean, true if to search all versions
- 5635       • {maxItems}: Integer, Max items to return
- 5636       • {skipCount}: Integer, Items to skip
- 5637       • {includeAllowableActions}: Boolean
- 5638       • {includeRelationships}: Boolean

5639

5640

5641 Example:

```
5642 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```
5643 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5644 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5645 open.org/ns/cmisis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5646 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5647 open.org/ns/cmisis/restatom/200908/">
5648 <cmisra:template>http://cmisexample.oasis-
5649 open.org/rep1/query?q={q}&searchAllVersions={searchAllVersions}&maxIte
5650 ms={maxItems}&skipCount={skipCount}&includeAllowableActions={includeAl
5651 lowableActions}=&includeRelationships={includeRelationships}</cmisra:templ
5652 ate>
5653 <cmisra:type>query</cmisra:type>
5654 <cmisra:mediatype>application/atom+xml;type=feed</cmisra:mediatype>
5655 </cmisra:uritemplate>
5656
```

5657

5658 Please also see the example documents included with the schema.

5659

### 5660 3.6.1.4 Type By Id

5661 Type: typebyid

5662 Media Type: application/atom+xml;type=entry

5663

5664 Service: getTypeDefinition

5665

5666 Variables that are supported by the template:

- 5667 • {id}: CMIS Type Id

5668

5669

5670 Example:

```
5671 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5672 <cmisra:uritemplate xmlns:cmis="http://docs.oasis-
5673 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5674 open.org/ns/cmisis/messaging/200908/" xmlns:atom="http://www.w3.org/2005/Atom"
5675 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
5676 open.org/ns/cmisis/restatom/200908/">
5677 <cmisra:template>http://cmisexample.oasis-
5678 open.org/rep1/type?id={id}</cmisra:template>
5679 <cmisra:type>query</cmisra:type>
5680 <cmisra:mediatype>application/atom+xml;type=entry</cmisra:mediatype>
5681 </cmisra:uritemplate>
```

5682

5683 Please also see the example documents included with the schema.

5684

5685

## 5686 3.6.2 HTTP Methods

### 5687 3.6.2.1 GET

5688 This retrieves the AtomPub Service document for a specified repository. This exposes the capabilities  
5689 defined in getRepositories and getRepositoryInfo in the Domain Model.

5690

- 5691 The optional argument MAY be specified:
- 5692     • repositoryId:
- 5693         ○ This query parameter allows a client to specify a different repository than the one that is
- 5694         referenced by the URI.
- 5695         ○ If specified, the repository MUST return the AtomPub services document for the specified
- 5696         repository if that repository exists.
- 5697         ○ If not specified, the repository MUST return the service document for the repository that is
- 5698         referenced by URI.
- 5699

## 5700 **3.7 Service Collections**

5701 These are the collections that are included on an AtomPub Service document in the workspace element.

5702 For any HTTP verb not specified on a resource, each implementation MAY choose to implement that HTTP

5703 verb in a repository-specific manner.

### 5704 **3.7.1 Root Folder Collection**

5705 This is a collection described in the service document. Please see ~~Folder Children~~. [Folder Children](#).

### 5706 **3.7.2 Query Collection**

5707 This is a collection for processing queries. If the implementation supports GET on this collection, then the

5708 implementation SHOULD at least return a feed consisting of zero or more atom entries. These atom

5709 entries should represent persisted objects related to query such as persisted queries, long running

5710 queries or search templates.

5711

5712 CMIS Services exposed via HTTP verbs:

5713     POST: Query

5714

5715 Media Type: application/atom+xml;type=feed

5716 Accept:

- 5717     • MUST support CMIS Query document,
- 5718     • MAY support other media type

5719

5720 Link Relations on resulting feed from Query Collection:

- 5721     • service: Points to service document containing the CMIS repository. The service document
- 5722         MUST contain only one workspace element.
  - 5723             ○ Media Type: application/atomsvc+xml
- 5724     • paging link relations as appropriate: first, next, previous, last

5725

5726 The following CMIS Atom extension element MAY be included inside the atom feed:

- 5727     • cmisra:numItems

5728

5729 The following CMIS Atom extension element MUST be included inside the atom entries:

- 5730     • cmisra:object inside atom:entry

5731

5732 **3.7.2.1 POST**

5733 This collection **MUST** accept CMIS Query documents (application/cmismquery+xml).

5734

5735 Upon submission (creation) of a query document, a response must be returned with a Location header  
5736 representing the feed for that query. If the query cannot be performed and an atom feed returned, the  
5737 repository **MUST** return the appropriate HTTP status code. In addition, the server **SHOULD** return the  
5738 feed directly. If the server does so, the server should also return the Content-Location header.

5739

5740 The feed returned **MUST** contain a set of atom entries representing the result set from the query.

5741

5742 The atom entries should contain the bare minimum necessary for Atom compliance [RFC4287]. The  
5743 atom entries **MUST** contain the CMIS extension element (cmisra:object) containing the properties  
5744 specified by the query in the select clause of the query statement.

5745

5746 If all the selected properties can be mapped to the same type reference, then the repository **MAY** include  
5747 additional information in the atom entry.

5748

5749 Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

5750

5751 Status Codes:

- 201 Success

5753

5754 Headers returned:

- Location Header
- Content-Location Header

5757

5758 Link Relations on resulting feed from POST to Query Collection:

- 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

5763

5764 Example client request:

```
5765 POST /Query HTTP/1.1
5766 Host: example.org
5767 Content-Length: 756
5768 Content-Type: application/cmismquery+xml
5769
5770
5771 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5772 <cmis:query xmlns:app="http://www.w3.org/2007/app"
5773 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5774 open.org/ns/cmism/core/200908/" xmlns:cmism="http://docs.oasis-
5775 open.org/ns/cmism/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5776 open.org/ns/cmism/restatom/200908/">
5777   <cmis:statement>SELECT cmis:objectId FROM cmis:document</cmis:statement>
5778   <cmis:searchAllVersions>true</cmis:searchAllVersions>
5779   <cmis:includeAllowableActions>>false</cmis:includeAllowableActions>
5780   <cmis:includeRelationships>none</cmis:includeRelationships>
5781   <cmis:renditionFilter>*</cmis:renditionFilter>
5782   <cmis:maxItems>50</cmis:maxItems>
5783   <cmis:skipCount>0</cmis:skipCount>
```

5784 </cmis:query>

5785  
5786

5787

5788 Example server response:

```
5789 HTTP/1.1 201 Created
5790 Date: Mon, 25 Jan 2010 10:21:00 -0800
5791 Content-Length: 1830
5792 Content-Type: application/atom+xml;type=feed
5793 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/44ce5b47-
5794 ebc3-4513-86e0-d3f46c77d0a8
5795 Location: http://cmisexample.oasis-open.org/rep1/queryresult/44ce5b47-ebc3-
5796 4513-86e0-d3f46c77d0a8
5797
5798
5799 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5800 <atom:feed xmlns:app="http://www.w3.org/2007/app"
5801 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5802 open.org/ns/cmismessaging/200908/" xmlns:cmism="http://docs.oasis-
5803 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
5804 open.org/ns/cmismessaging/200908/">
5805   <atom:title type="text">CMIS Query Result for SELECT cmis:objectId FROM
5806 cmis:document</atom:title>
5807   <atom:author>
5808     <atom:name>Al Brown</atom:name>
5809     <atom:uri>http://www.ibm.com/</atom:uri>
5810     <atom:email>albertcbrown@us.ibm.com</atom:email>
5811   </atom:author>
5812   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
5813   <atom:id>urn:uuid:811b1b9b-80f5-4788-b46c-aa77564e294b</atom:id>
5814   <atom:link type="application/atom+xml;type=feed" rel="self"
5815 href="http://cmisexample.oasis-open.org/rep1/11355977-434b-4e71-b83a-
5816 77dea9878e04/3"/>
5817   <atom:link type="application/atomsvc+xml" rel="service"
5818 href="http://cmisexample.oasis-open.org/rep1//service"/>
5819   <atom:entry>
5820     <atom:author>
5821       <atom:name>Al Brown</atom:name>
5822     </atom:author>
5823     <atom:content src="http://cmisexample.oasis-open.org/rep1/a3386ea0-
5824 0477-4a74-96bd-70d3dalc483a"/>
5825     <atom:id>urn:uuid:a3386ea0-0477-4a74-96bd-70d3dalc483a</atom:id>
5826     <atom:title type="text">Resulting Document</atom:title>
5827     <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
5828     <cmisra:object>
5829       <cmis:properties>
5830         <cmis:propertyId queryName="cmis:objectId" localName="rep-
5831 cmis:objectId" propertyDefinitionId="cmis:objectId">
5832           <cmis:value>a3386ea0-0477-4a74-96bd-
5833 70d3dalc483a</cmis:value>
5834         </cmis:propertyId>
5835       </cmis:properties>
5836     </cmisra:object>
5837   </atom:entry>
5838 </atom:feed>
5839
```

5840

5841 Please also see the example documents included with the schema.

5842

### 5843 3.7.3 Checked Out Collection

5844 This is a collection described in the service document that contains the private working copies (PWCs) of  
5845 the checkedout documents in the repository.

5846 CMIS Services:

5847 GET: getCheckedOutDocs

5848 POST: checkOut

5849 Media Type: application/atom+xml;type=feed

5850 Accept:

5851 • MUST support Atom Entry Documents with CMIS extensions

5852     ○ application/atom+xml;type=entry or

5853     ○ application/cmisaatom+xml

5854 • MAY support other media type

5855

5856 Link Relations:

5857 • service: Points to service document containing the CMIS repository. The service document  
5858 MUST contain only one workspace element.

5859     ○ Media Type: application/atomsvc+xml

5860 • paging link relations as appropriate: first, next, previous, last

5861

5862 The following CMIS Atom extension element MAY be included inside the atom feed:

5863 | • cmisra:numItems

5864

5865 The following CMIS Atom extension element MUST be included inside the atom entries:

5866 | • cmisra:object inside atom:entry

5867

#### 5868 3.7.3.1 GET

5869 The following arguments may be supplied. Please see the domain model for more information:

5870 • filter

5871 • folderId

5872 • maxItems

5873 • skipCount

5874 • renditionFilter

5875 • includeAllowableActions

5876 • includeRelationships

#### 5877 3.7.3.2 POST

5878 | When an atom entry is POST<sup>ed</sup> to this collection, the atom entry will be checked out. A Content-  
5879 Location header MUST be returned containing the location of the private working copy.

5880

5881 Example client request:

5882 POST /CheckedOut HTTP/1.1  
5883 Host: example.org

```
5884 Content-Length: 1044
5885 Content-Type: application/atom+xml;type=entry
5886
5887
5888 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5889 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5890 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5891 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5892 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5893 open.org/ns/cmisis/restatom/200908/">
5894   <atom:author>
5895     <atom:name>Al Brown</atom:name>
5896   </atom:author>
5897   <atom:content src="http://cmisexample.oasis-open.org/rep1/8d32d716-701b-
5898 4491-84e8-ad57c8230940"/>
5899   <atom:id>urn:uuid:8d32d716-701b-4491-84e8-ad57c8230940</atom:id>
5900   <atom:title type="text">CMIS Example Document to checkout</atom:title>
5901   <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
5902   <cmisra:object>
5903     <cmis:properties>
5904       <cmis:propertyId localName="rep-cmis:objectId"
5905 propertyDefinitionId="cmis:objectId">
5906         <cmis:value>8d32d716-701b-4491-84e8-ad57c8230940</cmis:value>
5907       </cmis:propertyId>
5908     </cmis:properties>
5909   </cmisra:object>
5910 </atom:entry>
5911
```

5912

5913 Example server response:

```
5914 HTTP/1.1 201 Created
5915 Date: Mon, 25 Jan 2010 10:21:00 -0800
5916 Content-Length: 7846
5917 Content-Type: application/atom+xml;type=entry
5918 Content-Location: http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-
5919 8fab-4aa6e6797dbe
5920 Location: http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5921 4aa6e6797dbe
5922
5923
5924 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
5925 <atom:entry xmlns:app="http://www.w3.org/2007/app"
5926 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
5927 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
5928 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
5929 open.org/ns/cmisis/restatom/200908/">
5930   <atom:author>
5931     <atom:name>Al Brown</atom:name>
5932     <atom:uri>http://www.ibm.com/</atom:uri>
5933     <atom:email>albertcbrown@us.ibm.com</atom:email>
5934   </atom:author>
5935   <atom:content src="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-
5936 491c-8fab-4aa6e6797dbe"/>
5937   <atom:id>urn:uuid:6cce57fc-4e31-491c-8fab-4aa6e6797dbe</atom:id>
5938   <atom:title type="text">CMIS Example Child of Folder</atom:title>
5939   <atom:updated>2010-01-25T10:21:00.396-08:00</atom:updated>
5940   <atom:link rel="self" href="http://cmisexample.oasis-
5941 open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
5942   <atom:link rel="edit" href="http://cmisexample.oasis-
5943 open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe"/>
```

```

5944     <atom:link type="application/cmism+xml;type=allowableActions"
5945     rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
5946     href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5947     4aa6e6797dbe/allowableactions"/>
5948     <atom:link type="application/atom+xml;type=entry" rel="describedby"
5949     href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5950     4aa6e6797dbe/type"/>
5951     <atom:link type="application/atomsvc+xml" rel="service"
5952     href="http://cmisexample.oasis-open.org/rep1//service"/>
5953     <atom:published>2010-01-25T10:21:00.396-08:00</atom:published>
5954     <atom:summary type="html">HTML summary of Entry 6cce57fc-4e31-491c-8fab-
5955     4aa6e6797dbe</atom:summary>
5956     <atom:link rel="edit-media" href="http://cmisexample.oasis-
5957     open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/edit-media"/>
5958     <atom:link rel="alternate" href="http://cmisexample.oasis-
5959     open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/alternate"/>
5960     <atom:link type="application/atom+xml;type=feed" rel="up"
5961     href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5962     4aa6e6797dbe/parents"/>
5963     <atom:link type="application/atom+xml;type=feed" rel="version-history"
5964     href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5965     4aa6e6797dbe/allversions"/>
5966     <atom:link type="application/atom+xml;type=entry" rel="current-version"
5967     href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5968     4aa6e6797dbe/latest"/>
5969     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
5970     open.org/ns/cmism/link/200908/relationships" href="http://cmisexample.oasis-
5971     open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/relationships"/>
5972     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
5973     open.org/ns/cmism/link/200908/policies" href="http://cmisexample.oasis-
5974     open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/policies"/>
5975     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
5976     open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
5977     open.org/rep1/6cce57fc-4e31-491c-8fab-4aa6e6797dbe/acl"/>
5978     <atom:link type="application/atom+xml;type=feed" rel="working-copy"
5979     href="http://cmisexample.oasis-open.org/rep1/6cce57fc-4e31-491c-8fab-
5980     4aa6e6797dbe/pwc"/>
5981     <cmisra:object>
5982         <cmis:properties>
5983             <cmis:propertyId localName="rep-cmism:objectId"
5984             propertyDefinitionId="cmism:objectId">
5985                 <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
5986             </cmis:propertyId>
5987             <cmis:propertyId localName="rep-cmism:objectTypeId"
5988             propertyDefinitionId="cmism:objectTypeId">
5989                 <cmis:value>customer</cmis:value>
5990             </cmis:propertyId>
5991             <cmis:propertyString localName="rep-cmism:name"
5992             propertyDefinitionId="cmism:name">
5993                 <cmis:value>CMIS Example Child of Folder</cmis:value>
5994             </cmis:propertyString>
5995             <cmis:propertyDateTime localName="rep-cmism:creationDate"
5996             propertyDefinitionId="cmism:creationDate">
5997                 <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
5998             </cmis:propertyDateTime>
5999             <cmis:propertyDateTime localName="rep-cmism:lastModificationDate"
6000             propertyDefinitionId="cmism:lastModificationDate">
6001                 <cmis:value>2010-01-25T10:21:00.396-08:00</cmis:value>
6002             </cmis:propertyDateTime>
6003             <cmis:propertyId localName="rep-cmism:baseTypeId"
6004             propertyDefinitionId="cmism:baseTypeId">
6005                 <cmis:value>cmism:document</cmis:value>
6006             </cmis:propertyId>

```

```

6007         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6008 propertyDefinitionId="cmis:lastModifiedBy">
6009             <cmis:value>Al Brown</cmis:value>
6010         </cmis:propertyString>
6011         <cmis:propertyString localName="rep-cmis:createdBy"
6012 propertyDefinitionId="cmis:createdBy">
6013             <cmis:value>Al Brown</cmis:value>
6014         </cmis:propertyString>
6015         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6016 propertyDefinitionId="cmis:isLatestVersion">
6017             <cmis:value>>true</cmis:value>
6018         </cmis:propertyBoolean>
6019         <cmis:propertyBoolean localName="rep-
6020 cmis:isVersionSeriesCheckedOut"
6021 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6022             <cmis:value>>true</cmis:value>
6023         </cmis:propertyBoolean>
6024         <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6025 propertyDefinitionId="cmis:isMajorVersion">
6026             <cmis:value>>false</cmis:value>
6027         </cmis:propertyBoolean>
6028         <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6029 propertyDefinitionId="cmis:isLatestMajorVersion">
6030             <cmis:value>>false</cmis:value>
6031         </cmis:propertyBoolean>
6032         <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6033 propertyDefinitionId="cmis:isImmutable">
6034             <cmis:value>>false</cmis:value>
6035         </cmis:propertyBoolean>
6036         <cmis:propertyString localName="rep-cmis:checkinComment"
6037 propertyDefinitionId="cmis:checkinComment">
6038             <cmis:value>Checkin comment</cmis:value>
6039         </cmis:propertyString>
6040         <cmis:propertyString localName="rep-cmis:versionLabel"
6041 propertyDefinitionId="cmis:versionLabel">
6042             <cmis:value>0.1</cmis:value>
6043         </cmis:propertyString>
6044         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6045 propertyDefinitionId="cmis:contentStreamMimeType">
6046             <cmis:value>text/plain</cmis:value>
6047         </cmis:propertyString>
6048         <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6049 propertyDefinitionId="cmis:contentStreamFileName">
6050             <cmis:value>text.txt</cmis:value>
6051         </cmis:propertyString>
6052         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6053 propertyDefinitionId="cmis:contentStreamLength">
6054             <cmis:value>4234</cmis:value>
6055         </cmis:propertyInteger>
6056         <cmis:propertyString displayName="Keywords for Document"
6057 localName="keywords" propertyDefinitionId="keywords">
6058             <cmis:value>document</cmis:value>
6059             <cmis:value>example</cmis:value>
6060             <cmis:value>sample</cmis:value>
6061             <cmis:value>cmis</cmis:value>
6062         </cmis:propertyString>
6063         <cmis:propertyId localName="rep-cmis:versionSeriesCheckedOutId"
6064 propertyDefinitionId="cmis:versionSeriesCheckedOutId">
6065             <cmis:value>6cce57fc-4e31-491c-8fab-4aa6e6797dbe</cmis:value>
6066         </cmis:propertyId>
6067         <cmis:propertyString localName="rep-
6068 cmis:versionSeriesCheckedOutBy"
6069 propertyDefinitionId="cmis:versionSeriesCheckedOutBy">
6070             <cmis:value>Al Brown</cmis:value>

```

6071  
6072  
6073  
6074  
6075

```
</cmis:propertyString>  
</cmis:properties>  
</cmisra:object>  
</atom:entry>
```

6076

6077 Please also see the example documents included with the schema.

6078

### 6079 **3.7.4 Unfiled Collection**

6080 This is a collection described in the service document that contains all the unfiled documents in the  
6081 repository. This collection **MUST** be available if un-filing or multi-filing is supported by the repository.

6082 A repository that supports un-filing **MAY** provide read access (GET). If read access is not provided, the  
6083 repository **SHOULD** respond to a read attempt with the HTTP status code 405 (notSupported).

6084 CMIS Services:

6085       POST: removeObjectFromFolder

6086 Media Type: application/atom+xml;type=feed

6087 Accept:

6088       • **MUST** support Atom Entry Documents with CMIS extensions

6089           ○ application/atom+xml;type=entry or

6090           ○ application/cmisaatom+xml

6091       • **MAY** support other media type

6092

6093 Link Relations:

6094       • **service**: Points to service document containing the CMIS repository. The service document  
6095       **MUST** contain only one workspace element.

6096           ○ Media Type: application/atomsvc+xml

6097       • **paging link relations** as appropriate: first, next, previous, last

6098

6099 The following CMIS Atom extension element **MAY** be included inside the atom feed:

6100       • cmisra:numItems

6101

6102 The following CMIS Atom extension element **MUST** be included inside the atom entries:

6103       • cmisra:object inside atom:entry

6104

#### 6105 **3.7.4.1 POST**

6106 This removes the object from all folders in the repository by default. If the optional argument removeFrom  
6107 is specified, the object will only be removed from that folder only.

6108

6109 | If the Atom Entry **POST**ed, does not have the CMIS extensions with a valid cmis:objectId property, the  
6110 document does not exist, or the document is not in that folder, the appropriate HTTP status code **MUST**  
6111 be returned.

6112

6113 This adheres to AtomPub model. Please see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 6114 • HTTP Success: 201
- 6115 • Location Header

6116

6117 The following arguments may be supplied. Please see the domain model for more information:

- 6118 • **removeFrom:** For repositories which support multi-filing, this parameter identifies which folder to  
6119 remove this object from. If specified, it indicates the folder from which the object shall be moved.  
6120 If not specified, the object will be removed from all folders.

6121

6122 Example client request:

```
6123 POST /Unfiled HTTP/1.1
6124 Host: example.org
6125 Content-Length: 1043
6126 Content-Type: application/atom+xml;type=entry
6127
6128
6129 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6130 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6131 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6132 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
6133 open.org/ns/cmismessaging/200908/" xmlns:cmis="http://docs.oasis-
6134 open.org/ns/cmismessaging/200908/">
6135   <atom:author>
6136     <atom:name>Al Brown</atom:name>
6137   </atom:author>
6138   <atom:content src="http://cmisexample.oasis-open.org/rep1/12aa2bec-6f43-
6139 47d1-99ef-21797867173c"/>
6140   <atom:id>urn:uuid:12aa2bec-6f43-47d1-99ef-21797867173c</atom:id>
6141   <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6142   <atom:updated>2010-01-25T10:21:00.427-08:00</atom:updated>
6143   <cmisra:object>
6144     <cmis:properties>
6145       <cmis:propertyId localName="rep-cmis:objectId"
6146 propertyDefinitionId="cmis:objectId">
6147         <cmis:value>12aa2bec-6f43-47d1-99ef-21797867173c</cmis:value>
6148       </cmis:propertyId>
6149     </cmis:properties>
6150   </cmisra:object>
6151 </atom:entry>
6152
```

6153

6154 Example server response:

```
6155 HTTP/1.1 201 Created
6156 Date: Mon, 25 Jan 2010 10:21:00 -0800
6157 Content-Length: 7234
6158 Content-Type: application/atom+xml;type=entry
6159 Content-Location: http://cmisexample.oasis-open.org/rep1/queryresult/15118373-
6160 8911-442b-9774-da3b102f224c
6161 Location: http://cmisexample.oasis-open.org/rep1/queryresult/15118373-8911-
6162 442b-9774-da3b102f224c
6163
6164
6165 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6166 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6167 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6168 open.org/ns/cmismessaging/200908/" xmlns:cmisra="http://docs.oasis-
6169 open.org/ns/cmismessaging/200908/" xmlns:cmis="http://docs.oasis-
6170 open.org/ns/cmismessaging/200908/">
6171   <atom:author>
6172     <atom:name>Al Brown</atom:name>
```

```

6173     <atom:uri>http://www.ibm.com/</atom:uri>
6174     <atom:email>albertcbrown@us.ibm.com</atom:email>
6175     </atom:author>
6176     <atom:content src="http://cmisexample.oasis-open.org/repl/15118373-8911-
6177 442b-9774-da3b102f224c"/>
6178     <atom:id>urn:uuid:15118373-8911-442b-9774-da3b102f224c</atom:id>
6179     <atom:title type="text">CMIS Example Document to unfiled</atom:title>
6180     <atom:updated>2010-01-25T10:21:00.443-08:00</atom:updated>
6181     <atom:link rel="self" href="http://cmisexample.oasis-
6182 open.org/repl/15118373-8911-442b-9774-da3b102f224c"/>
6183     <atom:link rel="edit" href="http://cmisexample.oasis-
6184 open.org/repl/15118373-8911-442b-9774-da3b102f224c"/>
6185     <atom:link type="application/cmisacl+xml;type=allowableActions"
6186 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions"
6187 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6188 da3b102f224c/allowableactions"/>
6189     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6190 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6191 da3b102f224c/type"/>
6192     <atom:link type="application/atomsvc+xml" rel="service"
6193 href="http://cmisexample.oasis-open.org/repl//service"/>
6194     <atom:published>2010-01-25T10:21:00.443-08:00</atom:published>
6195     <atom:summary type="html">HTML summary of Entry 15118373-8911-442b-9774-
6196 da3b102f224c</atom:summary>
6197     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6198 open.org/repl/15118373-8911-442b-9774-da3b102f224c/edit-media"/>
6199     <atom:link rel="alternate" href="http://cmisexample.oasis-
6200 open.org/repl/15118373-8911-442b-9774-da3b102f224c/alternate"/>
6201     <atom:link type="application/atom+xml;type=feed" rel="up"
6202 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6203 da3b102f224c/parents"/>
6204     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6205 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6206 da3b102f224c/allversions"/>
6207     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6208 href="http://cmisexample.oasis-open.org/repl/15118373-8911-442b-9774-
6209 da3b102f224c/latest"/>
6210     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6211 open.org/ns/cmisacl/link/200908/relationships" href="http://cmisexample.oasis-
6212 open.org/repl/15118373-8911-442b-9774-da3b102f224c/relationships"/>
6213     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6214 open.org/ns/cmisacl/link/200908/policies" href="http://cmisexample.oasis-
6215 open.org/repl/15118373-8911-442b-9774-da3b102f224c/policies"/>
6216     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6217 open.org/ns/cmisacl/link/200908/acl" href="http://cmisexample.oasis-
6218 open.org/repl/15118373-8911-442b-9774-da3b102f224c/acl"/>
6219     <cmisra:object>
6220       <cmis:properties>
6221         <cmis:propertyId localName="rep-cmis:objectId"
6222 propertyDefinitionId="cmis:objectId">
6223           <cmis:value>15118373-8911-442b-9774-da3b102f224c</cmis:value>
6224         </cmis:propertyId>
6225         <cmis:propertyId localName="rep-cmis:objectTypeId"
6226 propertyDefinitionId="cmis:objectTypeId">
6227           <cmis:value>customer</cmis:value>
6228         </cmis:propertyId>
6229         <cmis:propertyString localName="rep-cmis:name"
6230 propertyDefinitionId="cmis:name">
6231           <cmis:value>CMIS Example Document to unfiled</cmis:value>
6232         </cmis:propertyString>
6233         <cmis:propertyDateTime localName="rep-cmis:creationDate"
6234 propertyDefinitionId="cmis:creationDate">
6235           <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
6236         </cmis:propertyDateTime>

```

```

6237         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6238 propertyDefinitionId="cmis:lastModificationDate">
6239         <cmis:value>2010-01-25T10:21:00.443-08:00</cmis:value>
6240         </cmis:propertyDateTime>
6241         <cmis:propertyId localName="rep-cmis:baseTypeId"
6242 propertyDefinitionId="cmis:baseTypeId">
6243         <cmis:value>cmis:document</cmis:value>
6244         </cmis:propertyId>
6245         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6246 propertyDefinitionId="cmis:lastModifiedBy">
6247         <cmis:value>Al Brown</cmis:value>
6248         </cmis:propertyString>
6249         <cmis:propertyString localName="rep-cmis:createdBy"
6250 propertyDefinitionId="cmis:createdBy">
6251         <cmis:value>Al Brown</cmis:value>
6252         </cmis:propertyString>
6253         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6254 propertyDefinitionId="cmis:isLatestVersion">
6255         <cmis:value>>true</cmis:value>
6256         </cmis:propertyBoolean>
6257         <cmis:propertyBoolean localName="rep-
6258 cmis:isVersionSeriesCheckedOut"
6259 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6260         <cmis:value>>false</cmis:value>
6261         </cmis:propertyBoolean>
6262         <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6263 propertyDefinitionId="cmis:isMajorVersion">
6264         <cmis:value>>false</cmis:value>
6265         </cmis:propertyBoolean>
6266         <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6267 propertyDefinitionId="cmis:isLatestMajorVersion">
6268         <cmis:value>>false</cmis:value>
6269         </cmis:propertyBoolean>
6270         <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6271 propertyDefinitionId="cmis:isImmutable">
6272         <cmis:value>>false</cmis:value>
6273         </cmis:propertyBoolean>
6274         <cmis:propertyString localName="rep-cmis:checkinComment"
6275 propertyDefinitionId="cmis:checkinComment">
6276         <cmis:value>Checkin comment</cmis:value>
6277         </cmis:propertyString>
6278         <cmis:propertyString localName="rep-cmis:versionLabel"
6279 propertyDefinitionId="cmis:versionLabel">
6280         <cmis:value>0.1</cmis:value>
6281         </cmis:propertyString>
6282         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6283 propertyDefinitionId="cmis:contentStreamMimeType">
6284         <cmis:value>text/plain</cmis:value>
6285         </cmis:propertyString>
6286         <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6287 propertyDefinitionId="cmis:contentStreamFileName">
6288         <cmis:value>text.txt</cmis:value>
6289         </cmis:propertyString>
6290         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6291 propertyDefinitionId="cmis:contentStreamLength">
6292         <cmis:value>4234</cmis:value>
6293         </cmis:propertyInteger>
6294         <cmis:propertyString displayName="Keywords for Document"
6295 localName="keywords" propertyDefinitionId="keywords">
6296         <cmis:value>document</cmis:value>
6297         <cmis:value>example</cmis:value>
6298         <cmis:value>sample</cmis:value>
6299         <cmis:value>cmis</cmis:value>
6300         </cmis:propertyString>

```

```
6301     </cmis:properties>
6302     </cmisra:object>
6303 </atom:entry>
6304
```

6305  
6306 Please also see the example documents included with the schema.  
6307

### 6308 3.7.5 Types Children Collection

6309 This is a collection described in the service document that contains the types in the repository under the  
6310 specified parent type. If no parent type is specified, then the base types are returned in the feed. This  
6311 feed does not include any nesting and is a flat feed.

6312 CMIS Services:

6313 GET: getTypeChildren

6314 Media Type: application/atom+xml;type=feed

6315

6316 Link Relations:

- 6317 | • service: Points to service document containing the CMIS repository. The service document  
6318 | MUST contain only one workspace element.
  - 6319 | ○ Media Type: application/atomsvc+xml
- 6320 | • via: points to the type definition entry whose children represent this feed
- 6321 | • down: points to the atom feed document representing the descendents collection for this same  
6322 | type with media type of application/cmistree+xml
- 6323 | • paging link relations as appropriate: first, next, previous, last
- 6324 | • up: points to the parent type definition
  - 6325 | ○ If this is a children feed for a base object type, this link is not present.

6326

6327 This feed contains a set of atom entries for each child type definition.

6328

6329 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6330 | • cmisra:numItems

6331

6332 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6333 | • cmisra:type inside atom:entry

6334

6335

#### 6336 3.7.5.1 GET

6337 The following arguments may be supplied. Please see the domain model for more information:

- 6338 | • includePropertyDefinitions
- 6339 | • maxItems
- 6340 | • skipCount
- 6341 | • typeId

## 6342 **3.8 Collections**

6343 For any HTTP verb not specified on a resource, each implementation MAY choose to implement that HTTP  
6344 verb in a repository-specific manner.

6345

### 6346 **3.8.1 Relationships Collection**

6347 This is the set of relationships available (either source or target or both) from a specific item such as a  
6348 document, folder or policy.

6349 CMIS Services:

6350 GET: getObjectRelationships

6351 POST: createRelationship

6352 Media Type: application/atom+xml;type=feed

6353 Accept:

- 6354 • MUST support Atom Entry Documents with CMIS extensions
  - 6355 ○ application/atom+xml;type=entry or
  - 6356 ○ application/cmisaatom+xml
- 6357 • MAY support other media type

6358

6359 Link Relations:

- 6360 • service: Points to service document containing the CMIS repository. The service document  
6361 MUST contain only one workspace element.
  - 6362 ○ Media Type: application/atomsvc+xml
- 6363 • paging link relations as appropriate: first, next, previous, last

6364

6365 The following CMIS Atom extension element MAY be included inside the atom feed:

- 6366 • cmisra:numItems

6367

6368 The following CMIS Atom extension element MUST be included inside the atom entries:

- 6369 • cmisra:object inside atom:entry

6370

#### 6371 **3.8.1.1 GET**

6372 The following arguments may be supplied. Please see the domain model for more information:

- 6373 • typeId
- 6374 • includeSubRelationshipTypes
- 6375 • relationshipDirection
- 6376 • maxItems
- 6377 • skipCount
- 6378 • filter
- 6379 • includeAllowableActions

### 6380 3.8.1.2 POST

6381 When an atom entry with CMIS markup is posted to this collection, if that atom entry represents a new  
6382 CMIS relationship, then that relationship will be created.

6383 The server MUST return the appropriate HTTP status code if the source is different than the sourceId or  
6384 target different than the targetId for the source and targets specified in this collection.

6385 The server MUST return the appropriate status code if the cmis:objectId is not specified.

6386

6387 Example client request:

```
6388 POST /relationships/source/dbf0316c-47b5-47c9-a2fa-f005eb93f0a4 HTTP/1.1
6389 Host: example.org
6390 Content-Length: 1432
6391 Content-Type: application/atom+xml;type=entry
6392
6393
6394 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6395 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6396 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6397 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6398 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6399 open.org/ns/cmis/restatom/200908/"
6400 <atom:author>
6401 <atom:name>Al Brown</atom:name>
6402 </atom:author>
6403 <atom:content src="http://cmisexample.oasis-open.org/rep1/dab97641-8c94-
6404 4a12-a604-7532980f05cb"/>
6405 <atom:id>urn:uuid:dab97641-8c94-4a12-a604-7532980f05cb</atom:id>
6406 <atom:title type="text">New Relationship</atom:title>
6407 <atom:updated>2010-01-25T10:20:58.864-08:00</atom:updated>
6408 <cmisra:object>
6409 <cmis:properties>
6410 <cmis:propertyId localName="rep-cmis:objectId"
6411 propertyDefinitionId="cmis:objectId">
6412 <cmis:value>customerRelationships</cmis:value>
6413 </cmis:propertyId>
6414 <cmis:propertyId localName="rep-cmis:sourceId"
6415 propertyDefinitionId="cmis:sourceId">
6416 <cmis:value>dbf0316c-47b5-47c9-a2fa-f005eb93f0a4</cmis:value>
6417 </cmis:propertyId>
6418 <cmis:propertyId localName="rep-cmis:targetId"
6419 propertyDefinitionId="cmis:targetId">
6420 <cmis:value>b9baac7d-7584-445e-bcd1-29af9b25bf2f</cmis:value>
6421 </cmis:propertyId>
6422 </cmis:properties>
6423 </cmisra:object>
6424 </atom:entry>
6425
```

6426

6427 Example server response:

```
6428 HTTP/1.1 201 Created
6429 Date: Mon, 25 Jan 2010 10:20:58 -0800
6430 Content-Length: 4684
6431 Content-Type: application/atom+xml;type=entry
6432 Content-Location: http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-
6433 86df-3f4b157bb495
6434 Location: http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6435 3f4b157bb495
6436
6437
6438 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
```

```

6439 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6440 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6441 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
6442 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6443 open.org/ns/cmisis/restatom/200908/">
6444   <atom:author>
6445     <atom:name>Al Brown</atom:name>
6446     <atom:uri>http://www.ibm.com/</atom:uri>
6447     <atom:email>albertcbrown@us.ibm.com</atom:email>
6448   </atom:author>
6449   <atom:content src="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-
6450 4c27-86df-3f4b157bb495"/>
6451   <atom:id>urn:uuid:b3006a8f-345b-4c27-86df-3f4b157bb495</atom:id>
6452   <atom:title type="text">New Relationship</atom:title>
6453   <atom:updated>2010-01-25T10:20:58.880-08:00</atom:updated>
6454   <atom:link rel="self" href="http://cmisexample.oasis-
6455 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
6456   <atom:link rel="edit" href="http://cmisexample.oasis-
6457 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495"/>
6458   <atom:link type="application/cmisis+xml;type=allowableActions"
6459 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
6460 href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6461 3f4b157bb495/allowableactions"/>
6462   <atom:link type="application/atom+xml;type=entry" rel="describedby"
6463 href="http://cmisexample.oasis-open.org/rep1/b3006a8f-345b-4c27-86df-
6464 3f4b157bb495/type"/>
6465   <atom:link type="application/atomsvc+xml" rel="service"
6466 href="http://cmisexample.oasis-open.org/rep1//service"/>
6467   <atom:published>2010-01-25T10:20:58.880-08:00</atom:published>
6468   <atom:summary type="html">HTML summary of Entry b3006a8f-345b-4c27-86df-
6469 3f4b157bb495</atom:summary>
6470   <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
6471 open.org/ns/cmisis/link/200908/source" href="http://cmisexample.oasis-
6472 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/source"/>
6473   <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
6474 open.org/ns/cmisis/link/200908/target" href="http://cmisexample.oasis-
6475 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/target"/>
6476   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6477 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
6478 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/policies"/>
6479   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
6480 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
6481 open.org/rep1/b3006a8f-345b-4c27-86df-3f4b157bb495/acl"/>
6482   <cmisra:object>
6483     <cmis:properties>
6484       <cmis:propertyId localName="rep-cmisis:objectId"
6485 propertyDefinitionId="cmisis:objectId">
6486         <cmis:value>b3006a8f-345b-4c27-86df-3f4b157bb495</cmis:value>
6487       </cmis:propertyId>
6488       <cmis:propertyId localName="rep-cmisis:objectTypeId"
6489 propertyDefinitionId="cmisis:objectTypeId">
6490         <cmis:value>customerRelationships</cmis:value>
6491       </cmis:propertyId>
6492       <cmis:propertyString localName="rep-cmisis:name"
6493 propertyDefinitionId="cmisis:name">
6494         <cmis:value>New Relationship</cmis:value>
6495       </cmis:propertyString>
6496       <cmis:propertyDateTime localName="rep-cmisis:creationDate"
6497 propertyDefinitionId="cmisis:creationDate">
6498         <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>
6499       </cmis:propertyDateTime>
6500       <cmis:propertyDateTime localName="rep-cmisis:lastModificationDate"
6501 propertyDefinitionId="cmisis:lastModificationDate">
6502         <cmis:value>2010-01-25T10:20:58.880-08:00</cmis:value>

```

```

6503         </cmis:propertyDateTime>
6504         <cmis:propertyId localName="rep-cmis:baseTypeId"
6505 propertyDefinitionId="cmis:baseTypeId">
6506             <cmis:value>cmis:relationship</cmis:value>
6507         </cmis:propertyId>
6508         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6509 propertyDefinitionId="cmis:lastModifiedBy">
6510             <cmis:value>Al Brown</cmis:value>
6511         </cmis:propertyString>
6512         <cmis:propertyString localName="rep-cmis:createdBy"
6513 propertyDefinitionId="cmis:createdBy">
6514             <cmis:value>Al Brown</cmis:value>
6515         </cmis:propertyString>
6516         <cmis:propertyId localName="rep-cmis:sourceId"
6517 propertyDefinitionId="cmis:sourceId">
6518             <cmis:value>d4551c6d-30bd-4fc2-9c84-a55f11559e89</cmis:value>
6519         </cmis:propertyId>
6520         <cmis:propertyId localName="rep-cmis:targetId"
6521 propertyDefinitionId="cmis:targetId">
6522             <cmis:value>fe7e056f-c4bf-42f1-a03e-3ababcf2491d</cmis:value>
6523         </cmis:propertyId>
6524     </cmis:properties>
6525 </cmisra:object>
6526 </atom:entry>
6527

```

6528

6529 Please also see the example documents included with the schema.

6530

### 6531 **3.8.2 Folder Children Collection**

6532 This is a collection comprised of all the direct children of a particular folder represented as a feed.

6533 CMIS Services:

6534 GET: getChildren

6535 POST:

6536 createDocument

6537 or createFolder

6538 or createPolicy

6539 or moveObject

6540 or addObjectToFolder

6541

6542 Media Type: application/atom+xml;type=feed

6543

6544 Accept:

6545 | • MUST support Atom Entry Documents with CMIS extensions

6546 | • MAY support other media type

6547

6548 Link Relations:

6549 | • service: Points to service document containing the CMIS repository. The service document  
6550 | MUST contain only one workspace element.

6551 |     o Media Type: application/atomsvc+xml

6552 | • via: points to the atom entry of the folder generating this collection

- 6553 | • up: points to the atom entry document for this folder's parent
- 6554 |     ○ If the root folder, this link relation MUST NOT be included.
- 6555 |     ○ Media Type: application/atom+xml;type=entry
- 6556 | • down: points to the atom feed document representing the descendents feed with a media type of
- 6557 |     application/cmistree+xml
- 6558 |     ○ If a repository does not support capabilityGetDescendants, then this link SHOULD NOT
- 6559 |     be included.
- 6560 | • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>~~: [http://docs.oasis-](http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree)
- 6561 |     ~~[open.org/ns/cmisis/link/200908/foldertree](http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree)~~: Points to the folder tree for this folder. This is
- 6562 |     represented as a feed with CMIS hierarchy extensions.
- 6563 |     ○ Media Type: application/atom+xml;type=feed
- 6564 | • paging link relations as appropriate: first, next, previous, last
- 6565 |

6566 | The following CMIS Atom extension element MAY be included inside the atom feed:

- 6567 | • cmisra:numItems
- 6568 |

6569 | The following CMIS Atom extension element MUST be included inside the atom entries:

- 6570 | • cmisra:object inside atom:entry
- 6571 | • cmisra:pathSegment inside atom:entry if pathSegment is not false
- 6572 |

### 6573 | 3.8.2.1 GET

6574 | HTTP Code:

- 6575 | • 200 OK (Success)
- 6576 |

6577 | The following arguments may be supplied. Please see the domain model for more information:

- 6578 | • maxItems
- 6579 | • skipCount
- 6580 | • filter
- 6581 | • includeAllowableActions
- 6582 | • includeRelationships
- 6583 | • renditionFilter
- 6584 |     ○ If specified, renditions will be returned as links with relation alternate.
- 6585 | • orderBy
- 6586 | • includePathSegment

### 6587 | 3.8.2.2 POST

6588 | CMIS repositories MUST be compliant with RFC5023 for POSTing new entries into a collection. Please

6589 | see <http://tools.ietf.org/html/rfc5023#section-5.3>.

- 6590 | • HTTP Success: 201
- 6591 | • Location Header
- 6592 |

6593 | The following arguments MAY be supplied.

- 6594 | • sourceFolderId: This parameter indicates the folder from which the object shall be moved from to
- 6595 |     the current specified folder. This parameter is not allowed for create operations.
- 6596 |     ○ If specified moveObject will be performed.

- 6597 ○ If not specified, addObjectToFolder will be performed.
- 6598 ● versioningState: The optional argument versioningState MAY specify additional versioning
- 6599 behavior such as checkIn as major or minor. Please see CMIS Domain Model for more
- 6600 information on this parameter.

6601

6602 POSTing an Atom Entry document with CMIS markup:

6603 Adding a document to a folder:

6604 If the atom entry has a cmis property cmis:objectId that is valid for the repository, the object will

6605 be added to the folder.

6606

6607 When an object is added to the folder, in repositories that do not support multi-filing it will be

6608 removed from the previous folder and the operation treated as move. If the repository supports

6609 multiple folders, it will be added to the new folder.

6610 If the optional argument sourceFolderId is specified, then the object will be removed from the

6611 folder specified.

6612

6613 If atom:content is missing from the request, the repository MUST treat the missing atom:content

6614 element as an empty atom:content element.

6615 Example client request:

```
6616 POST /obj/1cd0d82f-d579-4897-9b0a-ad0917595445?sourceFolderId=313fd58d-2eab-
6617 41af-9517-06dad010d49 HTTP/1.1
6618 Host: example.org
6619 Content-Length: 1227
6620 Content-Type: application/atom+xml;type=entry
6621
6622 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
6623 <atom:entry xmlns:app="http://www.w3.org/2007/app"
6624 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
6625 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
6626 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
6627 open.org/ns/cmis/restatom/200908/">
6628 <atom:author>
6629 <atom:name>Al Brown</atom:name>
6630 </atom:author>
6631 <atom:id>urn:uuid:1cd0d82f-d579-4897-9b0a-ad0917595445</atom:id>
6632 <atom:title type="text">Document - To Be Moved</atom:title>
6633 <atom:updated>2010-01-25T10:20:58.708-08:00</atom:updated>
6634 <atom:content src="http://cmisexample.oasis-
6635 open.org/rep1/content/1cd0d82f-d579-4897-9b0a-ad0917595445"/>
6636 <cmisra:object>
6637 <cmis:properties>
6638 <cmis:propertyId localName="rep-cmis:objectId"
6639 propertyDefinitionId="cmis:objectId">
6640 <cmis:value>1cd0d82f-d579-4897-9b0a-ad0917595445</cmis:value>
6641 </cmis:propertyId>
6642 <cmis:propertyId localName="rep-cmis:objectTypeId"
6643 propertyDefinitionId="cmis:objectTypeId">
6644 <cmis:value>invoice</cmis:value>
6645 </cmis:propertyId>
6646 </cmis:properties>
6647 </cmisra:object>
6648 </atom:entry>
6649
6650
```

6651

6652 Example server response:

6653 HTTP/1.1 201 Created  
6654 Date: Mon, 25 Jan 2010 10:20:58 -0800  
6655 Content-Length: 7213  
6656 Content-Type: application/atom+xml;type=entry  
6657 Content-Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-  
6658 8141-4aed91d28b5b  
6659 Location: http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-  
6660 4aed91d28b5b  
6661  
6662  
6663 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
6664 <atom:entry xmlns:app="http://www.w3.org/2007/app"  
6665 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
6666 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-  
6667 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-  
6668 open.org/ns/cmisis/restatom/200908/">  
6669 <atom:author>  
6670 <atom:name>Al Brown</atom:name>  
6671 <atom:uri>http://www.ibm.com/</atom:uri>  
6672 <atom:email>albertcbrown@us.ibm.com</atom:email>  
6673 </atom:author>  
6674 <atom:content src="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-  
6675 49fb-8141-4aed91d28b5b"/>  
6676 <atom:id>urn:uuid:b4423b8a-e46e-49fb-8141-4aed91d28b5b</atom:id>  
6677 <atom:title type="text">Document - To Be Moved</atom:title>  
6678 <atom:updated>2010-01-25T10:20:58.786-08:00</atom:updated>  
6679 <atom:link rel="self" href="http://cmisexample.oasis-  
6680 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b"/>  
6681 <atom:link rel="edit" href="http://cmisexample.oasis-  
6682 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b"/>  
6683 <atom:link type="application/cmisis+xml;type=allowableActions"  
6684 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"  
6685 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-  
6686 4aed91d28b5b/allowableactions"/>  
6687 <atom:link type="application/atom+xml;type=entry" rel="describedby"  
6688 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-  
6689 4aed91d28b5b/type"/>  
6690 <atom:link type="application/atomsvc+xml" rel="service"  
6691 href="http://cmisexample.oasis-open.org/rep1//service"/>  
6692 <atom:published>2010-01-25T10:20:58.786-08:00</atom:published>  
6693 <atom:summary type="html">HTML summary of Entry b4423b8a-e46e-49fb-8141-  
6694 4aed91d28b5b</atom:summary>  
6695 <atom:link rel="edit-media" href="http://cmisexample.oasis-  
6696 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/edit-media"/>  
6697 <atom:link rel="alternate" href="http://cmisexample.oasis-  
6698 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/alternate"/>  
6699 <atom:link type="application/atom+xml;type=feed" rel="up"  
6700 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-  
6701 4aed91d28b5b/parents"/>  
6702 <atom:link type="application/atom+xml;type=feed" rel="version-history"  
6703 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-  
6704 4aed91d28b5b/allversions"/>  
6705 <atom:link type="application/atom+xml;type=entry" rel="current-version"  
6706 href="http://cmisexample.oasis-open.org/rep1/b4423b8a-e46e-49fb-8141-  
6707 4aed91d28b5b/latest"/>  
6708 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-  
6709 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-  
6710 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/relationships"/>  
6711 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-  
6712 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-  
6713 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/policies"/>  
6714 <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-  
6715 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-  
6716 open.org/rep1/b4423b8a-e46e-49fb-8141-4aed91d28b5b/acl"/>

```

6717     <cmisra:object>
6718         <cmis:properties>
6719             <cmis:propertyId localName="rep-cmis:objectId"
6720 propertyDefinitionId="cmis:objectId">
6721                 <cmis:value>b4423b8a-e46e-49fb-8141-4aed91d28b5b</cmis:value>
6722             </cmis:propertyId>
6723             <cmis:propertyId localName="rep-cmis:objectTypeId"
6724 propertyDefinitionId="cmis:objectTypeId">
6725                 <cmis:value>invoice</cmis:value>
6726             </cmis:propertyId>
6727             <cmis:propertyString localName="rep-cmis:name"
6728 propertyDefinitionId="cmis:name">
6729                 <cmis:value>Document - To Be Moved</cmis:value>
6730             </cmis:propertyString>
6731             <cmis:propertyDateTime localName="rep-cmis:creationDate"
6732 propertyDefinitionId="cmis:creationDate">
6733                 <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
6734             </cmis:propertyDateTime>
6735             <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6736 propertyDefinitionId="cmis:lastModificationDate">
6737                 <cmis:value>2010-01-25T10:20:58.786-08:00</cmis:value>
6738             </cmis:propertyDateTime>
6739             <cmis:propertyId localName="rep-cmis:baseTypeId"
6740 propertyDefinitionId="cmis:baseTypeId">
6741                 <cmis:value>cmis:document</cmis:value>
6742             </cmis:propertyId>
6743             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6744 propertyDefinitionId="cmis:lastModifiedBy">
6745                 <cmis:value>Al Brown</cmis:value>
6746             </cmis:propertyString>
6747             <cmis:propertyString localName="rep-cmis:createdBy"
6748 propertyDefinitionId="cmis:createdBy">
6749                 <cmis:value>Al Brown</cmis:value>
6750             </cmis:propertyString>
6751             <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6752 propertyDefinitionId="cmis:isLatestVersion">
6753                 <cmis:value>>true</cmis:value>
6754             </cmis:propertyBoolean>
6755             <cmis:propertyBoolean localName="rep-
6756 cmis:isVersionSeriesCheckedOut"
6757 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6758                 <cmis:value>>false</cmis:value>
6759             </cmis:propertyBoolean>
6760             <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6761 propertyDefinitionId="cmis:isMajorVersion">
6762                 <cmis:value>>false</cmis:value>
6763             </cmis:propertyBoolean>
6764             <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6765 propertyDefinitionId="cmis:isLatestMajorVersion">
6766                 <cmis:value>>false</cmis:value>
6767             </cmis:propertyBoolean>
6768             <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6769 propertyDefinitionId="cmis:isImmutable">
6770                 <cmis:value>>false</cmis:value>
6771             </cmis:propertyBoolean>
6772             <cmis:propertyString localName="rep-cmis:checkinComment"
6773 propertyDefinitionId="cmis:checkinComment">
6774                 <cmis:value>Checkin comment</cmis:value>
6775             </cmis:propertyString>
6776             <cmis:propertyString localName="rep-cmis:versionLabel"
6777 propertyDefinitionId="cmis:versionLabel">
6778                 <cmis:value>0.1</cmis:value>
6779             </cmis:propertyString>

```

```

6780         <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6781 propertyDefinitionId="cmis:contentStreamMimeType">
6782         <cmis:value>text/plain</cmis:value>
6783     </cmis:propertyString>
6784     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
6785 propertyDefinitionId="cmis:contentStreamFileName">
6786     <cmis:value>text.txt</cmis:value>
6787     </cmis:propertyString>
6788     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
6789 propertyDefinitionId="cmis:contentStreamLength">
6790     <cmis:value>4234</cmis:value>
6791     </cmis:propertyInteger>
6792     <cmis:propertyString displayName="Keywords for Document"
6793 localName="keywords" propertyDefinitionId="keywords">
6794     <cmis:value>document</cmis:value>
6795     <cmis:value>example</cmis:value>
6796     <cmis:value>sample</cmis:value>
6797     <cmis:value>cmis</cmis:value>
6798     </cmis:propertyString>
6799 </cmis:properties>
6800 </cmisra:object>
6801 </atom:entry>
6802

```

6803

6804 Please also see the example documents included with the schema.

6805

6806 Creating a CMIS Object (in that folder):

6807 If the cmis:objectId property is missing, the object will be created and then added to the folder. If  
6808 the cmis:objectId property is present but not a valid object Id, the repository MUST return the  
6809 appropriate HTTP status code.

6810

6811 For Documents:

6812 If Content Stream is not provided and it is required by the type definition, the repository  
6813 MUST return the appropriate HTTP status code.

6814

6815 Content Streams MAY be provided by any of the following mechanisms:

- 6816 ○ As part of the atom entry via the src attribute on the content element (AtomPub)
  - 6817 ■ src attribute: Implementers MAY support external references to content
  - 6818 ■ If the URI in the src attribute is not reachable, then an appropriate http
  - 6819 status code should be returned.
- 6820 ○ As part of the atom entry inlining via the content element (AtomPub)
  - 6821 ■ Please see the AtomPub specification RFC5023 for the processing
  - 6822 model of the content element.
- 6823 ○ If the cmisra:content is provided by the client inside the atom:entry, the  
6824 cmisra:content element MUST take precedence over the atom:content element.  
6825 (CMIS)
  - 6826 ● This element cmisra:content is base64 encoded
- 6827 ○ At a later time (AtomPub)
  - 6828 ■ At a later time by replacing the edit-media link with a new content

6829

6830 The optional argument versioningState MAY specify additional versioning behavior such  
6831 as checkin.

6832

6833

**Example client request:**

6834

```
POST /obj/bb2b208b-3acd-4abe-9788-8078a239f228 HTTP/1.1
```

6835

```
Host: example.org
```

6836

```
Content-Length: 1190
```

6837

```
Content-Type: application/atom+xml;type=entry
```

6838

6839

6840

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

6841

```
<atom:entry xmlns:app="http://www.w3.org/2007/app"
```

6842

```
xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
```

6843

```
open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
```

6844

```
open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
```

6845

```
open.org/ns/cmisis/restatom/200908/">
```

6846

```
<atom:author>
```

6847

```
<atom:name>Al Brown</atom:name>
```

6848

```
</atom:author>
```

6849

```
<atom:id>urn:uuid:bb2b208b-3acd-4abe-9788-8078a239f228</atom:id>
```

6850

```
<atom:title type="text">New Invoice</atom:title>
```

6851

```
<atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
```

6852

```
<atom:content type="text">this is the content of the new
```

6853

```
document</atom:content>
```

6854

```
<cmisra:object>
```

6855

```
<cmis:properties>
```

6856

```
<cmis:propertyId localName="rep-cmis:objectId"
```

6857

```
propertyDefinitionId="cmis:objectId">
```

6858

```
<cmis:value>bb2b208b-3acd-4abe-9788-8078a239f228</cmis:value>
```

6859

```
</cmis:propertyId>
```

6860

```
<cmis:propertyId localName="rep-cmis:objectTypeId"
```

6861

```
propertyDefinitionId="cmis:objectTypeId">
```

6862

```
<cmis:value>invoice</cmis:value>
```

6863

```
</cmis:propertyId>
```

6864

```
</cmis:properties>
```

6865

```
</cmisra:object>
```

6866

```
</atom:entry>
```

6867

6868

6869

**Example server response:**

6870

```
HTTP/1.1 201 Created
```

6871

```
Date: Mon, 25 Jan 2010 10:20:58 -0800
```

6872

```
Content-Length: 7191
```

6873

```
Content-Type: application/atom+xml;type=entry
```

6874

```
Content-Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-
```

6875

```
ad0b-10ea94c4b93d
```

6876

```
Location: http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
```

6877

```
10ea94c4b93d
```

6878

6879

6880

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

6881

```
<atom:entry xmlns:app="http://www.w3.org/2007/app"
```

6882

```
xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
```

6883

```
open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
```

6884

```
open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
```

6885

```
open.org/ns/cmisis/restatom/200908/">
```

6886

```
<atom:author>
```

6887

```
<atom:name>Al Brown</atom:name>
```

6888

```
<atom:uri>http://www.ibm.com/</atom:uri>
```

6889

```
<atom:email>albertcbrown@us.ibm.com</atom:email>
```

6890

```
</atom:author>
```

6891

```
<atom:content src="http://cmisexample.oasis-open.org/rep1/13475008-6a20-
```

6892

```
4454-ad0b-10ea94c4b93d"/>
```

6893

```
<atom:id>urn:uuid:13475008-6a20-4454-ad0b-10ea94c4b93d</atom:id>
```

```

6894     <atom:title type="text">New Invoice</atom:title>
6895     <atom:updated>2010-01-25T10:20:58.818-08:00</atom:updated>
6896     <atom:link rel="self" href="http://cmisexample.oasis-
6897 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>
6898     <atom:link rel="edit" href="http://cmisexample.oasis-
6899 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d"/>
6900     <atom:link type="application/cmism+xml;type=allowableActions"
6901 rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
6902 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6903 10ea94c4b93d/allowableactions"/>
6904     <atom:link type="application/atom+xml;type=entry" rel="describedby"
6905 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6906 10ea94c4b93d/type"/>
6907     <atom:link type="application/atomsvc+xml" rel="service"
6908 href="http://cmisexample.oasis-open.org/rep1//service"/>
6909     <atom:published>2010-01-25T10:20:58.833-08:00</atom:published>
6910     <atom:summary type="html">HTML summary of Entry 13475008-6a20-4454-ad0b-
6911 10ea94c4b93d</atom:summary>
6912     <atom:link rel="edit-media" href="http://cmisexample.oasis-
6913 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/edit-media"/>
6914     <atom:link rel="alternate" href="http://cmisexample.oasis-
6915 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/alternate"/>
6916     <atom:link type="application/atom+xml;type=feed" rel="up"
6917 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6918 10ea94c4b93d/parents"/>
6919     <atom:link type="application/atom+xml;type=feed" rel="version-history"
6920 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6921 10ea94c4b93d/allversions"/>
6922     <atom:link type="application/atom+xml;type=entry" rel="current-version"
6923 href="http://cmisexample.oasis-open.org/rep1/13475008-6a20-4454-ad0b-
6924 10ea94c4b93d/latest"/>
6925     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6926 open.org/ns/cmism/link/200908/relationships" href="http://cmisexample.oasis-
6927 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/relationships"/>
6928     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
6929 open.org/ns/cmism/link/200908/policies" href="http://cmisexample.oasis-
6930 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/policies"/>
6931     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
6932 open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
6933 open.org/rep1/13475008-6a20-4454-ad0b-10ea94c4b93d/acl"/>
6934     <cmisra:object>
6935       <cmis:properties>
6936         <cmis:propertyId localName="rep-cmis:objectId"
6937 propertyDefinitionId="cmis:objectId">
6938           <cmis:value>13475008-6a20-4454-ad0b-10ea94c4b93d</cmis:value>
6939         </cmis:propertyId>
6940         <cmis:propertyId localName="rep-cmis:objectTypeId"
6941 propertyDefinitionId="cmis:objectTypeId">
6942           <cmis:value>invoice</cmis:value>
6943         </cmis:propertyId>
6944         <cmis:propertyString localName="rep-cmis:name"
6945 propertyDefinitionId="cmis:name">
6946           <cmis:value>New Invoice</cmis:value>
6947         </cmis:propertyString>
6948         <cmis:propertyDateTime localName="rep-cmis:creationDate"
6949 propertyDefinitionId="cmis:creationDate">
6950           <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
6951         </cmis:propertyDateTime>
6952         <cmis:propertyDateTime localName="rep-cmis:lastModificationDate"
6953 propertyDefinitionId="cmis:lastModificationDate">
6954           <cmis:value>2010-01-25T10:20:58.833-08:00</cmis:value>
6955         </cmis:propertyDateTime>
6956         <cmis:propertyId localName="rep-cmis:baseTypeId"
6957 propertyDefinitionId="cmis:baseTypeId">

```

```

6958         <cmis:value>cmis:document</cmis:value>
6959     </cmis:propertyId>
6960     <cmis:propertyString localName="rep-cmis:lastModifiedBy"
6961 propertyDefinitionId="cmis:lastModifiedBy">
6962         <cmis:value>Al Brown</cmis:value>
6963     </cmis:propertyString>
6964     <cmis:propertyString localName="rep-cmis:createdBy"
6965 propertyDefinitionId="cmis:createdBy">
6966         <cmis:value>Al Brown</cmis:value>
6967     </cmis:propertyString>
6968     <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
6969 propertyDefinitionId="cmis:isLatestVersion">
6970         <cmis:value>>true</cmis:value>
6971     </cmis:propertyBoolean>
6972     <cmis:propertyBoolean localName="rep-
6973 cmis:isVersionSeriesCheckedOut"
6974 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
6975         <cmis:value>>false</cmis:value>
6976     </cmis:propertyBoolean>
6977     <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
6978 propertyDefinitionId="cmis:isMajorVersion">
6979         <cmis:value>>false</cmis:value>
6980     </cmis:propertyBoolean>
6981     <cmis:propertyBoolean localName="rep-cmis:isLatestMajorVersion"
6982 propertyDefinitionId="cmis:isLatestMajorVersion">
6983         <cmis:value>>false</cmis:value>
6984     </cmis:propertyBoolean>
6985     <cmis:propertyBoolean localName="rep-cmis:isImmutable"
6986 propertyDefinitionId="cmis:isImmutable">
6987         <cmis:value>>false</cmis:value>
6988     </cmis:propertyBoolean>
6989     <cmis:propertyString localName="rep-cmis:checkinComment"
6990 propertyDefinitionId="cmis:checkinComment">
6991         <cmis:value>Checkin comment</cmis:value>
6992     </cmis:propertyString>
6993     <cmis:propertyString localName="rep-cmis:versionLabel"
6994 propertyDefinitionId="cmis:versionLabel">
6995         <cmis:value>0.1</cmis:value>
6996     </cmis:propertyString>
6997     <cmis:propertyString localName="rep-cmis:contentStreamMimeType"
6998 propertyDefinitionId="cmis:contentStreamMimeType">
6999         <cmis:value>text/plain</cmis:value>
7000     </cmis:propertyString>
7001     <cmis:propertyString localName="rep-cmis:contentStreamFileName"
7002 propertyDefinitionId="cmis:contentStreamFileName">
7003         <cmis:value>text.txt</cmis:value>
7004     </cmis:propertyString>
7005     <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7006 propertyDefinitionId="cmis:contentStreamLength">
7007         <cmis:value>4234</cmis:value>
7008     </cmis:propertyInteger>
7009     <cmis:propertyString displayName="Keywords for Document"
7010 localName="keywords" propertyDefinitionId="keywords">
7011         <cmis:value>document</cmis:value>
7012         <cmis:value>example</cmis:value>
7013         <cmis:value>sample</cmis:value>
7014         <cmis:value>cmis</cmis:value>
7015     </cmis:propertyString>
7016 </cmis:properties>
7017 </cmisra:object>
7018 </atom:entry>
7019

```

7020

7021 Please also see the example documents included with the schema.

7022

7023 POSTing other document formats: (AtomPub)

7024 The behavior is repository specific when a non Atom entry or an atom document without the  
7025 CMIS elements is posted to a folder collection.

7026 For example, the repository MAY auto-create a document with a specific type (document) the  
7027 client could edit.

7028 If the repository does not support this scenario or another exception occurs, then the repository  
7029 MUST return the appropriate HTTP status code.

7030

7031 Optional arguments:

7032 | • versioningState (for createDocument)

7033 | • sourceFolderId (for moveObject)

7034

### 7035 3.8.3 Policies Collection

7036 This is an atom feed of all the policy objects currently applied to a specific object. This is the only  
7037 | collection where the URIs of the objects in the collection MUST be specific to that collection. A DELETE  
7038 on the policy object in the collection is a removal of the policy from the object NOT a deletion of the policy  
7039 object itself.

7040

7041 CMIS Services:

7042 GET: getAppliedPolicies

7043 POST: applyPolicy (to object representing this collection of policies)

7044 DELETE: removePolicy

7045 Media Type: application/atom+xml;type=feed

7046 Accept:

7047 | • MUST support Atom Entry Documents with CMIS extensions

7048 | ○ application/atom+xml;type=entry or

7049 | ○ application/cmisaatom+xml

7050 | • MAY support other media type

7051

7052 Link Relations:

7053 | • service: Points to service document containing the CMIS repository. The service document  
7054 MUST contain only one workspace element.

7055 | ○ Media Type: application/atomsvc+xml

7056 | • via: points to the atom entry of the resource generating this collection

7057 | • paging link relations as appropriate: first, next, previous, last

7058

7059 The policy entries displayed here are specific to the object generating this collection. A DELETE method  
7060 on those URIs will invoke removePolicy().

7061

7062 The following CMIS Atom extension element MAY be included inside the atom feed:

7063 | • cmisra:numItems

7064

7065 The following CMIS Atom extension element **MUST** be included inside the atom entries:

- 7066 • cmisra:object inside atom:entry

7067

### 7068 3.8.3.1 GET

7069 The following arguments may be supplied. Please see the domain model for more information:

- 7070 • filter

### 7071 3.8.3.2 POST

7072 When an Atom Entry representing a Policy is posted to this collection, the policy will be applied to the  
7073 object.

7074

7075 Example client request:

```
7076 POST /policies/f3670f66-62ee-487f-b733-999a69237024 HTTP/1.1
7077 Host: example.org
7078 Content-Length: 1039
7079 Content-Type: application/atom+xml;type=entry
7080
7081 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7082 <atom:entry xmlns:app="http://www.w3.org/2007/app"
7083 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
7084 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
7085 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
7086 open.org/ns/cmisis/restatom/200908/">
7087   <atom:author>
7088     <atom:name>Al Brown</atom:name>
7089   </atom:author>
7090   <atom:content src="http://cmisexample.oasis-open.org/rep1/f3670f66-62ee-
7091 487f-b733-999a69237024"/>
7092   <atom:id>urn:uuid:f3670f66-62ee-487f-b733-999a69237024</atom:id>
7093   <atom:title type="text">Security Policy for Invoices</atom:title>
7094   <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
7095   <cmisra:object>
7096     <cmis:properties>
7097       <cmis:propertyId localName="rep-cmis:objectId"
7098 propertyDefinitionId="cmis:objectId">
7099         <cmis:value>f3670f66-62ee-487f-b733-999a69237024</cmis:value>
7100       </cmis:propertyId>
7101     </cmis:properties>
7102   </cmisra:object>
7103 </atom:entry>
7104
7105
```

7106

7107 Example server response:

```
7108 HTTP/1.1 201 Created
7109 Date: Mon, 25 Jan 2010 10:20:58 -0800
7110 Content-Length: 4043
7111 Content-Type: application/atom+xml;type=entry
7112 Content-Location: http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-
7113 bdf6-690761576116
7114 Location: http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-bdf6-
7115 690761576116
7116
7117 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7118
```

```

7119 <atom:entry xmlns:app="http://www.w3.org/2007/app"
7120 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
7121 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
7122 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
7123 open.org/ns/cmisis/restatom/200908/">
7124   <atom:author>
7125     <atom:name>Al Brown</atom:name>
7126     <atom:uri>http://www.ibm.com/</atom:uri>
7127     <atom:email>albertcbrown@us.ibm.com</atom:email>
7128   </atom:author>
7129   <atom:content src="http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-
7130 4354-bdfe-690761576116"/>
7131   <atom:id>urn:uuid:55cca51b-6cfa-4354-bdfe-690761576116</atom:id>
7132   <atom:title type="text">Security Policy for Invoices</atom:title>
7133   <atom:updated>2010-01-25T10:20:58.849-08:00</atom:updated>
7134   <atom:link rel="self" href="http://cmisexample.oasis-
7135 open.org/rep1/55cca51b-6cfa-4354-bdfe-690761576116"/>
7136   <atom:link rel="edit" href="http://cmisexample.oasis-
7137 open.org/rep1/55cca51b-6cfa-4354-bdfe-690761576116"/>
7138   <atom:link type="application/cmisis+xml;type=allowableActions"
7139 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7140 href="http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-bdfe-
7141 690761576116/allowableactions"/>
7142   <atom:link type="application/atom+xml;type=entry" rel="describedby"
7143 href="http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-bdfe-
7144 690761576116/type"/>
7145   <atom:link type="application/atomsvc+xml" rel="service"
7146 href="http://cmisexample.oasis-open.org/rep1//service"/>
7147   <atom:published>2010-01-25T10:20:58.849-08:00</atom:published>
7148   <atom:summary type="html">HTML summary of Entry 55cca51b-6cfa-4354-bdfe-
7149 690761576116</atom:summary>
7150   <atom:link type="application/atom+xml;type=feed" rel="up"
7151 href="http://cmisexample.oasis-open.org/rep1/55cca51b-6cfa-4354-bdfe-
7152 690761576116/parents"/>
7153   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7154 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
7155 open.org/rep1/55cca51b-6cfa-4354-bdfe-690761576116/relationships"/>
7156   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
7157 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
7158 open.org/rep1/55cca51b-6cfa-4354-bdfe-690761576116/acl"/>
7159   <cmisra:object>
7160     <cmis:properties>
7161       <cmis:propertyId localName="rep-cmisis:objectId"
7162 propertyDefinitionId="cmisis:objectId">
7163         <cmis:value>55cca51b-6cfa-4354-bdfe-690761576116</cmis:value>
7164       </cmis:propertyId>
7165       <cmis:propertyId localName="rep-cmisis:objectTypeId"
7166 propertyDefinitionId="cmisis:objectTypeId">
7167         <cmis:value>generalSecurityPolicy</cmis:value>
7168       </cmis:propertyId>
7169       <cmis:propertyString localName="rep-cmisis:name"
7170 propertyDefinitionId="cmisis:name">
7171         <cmis:value>Security Policy for Invoices</cmis:value>
7172       </cmis:propertyString>
7173       <cmis:propertyDateTime localName="rep-cmisis:creationDate"
7174 propertyDefinitionId="cmisis:creationDate">
7175         <cmis:value>2010-01-25T10:20:58.849-08:00</cmis:value>
7176       </cmis:propertyDateTime>
7177       <cmis:propertyDateTime localName="rep-cmisis:lastModificationDate"
7178 propertyDefinitionId="cmisis:lastModificationDate">
7179         <cmis:value>2010-01-25T10:20:58.864-08:00</cmis:value>
7180       </cmis:propertyDateTime>
7181       <cmis:propertyId localName="rep-cmisis:baseTypeId"
7182 propertyDefinitionId="cmisis:baseTypeId">

```

```

7183         <cmis:value>cmis:policy</cmis:value>
7184     </cmis:propertyId>
7185     <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7186 propertyDefinitionId="cmis:lastModifiedBy">
7187         <cmis:value>Al Brown</cmis:value>
7188     </cmis:propertyString>
7189     <cmis:propertyString localName="rep-cmis:createdBy"
7190 propertyDefinitionId="cmis:createdBy">
7191         <cmis:value>Al Brown</cmis:value>
7192     </cmis:propertyString>
7193 </cmis:properties>
7194 </cmisra:object>
7195 </atom:entry>
7196

```

7197

7198 Please also see the example documents included with the schema.

### 7199 **3.8.3.3 DELETE**

7200 | This is the only collection where the URI's of the objects in the collection MUST be specific to that  
7201 | collection. A DELETE on the policy object in the collection is a removal of the policy from the object NOT  
7202 | a deletion of the policy object itself.

7203

## 7204 **3.9 Feeds**

7205 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP  
7206 verb in a repository-specific manner.

### 7207 **3.9.1 Object Parents Feed**

7208 This is the set of parents for a specific object.

7209 CMIS Services:

7210 GET: getObjectParents

7211 Media Type: application/atom+xml;type=feed

7212

7213 Link Relations:

- 7214 | • service: Points to service document containing the CMIS repository. The service document  
7215 | MUST contain only one workspace element.
  - 7216 | ○ Media Type: application/atomsvc+xml
- 7217 | • via: points to the atom entry of object who's parents are represented by this collection

7218

7219 This feed contains a set of atom entries for each parent of the object that MUST contain:

- 7220 | • cmisra:object inside atom:entry
- 7221 | • cmisra:relativePathSegment inside atom:entry for the name of the object inside the folder

7222

7223 Example:

```

7224 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7225 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7226 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7227 xmlns:atom="http://www.w3.org/2005/Atom"
7228 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7229 open.org/ns/cmisis/restatom/200908/">

```

```

7230     <atom:title type="text">Parent Feed for 268d30b5-91a0-47f0-b985-
7231 6765e178f0bb</atom:title>
7232     <atom:author>
7233       <atom:name>Al Brown</atom:name>
7234       <atom:uri>http://www.ibm.com/</atom:uri>
7235       <atom:email>albertcbrown@us.ibm.com</atom:email>
7236     </atom:author>
7237     <atom:updated>2010-01-25T10:20:59.818-08:00</atom:updated>
7238     <atom:id>urn:uuid:6f541940-4abf-471b-99f0-8e6f66d53789</atom:id>
7239     <atom:link type="application/atom+xml;type=feed" rel="self"
7240 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7241 6765e178f0bb/3"/>
7242     <atom:link type="application/atomsvc+xml" rel="service"
7243 href="http://cmisexample.oasis-open.org/rep1//service"/>
7244     <atom:link type="application/atom+xml;type=entry" rel="via"
7245 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7246 6765e178f0bb"/>
7247     <atom:link type="application/atom+xml;type=feed" rel="first"
7248 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7249 6765e178f0bb/first"/>
7250     <atom:link type="application/atom+xml;type=feed" rel="next"
7251 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7252 6765e178f0bb/4"/>
7253     <atom:link type="application/atom+xml;type=feed" rel="previous"
7254 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7255 6765e178f0bb/2"/>
7256     <atom:link type="application/atom+xml;type=feed" rel="last"
7257 href="http://cmisexample.oasis-open.org/rep1/268d30b5-91a0-47f0-b985-
7258 6765e178f0bb/last"/>
7259     <cmisra:numItems>1</cmisra:numItems>
7260     <atom:entry>
7261       <atom:author>
7262         <atom:name>Al Brown</atom:name>
7263         <atom:uri>http://www.ibm.com/</atom:uri>
7264         <atom:email>albertcbrown@us.ibm.com</atom:email>
7265       </atom:author>
7266       <atom:content src="http://cmisexample.oasis-open.org/rep1/661d6945-
7267 8f75-4dea-8799-7ba07b0e510e"/>
7268       <atom:id>urn:uuid:661d6945-8f75-4dea-8799-7ba07b0e510e</atom:id>
7269       <atom:title type="text">Customer Folder</atom:title>
7270       <atom:updated>2010-01-25T10:20:59.833-08:00</atom:updated>
7271       <atom:link rel="self" href="http://cmisexample.oasis-
7272 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
7273       <atom:link rel="edit" href="http://cmisexample.oasis-
7274 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e"/>
7275       <atom:link type="application/cmisis+xml;type=allowableActions"
7276 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7277 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7278 7ba07b0e510e/allowableactions"/>
7279       <atom:link type="application/atom+xml;type=entry" rel="describedby"
7280 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7281 7ba07b0e510e/type"/>
7282       <atom:link type="application/atomsvc+xml" rel="service"
7283 href="http://cmisexample.oasis-open.org/rep1//service"/>
7284       <atom:published>2010-01-25T10:20:59.833-08:00</atom:published>
7285       <atom:summary type="html">HTML summary of Entry 661d6945-8f75-4dea-
7286 8799-7ba07b0e510e</atom:summary>
7287       <atom:link type="application/atom+xml;type=entry" rel="up"
7288 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7289 7ba07b0e510e/up"/>
7290       <atom:link type="application/atom+xml;type=feed" rel="down"
7291 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7292 7ba07b0e510e/children"/>

```

```

7293     <atom:link type="application/cmistree+xml" rel="down"
7294 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7295 7ba07b0e510e/tree"/>
7296     <atom:link type="application/atom+xml;type=feed"
7297 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7298 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7299 7ba07b0e510e/foldertree"/>
7300     <atom:link type="application/atom+xml;type=feed"
7301 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
7302 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7303 7ba07b0e510e/relationships"/>
7304     <atom:link type="application/atom+xml;type=feed"
7305 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
7306 href="http://cmisexample.oasis-open.org/rep1/661d6945-8f75-4dea-8799-
7307 7ba07b0e510e/policies"/>
7308     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7309 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
7310 open.org/rep1/661d6945-8f75-4dea-8799-7ba07b0e510e/acl"/>
7311     <cmisra:object>
7312         <cmis:properties>
7313             <cmis:propertyId localName="rep-cmis:objectId"
7314 propertyDefinitionId="cmis:objectId">
7315                 <cmis:value>661d6945-8f75-4dea-8799-
7316 7ba07b0e510e</cmis:value>
7317             </cmis:propertyId>
7318             <cmis:propertyId localName="rep-cmis:objectTypeId"
7319 propertyDefinitionId="cmis:objectTypeId">
7320                 <cmis:value>customer</cmis:value>
7321             </cmis:propertyId>
7322             <cmis:propertyString localName="rep-cmis:name"
7323 propertyDefinitionId="cmis:name">
7324                 <cmis:value>Customer Folder</cmis:value>
7325             </cmis:propertyString>
7326             <cmis:propertyDateTime localName="rep-cmis:creationDate"
7327 propertyDefinitionId="cmis:creationDate">
7328                 <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
7329             </cmis:propertyDateTime>
7330             <cmis:propertyDateTime localName="rep-
7331 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7332                 <cmis:value>2010-01-25T10:20:59.833-08:00</cmis:value>
7333             </cmis:propertyDateTime>
7334             <cmis:propertyId localName="rep-cmis:baseTypeId"
7335 propertyDefinitionId="cmis:baseTypeId">
7336                 <cmis:value>cmis:folder</cmis:value>
7337             </cmis:propertyId>
7338             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7339 propertyDefinitionId="cmis:lastModifiedBy">
7340                 <cmis:value>Al Brown</cmis:value>
7341             </cmis:propertyString>
7342             <cmis:propertyString localName="rep-cmis:createdBy"
7343 propertyDefinitionId="cmis:createdBy">
7344                 <cmis:value>Al Brown</cmis:value>
7345             </cmis:propertyString>
7346             <cmis:propertyId localName="rep-cmis:parentId"
7347 propertyDefinitionId="cmis:parentId">
7348                 <cmis:value>661d6945-8f75-4dea-8799-
7349 7ba07b0e510eup</cmis:value>
7350             </cmis:propertyId>
7351         </cmis:properties>
7352     </cmisra:object>
7353     <cmisra:relativePathSegment>customer1</cmisra:relativePathSegment>
7354 </atom:entry>
7355 </atom:feed>

```

7356

7357 Please also see the example documents included with the schema.

### 7358 **3.9.1.1 GET**

7359 The following arguments may be supplied. Please see the domain model for more information:

- 7360 • filter
- 7361 • includeAllowableActions
- 7362 • includeRelationships
- 7363 • renditionFilter
- 7364 • includeRelativePathSegment
- 7365 ○ If true, then the cmisra:relativePathSegment element MUST be included in the response.

### 7366 **3.9.2 Changes**

7367 This is a link relationship described in the service document that contains the changes in the repository in  
7368 the workspace element. The link relation pointing to this feed is ~~[http://docs.oasis-](http://docs.oasis-<br/>7369 open.org/ns/cmisis/link/200908/changes)~~  
~~[open.org/ns/cmisis/link/200908/changes](http://docs.oasis-open.org/ns/cmisis/link/200908/changes)~~  
~~<http://docs.oasis-open.org/ns/cmisis/link/200908/changes>~~.

7370  
7371 *The ChangeLog Token is specified in the URI specified by the paging link notations. Through this binding  
7372 it is not possible to retrieve the ChangeLog Token from the URIs.*

7373

7374 CMIS Services:

7375 GET: getContentChanges()

7376 Media Type: application/atom+xml;type=feed

7377 Link Relations:

- 7378 • service: Points to service document containing the CMIS repository. The service document  
7379 MUST contain only one workspace element.
  - 7380 ○ Media Type: application/atomsvc+xml
- 7381 • paging link relations as appropriate: first, next, previous, last
  - 7382 ○ ChangeLogToken is incorporated into the URI specified by the next link relation

7383

7384 This feed MUST be ordered from oldest first to newest.

7385

7386 If the next changes does not exist yet, the link relation next MAY be available. If the next link relation is  
7387 not available, the client should revisit the feed in the future and look for new items and the next link  
7388 relation.

7389

7390 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7391 • cmisra:numItems

7392

7393 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7394 • cmisra:object inside atom:entry

7395

7396 Example:

7397 

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

```

7398 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
7399 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
7400 xmlns:atom="http://www.w3.org/2005/Atom"
7401 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7402 open.org/ns/cmisis/restatom/200908/">
7403   <atom:title type="text">changelog feed</atom:title>
7404   <atom:author>
7405     <atom:name>Al Brown</atom:name>
7406     <atom:uri>http://www.ibm.com/</atom:uri>
7407     <atom:email>albertcbrown@us.ibm.com</atom:email>
7408   </atom:author>
7409   <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7410   <atom:id>urn:uuid:0bfc5306-fc76-4cd8-a0c0-7653dd43c0ff</atom:id>
7411   <atom:link type="application/atom+xml;type=feed" rel="self"
7412 href="http://cmisexample.oasis-open.org/repl/oId/3"/>
7413   <atom:link type="application/atomsvc+xml" rel="service"
7414 href="http://cmisexample.oasis-open.org/repl//service"/>
7415   <atom:link type="application/atom+xml;type=feed" rel="first"
7416 href="http://cmisexample.oasis-open.org/repl/oId/first"/>
7417   <atom:link type="application/atom+xml;type=feed" rel="next"
7418 href="http://cmisexample.oasis-open.org/repl/oId/4"/>
7419   <atom:link type="application/atom+xml;type=feed" rel="previous"
7420 href="http://cmisexample.oasis-open.org/repl/oId/2"/>
7421   <atom:link type="application/atom+xml;type=feed" rel="last"
7422 href="http://cmisexample.oasis-open.org/repl/oId/last"/>
7423   <cmisra:numItems>2</cmisra:numItems>
7424   <atom:entry>
7425     <atom:author>
7426       <atom:name>Al Brown</atom:name>
7427       <atom:uri>http://www.ibm.com/</atom:uri>
7428       <atom:email>albertcbrown@us.ibm.com</atom:email>
7429     </atom:author>
7430     <atom:content src="http://cmisexample.oasis-open.org/repl/3f724c1d-
7431 12c8-43f2-919f-674df52b6ebd"/>
7432     <atom:id>urn:uuid:3f724c1d-12c8-43f2-919f-674df52b6ebd</atom:id>
7433     <atom:title type="text">CMIS Example Folder as Customer Policy
7434 type</atom:title>
7435     <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7436     <atom:link rel="self" href="http://cmisexample.oasis-
7437 open.org/repl/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
7438     <atom:link rel="edit" href="http://cmisexample.oasis-
7439 open.org/repl/3f724c1d-12c8-43f2-919f-674df52b6ebd"/>
7440     <atom:link type="application/cmisis+xml;type=allowableActions"
7441 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
7442 href="http://cmisexample.oasis-open.org/repl/3f724c1d-12c8-43f2-919f-
7443 674df52b6ebd/allowableactions"/>
7444     <atom:link type="application/atom+xml;type=entry" rel="describedby"
7445 href="http://cmisexample.oasis-open.org/repl/3f724c1d-12c8-43f2-919f-
7446 674df52b6ebd/type"/>
7447     <atom:link type="application/atomsvc+xml" rel="service"
7448 href="http://cmisexample.oasis-open.org/repl//service"/>
7449     <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
7450     <atom:summary type="html">HTML summary of Entry 3f724c1d-12c8-43f2-
7451 919f-674df52b6ebd</atom:summary>
7452     <atom:link type="application/atom+xml;type=entry" rel="up"
7453 href="http://cmisexample.oasis-open.org/repl/3f724c1d-12c8-43f2-919f-
7454 674df52b6ebd/up"/>
7455     <atom:link type="application/atom+xml;type=feed" rel="down"
7456 href="http://cmisexample.oasis-open.org/repl/3f724c1d-12c8-43f2-919f-
7457 674df52b6ebd/children"/>
7458     <atom:link type="application/cmistree+xml" rel="down"
7459 href="http://cmisexample.oasis-open.org/repl/3f724c1d-12c8-43f2-919f-
7460 674df52b6ebd/tree"/>

```

```

7461     <atom:link type="application/atom+xml;type=feed"
7462 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7463 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7464 674df52b6ebd/foldertree"/>
7465     <atom:link type="application/atom+xml;type=feed"
7466 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/relationships"
7467 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7468 674df52b6ebd/relationships"/>
7469     <atom:link type="application/atom+xml;type=feed"
7470 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/policies"
7471 href="http://cmisexample.oasis-open.org/rep1/3f724c1d-12c8-43f2-919f-
7472 674df52b6ebd/policies"/>
7473     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7474 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
7475 open.org/rep1/3f724c1d-12c8-43f2-919f-674df52b6ebd/acl"/>
7476     <cmisra:object>
7477         <cmis:properties>
7478             <cmis:propertyId localName="rep-cmis:objectId"
7479 propertyDefinitionId="cmis:objectId">
7480                 <cmis:value>3f724c1d-12c8-43f2-919f-
7481 674df52b6ebd</cmis:value>
7482             </cmis:propertyId>
7483             <cmis:propertyId localName="rep-cmis:objectTypeId"
7484 propertyDefinitionId="cmis:objectTypeId">
7485                 <cmis:value>customerpolicy</cmis:value>
7486             </cmis:propertyId>
7487             <cmis:propertyString localName="rep-cmis:name"
7488 propertyDefinitionId="cmis:name">
7489                 <cmis:value>CMIS Example Folder as Customer Policy
7490 type</cmis:value>
7491             </cmis:propertyString>
7492             <cmis:propertyDateTime localName="rep-cmis:creationDate"
7493 propertyDefinitionId="cmis:creationDate">
7494                 <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
7495             </cmis:propertyDateTime>
7496             <cmis:propertyDateTime localName="rep-
7497 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7498                 <cmis:value>2010-01-25T10:20:59.255-08:00</cmis:value>
7499             </cmis:propertyDateTime>
7500             <cmis:propertyId localName="rep-cmis:baseTypeId"
7501 propertyDefinitionId="cmis:baseTypeId">
7502                 <cmis:value>cmis:folder</cmis:value>
7503             </cmis:propertyId>
7504             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7505 propertyDefinitionId="cmis:lastModifiedBy">
7506                 <cmis:value>Al Brown</cmis:value>
7507             </cmis:propertyString>
7508             <cmis:propertyString localName="rep-cmis:createdBy"
7509 propertyDefinitionId="cmis:createdBy">
7510                 <cmis:value>Al Brown</cmis:value>
7511             </cmis:propertyString>
7512             <cmis:propertyId localName="rep-cmis:parentId"
7513 propertyDefinitionId="cmis:parentId">
7514                 <cmis:value>3f724c1d-12c8-43f2-919f-
7515 674df52b6ebdup</cmis:value>
7516             </cmis:propertyId>
7517         </cmis:properties>
7518         <cmis:changeEventInfo>
7519             <cmis:changeType>updated</cmis:changeType>
7520             <cmis:changeTime>2010-01-25T10:20:59.255-
7521 08:00</cmis:changeTime>
7522         </cmis:changeEventInfo>
7523     </cmisra:object>
7524     <cmisra:pathSegment>policy</cmisra:pathSegment>

```

```

7525     </atom:entry>
7526     <atom:entry>
7527         <atom:author>
7528             <atom:name>Al Brown</atom:name>
7529             <atom:uri>http://www.ibm.com/</atom:uri>
7530             <atom:email>albertcbrown@us.ibm.com</atom:email>
7531         </atom:author>
7532         <atom:content src="http://cmisexample.oasis-open.org/rep1/6e27bada-
7533 b5a2-4a39-be2c-269806eb0d42"/>
7534         <atom:id>urn:uuid:6e27bada-b5a2-4a39-be2c-269806eb0d42</atom:id>
7535         <atom:title type="text">CMIS Example Document</atom:title>
7536         <atom:updated>2010-01-25T10:20:59.255-08:00</atom:updated>
7537         <atom:link rel="self" href="http://cmisexample.oasis-
7538 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"/>
7539         <atom:link rel="edit" href="http://cmisexample.oasis-
7540 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42"/>
7541         <atom:link type="application/cmisacl+xml;type=allowableActions"
7542 rel="http://docs.oasis-open.org/ns/cmisacl/200908/allowableactions"
7543 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7544 269806eb0d42/allowableactions"/>
7545         <atom:link type="application/atom+xml;type=entry" rel="describedby"
7546 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7547 269806eb0d42/type"/>
7548         <atom:link type="application/atomsvc+xml" rel="service"
7549 href="http://cmisexample.oasis-open.org/rep1//service"/>
7550         <atom:published>2010-01-25T10:20:59.255-08:00</atom:published>
7551         <atom:summary type="html">HTML summary of Entry 6e27bada-b5a2-4a39-
7552 be2c-269806eb0d42</atom:summary>
7553         <atom:link rel="edit-media" href="http://cmisexample.oasis-
7554 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/edit-media"/>
7555         <atom:link rel="alternate" href="http://cmisexample.oasis-
7556 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/alternate"/>
7557         <atom:link type="application/atom+xml;type=feed" rel="up"
7558 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7559 269806eb0d42/parents"/>
7560         <atom:link type="application/atom+xml;type=feed" rel="version-history"
7561 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7562 269806eb0d42/allversions"/>
7563         <atom:link type="application/atom+xml;type=entry" rel="current-
7564 version" href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7565 269806eb0d42/latest"/>
7566         <atom:link type="application/atom+xml;type=feed"
7567 rel="http://docs.oasis-open.org/ns/cmisacl/200908/relationships"
7568 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7569 269806eb0d42/relationships"/>
7570         <atom:link type="application/atom+xml;type=feed"
7571 rel="http://docs.oasis-open.org/ns/cmisacl/200908/policies"
7572 href="http://cmisexample.oasis-open.org/rep1/6e27bada-b5a2-4a39-be2c-
7573 269806eb0d42/policies"/>
7574         <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7575 open.org/ns/cmisacl/200908/acl" href="http://cmisexample.oasis-
7576 open.org/rep1/6e27bada-b5a2-4a39-be2c-269806eb0d42/acl"/>
7577         <cmisra:object>
7578             <cmis:properties>
7579                 <cmis:propertyId localName="rep-cmis:objectId"
7580 propertyDefinitionId="cmis:objectId">
7581                     <cmis:value>6e27bada-b5a2-4a39-be2c-
7582 269806eb0d42</cmis:value>
7583                 </cmis:propertyId>
7584                 <cmis:propertyId localName="rep-cmis:objectTypeId"
7585 propertyDefinitionId="cmis:objectTypeId">
7586                     <cmis:value>document</cmis:value>
7587                 </cmis:propertyId>

```

```

7588         <cmis:propertyString localName="rep-cmis:name"
7589 propertyDefinitionId="cmis:name">
7590             <cmis:value>CMIS Example Document</cmis:value>
7591         </cmis:propertyString>
7592         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7593 propertyDefinitionId="cmis:creationDate">
7594             <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
7595         </cmis:propertyDateTime>
7596         <cmis:propertyDateTime localName="rep-
7597 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7598             <cmis:value>2010-01-25T10:20:59.271-08:00</cmis:value>
7599         </cmis:propertyDateTime>
7600         <cmis:propertyId localName="rep-cmis:baseTypeId"
7601 propertyDefinitionId="cmis:baseTypeId">
7602             <cmis:value>cmis:document</cmis:value>
7603         </cmis:propertyId>
7604         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7605 propertyDefinitionId="cmis:lastModifiedBy">
7606             <cmis:value>Al Brown</cmis:value>
7607         </cmis:propertyString>
7608         <cmis:propertyString localName="rep-cmis:createdBy"
7609 propertyDefinitionId="cmis:createdBy">
7610             <cmis:value>Al Brown</cmis:value>
7611         </cmis:propertyString>
7612         <cmis:propertyBoolean localName="rep-cmis:isLatestVersion"
7613 propertyDefinitionId="cmis:isLatestVersion">
7614             <cmis:value>>true</cmis:value>
7615         </cmis:propertyBoolean>
7616         <cmis:propertyBoolean localName="rep-
7617 cmis:isVersionSeriesCheckedOut"
7618 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7619             <cmis:value>>false</cmis:value>
7620         </cmis:propertyBoolean>
7621         <cmis:propertyBoolean localName="rep-cmis:isMajorVersion"
7622 propertyDefinitionId="cmis:isMajorVersion">
7623             <cmis:value>>false</cmis:value>
7624         </cmis:propertyBoolean>
7625         <cmis:propertyBoolean localName="rep-
7626 cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
7627             <cmis:value>>false</cmis:value>
7628         </cmis:propertyBoolean>
7629         <cmis:propertyBoolean localName="rep-cmis:isImmutable"
7630 propertyDefinitionId="cmis:isImmutable">
7631             <cmis:value>>false</cmis:value>
7632         </cmis:propertyBoolean>
7633         <cmis:propertyString localName="rep-cmis:checkinComment"
7634 propertyDefinitionId="cmis:checkinComment">
7635             <cmis:value>Checkin comment</cmis:value>
7636         </cmis:propertyString>
7637         <cmis:propertyString localName="rep-cmis:versionLabel"
7638 propertyDefinitionId="cmis:versionLabel">
7639             <cmis:value>0.1</cmis:value>
7640         </cmis:propertyString>
7641         <cmis:propertyString localName="rep-
7642 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
7643             <cmis:value>text/plain</cmis:value>
7644         </cmis:propertyString>
7645         <cmis:propertyString localName="rep-
7646 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
7647             <cmis:value>text.txt</cmis:value>
7648         </cmis:propertyString>
7649         <cmis:propertyInteger localName="rep-cmis:contentStreamLength"
7650 propertyDefinitionId="cmis:contentStreamLength">
7651             <cmis:value>4234</cmis:value>

```

```

7652         </cmis:propertyInteger>
7653         <cmis:propertyString displayName="Keywords for Document"
7654 localName="keywords" propertyDefinitionId="keywords">
7655         <cmis:value>document</cmis:value>
7656         <cmis:value>example</cmis:value>
7657         <cmis:value>sample</cmis:value>
7658         <cmis:value>cmis</cmis:value>
7659         </cmis:propertyString>
7660     </cmis:properties>
7661     <cmis:changeEventInfo>
7662         <cmis:changeType>updated</cmis:changeType>
7663         <cmis:changeTime>2010-01-25T10:20:59.271-
7664 08:00</cmis:changeTime>
7665     </cmis:changeEventInfo>
7666 </cmisra:object>
7667 <cmisra:pathSegment>invoice.pdf</cmisra:pathSegment>
7668 </atom:entry>
7669 </atom:feed>

```

7670

7671 Please also see the example documents included with the schema.

### 7672 **3.9.2.1 GET**

7673 The following optional parameters may be supplied:

- 7674 • filter
- 7675 • maxItems
- 7676 • includeACL
- 7677 • includePolicyIds
- 7678 • includeProperties
- 7679 • changeLogToken: If this parameter is specified, start the changes from the specified token. The
- 7680 changeLogToken is embedded in the paging link relations for normal iteration through the change
- 7681 list.

### 7682 **3.9.3 Folder Descendants**

7683 This is a hierarchical feed comprising items under a specified folder to a specified depth. This is available  
7684 via the link relation down with the application/cmistree+xml media type. Please see the Hierarchical Atom  
7685 Entries for more information on format.

7686

7687 If a repository does not support capabilityGetDescendants, then these resources SHOULD NOT be  
7688 exposed.

7689

7690 CMIS Services:

7691 GET: getDescendants

7692 DELETE: deleteTree

7693 Media Type: application/atom+xml;type=feed

7694

7695 Link Relations:

- 7696 • service: Points to service document containing the CMIS repository. The service document
- 7697 MUST contain only one workspace element.
  - 7698 ○ Media Type: application/atomsvc+xml
- 7699 • via: points to the atom entry of the folder generating this collection

- 7700 • up: points to the atom entry document for this folder's parent
  - 7701 ○ Media Type: application/atom+xml;type=entry
  - 7702 ○ If the root folder, this link relation MUST not be included.
- 7703 • down:
  - 7704 ○ points to the atom feed document representing the children feed for this same folder with
  - 7705 media type of application/atom+xml;type=entry
  - 7706 ○ Since this is the descendants, the descendants link SHOULD NOT be included
- 7707 • paging link relations MAY be included as appropriate: first, next, previous, last
  - 7708 ○ Repositories may support these paging link relations on a particular cmisra:children
  - 7709 element.
- 7710 • <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: <http://docs.oasis-open.org/ns/cmis/link/200908/foldertree>: Points to the folder tree for this folder
- 7711

7712

7713 The following CMIS Atom extension element MAY be included inside the atom feed:

- 7714 • cmisra:numItems

7715

7716 The following CMIS Atom extension element MUST be included inside the atom entries:

- 7717 • cmisra:object inside atom:entry
- 7718 • cmisra:pathSegment inside atom:entry
- 7719 • cmisra:children inside atom:entry

7720

7721 Example:

```

7722 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
7723 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmis/core/200908/"
7724 xmlns:cmism="http://docs.oasis-open.org/ns/cmis/messaging/200908/"
7725 xmlns:atom="http://www.w3.org/2005/Atom"
7726 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
7727 open.org/ns/cmis/restatom/200908/"
7728 <atom:title type="text">Feed for folder1</atom:title>
7729 <atom:author>
7730 <atom:name>Al Brown</atom:name>
7731 <atom:uri>http://www.ibm.com/</atom:uri>
7732 <atom:email>albertcbrown@us.ibm.com</atom:email>
7733 </atom:author>
7734 <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
7735 <atom:id>urn:uuid:cb0a47d4-8d09-46f9-9b09-584acad684af</atom:id>
7736 <atom:link type="application/atom+xml;type=feed" rel="self"
7737 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7738 040ec0c7c05a/3"/>
7739 <atom:link type="application/atomsvc+xml" rel="service"
7740 href="http://cmisexample.oasis-open.org/rep1//service"/>
7741 <atom:link type="application/atom+xml;type=entry" rel="via"
7742 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7743 040ec0c7c05a"/>
7744 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
7745 open.org/ns/cmis/link/200908/foldertree" href="http://cmisexample.oasis-
7746 open.org/rep1/f083dd6f-1465-4516-97ce-040ec0c7c05a/foldertree"/>
7747 <atom:link type="application/atom+xml;type=feed" rel="down"
7748 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7749 040ec0c7c05a/children"/>
7750 <atom:link type="application/atom+xml;type=entry" rel="up"
7751 href="http://cmisexample.oasis-open.org/rep1/03dcf5b8-5f82-45a1-b276-
7752 44d88069eec3"/>
7753 <cmisra:numItems>1</cmisra:numItems>

```

```

7754 <atom:entry>
7755 <atom:author>
7756 <atom:name>Al Brown</atom:name>
7757 <atom:uri>http://www.ibm.com/</atom:uri>
7758 <atom:email>albertcbrown@us.ibm.com</atom:email>
7759 </atom:author>
7760 <atom:content src="http://cmisexample.oasis-open.org/rep1/8e5a512c-
7761 8f2d-4387-a283-f3f30bbc312f"/>
7762 <atom:id>urn:uuid:8e5a512c-8f2d-4387-a283-f3f30bbc312f</atom:id>
7763 <atom:title type="text">CMIS Example Folder as Customer
7764 type</atom:title>
7765 <atom:updated>2010-01-25T10:20:59.364-08:00</atom:updated>
7766 <atom:link rel="self" href="http://cmisexample.oasis-
7767 open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
7768 <atom:link rel="edit" href="http://cmisexample.oasis-
7769 open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f"/>
7770 <atom:link type="application/cmismedia+xml;type=allowableActions"
7771 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/allowableactions"
7772 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7773 f3f30bbc312f/allowableactions"/>
7774 <atom:link type="application/atom+xml;type=entry" rel="describedby"
7775 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7776 f3f30bbc312f/type"/>
7777 <atom:link type="application/atomsvc+xml" rel="service"
7778 href="http://cmisexample.oasis-open.org/rep1//service"/>
7779 <atom:published>2010-01-25T10:20:59.380-08:00</atom:published>
7780 <atom:summary type="html">HTML summary of Entry 8e5a512c-8f2d-4387-
7781 a283-f3f30bbc312f</atom:summary>
7782 <atom:link type="application/atom+xml;type=entry" rel="up"
7783 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7784 f3f30bbc312f/up"/>
7785 <atom:link type="application/atom+xml;type=feed" rel="down"
7786 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7787 f3f30bbc312f/children"/>
7788 <atom:link type="application/cmistree+xml" rel="down"
7789 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7790 f3f30bbc312f/tree"/>
7791 <atom:link type="application/atom+xml;type=feed"
7792 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/foldertree"
7793 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7794 f3f30bbc312f/foldertree"/>
7795 <atom:link type="application/atom+xml;type=feed"
7796 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/relationships"
7797 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7798 f3f30bbc312f/relationships"/>
7799 <atom:link type="application/atom+xml;type=feed"
7800 rel="http://docs.oasis-open.org/ns/cmismedia/link/200908/policies"
7801 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7802 f3f30bbc312f/policies"/>
7803 <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
7804 open.org/ns/cmismedia/link/200908/acl" href="http://cmisexample.oasis-
7805 open.org/rep1/8e5a512c-8f2d-4387-a283-f3f30bbc312f/acl"/>
7806 <cmisra:object>
7807 <cmis:properties>
7808 <cmis:propertyId localName="rep-cmis:objectId"
7809 propertyDefinitionId="cmis:objectId">
7810 <cmis:value>8e5a512c-8f2d-4387-a283-
7811 f3f30bbc312f</cmis:value>
7812 </cmis:propertyId>
7813 <cmis:propertyId localName="rep-cmis:objectTypeId"
7814 propertyDefinitionId="cmis:objectTypeId">
7815 <cmis:value>customer</cmis:value>
7816 </cmis:propertyId>

```

```

7817         <cmis:propertyString localName="rep-cmis:name"
7818 propertyDefinitionId="cmis:name">
7819             <cmis:value>CMIS Example Folder as Customer
7820 type</cmis:value>
7821         </cmis:propertyString>
7822         <cmis:propertyDateTime localName="rep-cmis:creationDate"
7823 propertyDefinitionId="cmis:creationDate">
7824             <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7825         </cmis:propertyDateTime>
7826         <cmis:propertyDateTime localName="rep-
7827 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7828             <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7829         </cmis:propertyDateTime>
7830         <cmis:propertyId localName="rep-cmis:baseTypeId"
7831 propertyDefinitionId="cmis:baseTypeId">
7832             <cmis:value>cmis:folder</cmis:value>
7833         </cmis:propertyId>
7834         <cmis:propertyString localName="rep-cmis:lastModifiedBy"
7835 propertyDefinitionId="cmis:lastModifiedBy">
7836             <cmis:value>Al Brown</cmis:value>
7837         </cmis:propertyString>
7838         <cmis:propertyString localName="rep-cmis:createdBy"
7839 propertyDefinitionId="cmis:createdBy">
7840             <cmis:value>Al Brown</cmis:value>
7841         </cmis:propertyString>
7842         <cmis:propertyId localName="rep-cmis:parentId"
7843 propertyDefinitionId="cmis:parentId">
7844             <cmis:value>8e5a512c-8f2d-4387-a283-
7845 f3f30bbc312fup</cmis:value>
7846         </cmis:propertyId>
7847     </cmis:properties>
7848 </cmisra:object>
7849 <cmisra:pathSegment>customer</cmisra:pathSegment>
7850 <cmisra:children>
7851     <atom:feed>
7852         <atom:title type="text">CMIS Example Folder as Customer
7853 type</atom:title>
7854         <atom:author>
7855             <atom:name>Al Brown</atom:name>
7856             <atom:uri>http://www.ibm.com/</atom:uri>
7857             <atom:email>albertcbrown@us.ibm.com</atom:email>
7858         </atom:author>
7859         <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
7860         <atom:id>urn:uuid:67ee5e9f-d2e3-457d-9dec-
7861 be718e780452</atom:id>
7862         <atom:link type="application/atom+xml;type=feed" rel="self"
7863 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7864 f3f30bbc312f/3"/>
7865         <atom:link type="application/atomsvc+xml" rel="service"
7866 href="http://cmisexample.oasis-open.org/rep1//service"/>
7867         <atom:link type="application/atom+xml;type=entry" rel="via"
7868 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7869 f3f30bbc312f"/>
7870         <atom:link type="application/atom+xml;type=feed"
7871 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree"
7872 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7873 f3f30bbc312f/foldertree"/>
7874         <atom:link type="application/atom+xml;type=feed" rel="down"
7875 href="http://cmisexample.oasis-open.org/rep1/8e5a512c-8f2d-4387-a283-
7876 f3f30bbc312f/children"/>
7877         <atom:link type="application/atom+xml;type=entry" rel="up"
7878 href="http://cmisexample.oasis-open.org/rep1/f083dd6f-1465-4516-97ce-
7879 040ec0c7c05a"/>
7880     </atom:feed>
</cmisra:numItems>1</cmisra:numItems>

```

```

7881         <atom:entry>
7882             <atom:author>
7883                 <atom:name>Al Brown</atom:name>
7884                 <atom:uri>http://www.ibm.com/</atom:uri>
7885                 <atom:email>albertcbrown@us.ibm.com</atom:email>
7886             </atom:author>
7887             <atom:content src="http://cmisexample.oasis-
7888 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7889             <atom:id>urn:uuid:8c2dbba5-ea33-469f-a052-
7890 9f01e636c72a</atom:id>
7891             <atom:title type="text">CMIS Example Doc as Invoice
7892 type</atom:title>
7893             <atom:updated>2010-01-25T10:20:59.380-08:00</atom:updated>
7894             <atom:link rel="self" href="http://cmisexample.oasis-
7895 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7896             <atom:link rel="edit" href="http://cmisexample.oasis-
7897 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a"/>
7898             <atom:link
7899 type="application/cmisacl+xml;type=allowableActions" rel="http://docs.oasis-
7900 open.org/ns/cmisacl/link/200908/allowableactions" href="http://cmisexample.oasis-
7901 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a/allowableactions"/>
7902             <atom:link type="application/atom+xml;type=entry"
7903 rel="describedby" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-
7904 469f-a052-9f01e636c72a/type"/>
7905             <atom:link type="application/atomsvc+xml" rel="service"
7906 href="http://cmisexample.oasis-open.org/rep1//service"/>
7907             <atom:published>2010-01-25T10:20:59.380-
7908 08:00</atom:published>
7909             <atom:summary type="html">HTML summary of Entry 8c2dbba5-
7910 ea33-469f-a052-9f01e636c72a</atom:summary>
7911             <atom:link rel="edit-media"
7912 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7913 9f01e636c72a/edit-media"/>
7914             <atom:link rel="alternate" href="http://cmisexample.oasis-
7915 open.org/rep1/8c2dbba5-ea33-469f-a052-9f01e636c72a/alternate"/>
7916             <atom:link type="application/atom+xml;type=feed" rel="up"
7917 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7918 9f01e636c72a/parents"/>
7919             <atom:link type="application/atom+xml;type=feed"
7920 rel="version-history" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-
7921 ea33-469f-a052-9f01e636c72a/allversions"/>
7922             <atom:link type="application/atom+xml;type=entry"
7923 rel="current-version" href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-
7924 ea33-469f-a052-9f01e636c72a/latest"/>
7925             <atom:link type="application/atom+xml;type=feed"
7926 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/relationships"
7927 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7928 9f01e636c72a/relationships"/>
7929             <atom:link type="application/atom+xml;type=feed"
7930 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/policies"
7931 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7932 9f01e636c72a/policies"/>
7933             <atom:link type="application/cmisacl+xml"
7934 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/acl"
7935 href="http://cmisexample.oasis-open.org/rep1/8c2dbba5-ea33-469f-a052-
7936 9f01e636c72a/acl"/>
7937             <cmisra:object>
7938                 <cmis:properties>
7939                     <cmis:propertyId localName="rep-cmis:objectId"
7940 propertyDefinitionId="cmis:objectId">
7941 <cmis:value>8c2dbba5-ea33-469f-a052-9f01e636c72a</cmis:value>
7942                     </cmis:propertyId>
7943                     <cmis:propertyId localName="rep-cmis:objectTypeId"
7944 propertyDefinitionId="cmis:objectTypeId">

```

```

7945 <cmis:value>invoice</cmis:value>
7946 </cmis:propertyId>
7947 <cmis:propertyString localName="rep-cmis:name"
7948 propertyDefinitionId="cmis:name">
7949 <cmis:value>CMIS Example Doc as Invoice type</cmis:value>
7950 </cmis:propertyString>
7951 <cmis:propertyDateTime localName="rep-
7952 cmis:creationDate" propertyDefinitionId="cmis:creationDate">
7953 <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7954 </cmis:propertyDateTime>
7955 <cmis:propertyDateTime localName="rep-
7956 cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
7957 <cmis:value>2010-01-25T10:20:59.380-08:00</cmis:value>
7958 </cmis:propertyDateTime>
7959 <cmis:propertyId localName="rep-cmis:baseTypeId"
7960 propertyDefinitionId="cmis:baseTypeId">
7961 <cmis:value>cmis:document</cmis:value>
7962 </cmis:propertyId>
7963 <cmis:propertyString localName="rep-
7964 cmis:lastModifiedBy" propertyDefinitionId="cmis:lastModifiedBy">
7965 <cmis:value>Al Brown</cmis:value>
7966 </cmis:propertyString>
7967 <cmis:propertyString localName="rep-
7968 cmis:createdBy" propertyDefinitionId="cmis:createdBy">
7969 <cmis:value>Al Brown</cmis:value>
7970 </cmis:propertyString>
7971 <cmis:propertyBoolean localName="rep-
7972 cmis:isLatestVersion" propertyDefinitionId="cmis:isLatestVersion">
7973 <cmis:value>>true</cmis:value>
7974 </cmis:propertyBoolean>
7975 <cmis:propertyBoolean localName="rep-
7976 cmis:isVersionSeriesCheckedOut"
7977 propertyDefinitionId="cmis:isVersionSeriesCheckedOut">
7978 <cmis:value>>false</cmis:value>
7979 </cmis:propertyBoolean>
7980 <cmis:propertyBoolean localName="rep-
7981 cmis:isMajorVersion" propertyDefinitionId="cmis:isMajorVersion">
7982 <cmis:value>>false</cmis:value>
7983 </cmis:propertyBoolean>
7984 <cmis:propertyBoolean localName="rep-
7985 cmis:isLatestMajorVersion" propertyDefinitionId="cmis:isLatestMajorVersion">
7986 <cmis:value>>false</cmis:value>
7987 </cmis:propertyBoolean>
7988 <cmis:propertyBoolean localName="rep-
7989 cmis:isImmutable" propertyDefinitionId="cmis:isImmutable">
7990 <cmis:value>>false</cmis:value>
7991 </cmis:propertyBoolean>
7992 <cmis:propertyString localName="rep-
7993 cmis:checkinComment" propertyDefinitionId="cmis:checkinComment">
7994 <cmis:value>Checkin comment</cmis:value>
7995 </cmis:propertyString>
7996 <cmis:propertyString localName="rep-
7997 cmis:versionLabel" propertyDefinitionId="cmis:versionLabel">
7998 <cmis:value>0.1</cmis:value>
7999 </cmis:propertyString>
8000 <cmis:propertyString localName="rep-
8001 cmis:contentStreamMimeType" propertyDefinitionId="cmis:contentStreamMimeType">
8002 <cmis:value>text/plain</cmis:value>
8003 </cmis:propertyString>
8004 <cmis:propertyString localName="rep-
8005 cmis:contentStreamFileName" propertyDefinitionId="cmis:contentStreamFileName">
8006 <cmis:value>text.txt</cmis:value>
8007 </cmis:propertyString>

```

```

8008         <cmis:propertyInteger localName="rep-
8009 cmis:contentStreamLength" propertyDefinitionId="cmis:contentStreamLength">
8010 <cmis:value>4234</cmis:value>
8011         </cmis:propertyInteger>
8012         <cmis:propertyString displayName="Keywords for
8013 Document" localName="keywords" propertyDefinitionId="keywords">
8014 <cmis:value>document</cmis:value>
8015 <cmis:value>example</cmis:value>
8016 <cmis:value>sample</cmis:value>
8017 <cmis:value>cmis</cmis:value>
8018         </cmis:propertyString>
8019     </cmis:properties>
8020 </cmisra:object>
8021 <cmisra:pathSegment>invoice1.pdf</cmisra:pathSegment>
8022 </atom:entry>
8023 </atom:feed>
8024 </cmisra:children>
8025 </atom:entry>
8026 </atom:feed>

```

8027

8028 Please also see the example documents included with the schema.

### 8029 **3.9.3.1 GET**

8030 The following arguments may be supplied. Please see the domain model for more information:

- 8031 • filter
- 8032 • depth
- 8033 • includeAllowableActions
- 8034 • includeRelationships
- 8035 • renditionFilter
- 8036 • includePathSegment

### 8037 **3.9.3.2 DELETE**

8038 This deletes the folder and all sub-folders. The following arguments may be supplied. Please see the  
8039 domain model for more information:

- 8040 • continueOnFailure
- 8041 • unfileObjects

8042

8043 Status Code:

- 8044 • 200 OK if successful. Body contains entity describing the status
- 8045 • 202 Accepted, if accepted but deletion not yet taking place
- 8046 • 204 No Content, if successful with no content
- 8047 • 403 Forbidden, if permission is denied
- 8048 • 401 Unauthorized, if not authenticated
- 8049 • 500 Internal Server Error. The body SHOULD contain an entity describing the status

8050

8051 If the delete method does not delete all items, invoking GET with infinite depth on this URI will return the  
8052 items not deleted. Subsequent DELETE methods can be invoked on this URI.

8053 Note: If the repository does not implement get on this resource, or the canGetDescendants is false, there  
8054 is no mechanism to identify the resources that were not removed.

### 8055 3.9.4 Folder Tree

8056 This is a hierarchical feed comprising all the folders under a specified folder. This is available via the link  
8057 relation foldertree with media type application/atom+xml;type=feed. Please see the Hierarchical Atom  
8058 Entries for more information on format.

8059

8060 CMIS Services:

8061 GET: getFolderTree

8062 DELETE: deleteTree

8063 Media Type: application/atom+xml;type=feed

8064

8065 Link Relations:

- 8066 • service: Points to service document containing the CMIS repository. The service document  
8067 MUST contain only one workspace element.
  - 8068 ○ Media Type: application/atomsvc+xml
- 8069 • via: points to the atom entry of the folder generating this collection
- 8070 • up: points to the atom entry document of this folder's parent
  - 8071 ○ If the root folder, this link relation MUST not be included.
  - 8072 ○ Media Type: application/atom+xml;type=entry
- 8073 • down:
  - 8074 ○ application/atom+xml : Points to the atom feed document representing the children feed  
8075 for this same folder
  - 8076 ○ application/cmistree+xml: Points to the descendants feed of the same folder. If a  
8077 repository does not support capabilityGetDescendants, then this link SHOULD NOT be  
8078 included.
- 8079 • paging link relations MAY be included as appropriate: first, next, previous, last
  - 8080 ○ Repositories may support these paging link relations on a particular cmisra:children  
8081 element.

8082

8083 This feed contains a set of atom entries for each sub-folder in the folder.

8084

8085 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8086 • cmisra:numItems

8087

8088 The following CMIS Atom extension element MUST be included inside the atom entries:

- 8089 • cmisra:object inside atom:entry
- 8090 • cmisra:pathSegment inside atom:entry
- 8091 • cmisra:children inside atom:entry

8092

8093 Example:

```
8094 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
8095 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"  
8096 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"  
8097 xmlns:atom="http://www.w3.org/2005/Atom"  
8098 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-  
8099 open.org/ns/cmisis/restatom/200908/">  
8100 <atom:title type="text">FolderTree Feed of Folder1</atom:title>
```

```

8101 <atom:author>
8102   <atom:name>Al Brown</atom:name>
8103   <atom:uri>http://www.ibm.com/</atom:uri>
8104   <atom:email>albertcbrown@us.ibm.com</atom:email>
8105 </atom:author>
8106 <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
8107 <atom:id>urn:uuid:f87e5678-dd24-4214-9e71-635f060beb7d</atom:id>
8108 <atom:link type="application/atom+xml;type=feed" rel="self"
8109 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8110 b65edc3e1854/3"/>
8111 <atom:link type="application/atomsvc+xml" rel="service"
8112 href="http://cmisexample.oasis-open.org/rep1//service"/>
8113 <atom:link type="application/atom+xml;type=entry" rel="via"
8114 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8115 b65edc3e1854"/>
8116 <atom:link type="application/cmistree+xml" rel="down"
8117 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8118 b65edc3e1854/tree"/>
8119 <atom:link type="application/atom+xml;type=feed" rel="down"
8120 href="http://cmisexample.oasis-open.org/rep1/6e327a3c-a246-4cee-8176-
8121 b65edc3e1854/children"/>
8122 <atom:link type="application/atom+xml;type=entry" rel="up"
8123 href="http://cmisexample.oasis-open.org/rep1/3056c4d7-4e16-49cb-a750-
8124 ad7a3844alaa"/>
8125 <cmisra:numItems>1</cmisra:numItems>
8126 <atom:entry>
8127   <atom:author>
8128     <atom:name>Al Brown</atom:name>
8129     <atom:uri>http://www.ibm.com/</atom:uri>
8130     <atom:email>albertcbrown@us.ibm.com</atom:email>
8131   </atom:author>
8132   <atom:content src="http://cmisexample.oasis-open.org/rep1/c7b5a83e-
8133 37b6-4f5a-b646-50892252a180"/>
8134   <atom:id>urn:uuid:c7b5a83e-37b6-4f5a-b646-50892252a180</atom:id>
8135   <atom:title type="text">Customer Folder</atom:title>
8136   <atom:updated>2010-01-25T10:20:59.521-08:00</atom:updated>
8137   <atom:link rel="self" href="http://cmisexample.oasis-
8138 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
8139   <atom:link rel="edit" href="http://cmisexample.oasis-
8140 open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180"/>
8141   <atom:link type="application/cmis+xml;type=allowableActions"
8142 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
8143 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8144 50892252a180/allowableactions"/>
8145   <atom:link type="application/atom+xml;type=entry" rel="describedby"
8146 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8147 50892252a180/type"/>
8148   <atom:link type="application/atomsvc+xml" rel="service"
8149 href="http://cmisexample.oasis-open.org/rep1//service"/>
8150   <atom:published>2010-01-25T10:20:59.521-08:00</atom:published>
8151   <atom:summary type="html">HTML summary of Entry c7b5a83e-37b6-4f5a-
8152 b646-50892252a180</atom:summary>
8153   <atom:link type="application/atom+xml;type=entry" rel="up"
8154 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8155 50892252a180/up"/>
8156   <atom:link type="application/atom+xml;type=feed" rel="down"
8157 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8158 50892252a180/children"/>
8159   <atom:link type="application/cmistree+xml" rel="down"
8160 href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8161 50892252a180/tree"/>

```

```

8162     <atom:link type="application/atom+xml;type=feed"
8163     rel="http://docs.oasis-open.org/ns/cmism/link/200908/foldertree"
8164     href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8165     50892252a180/foldertree"/>
8166     <atom:link type="application/atom+xml;type=feed"
8167     rel="http://docs.oasis-open.org/ns/cmism/link/200908/relationships"
8168     href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8169     50892252a180/relationships"/>
8170     <atom:link type="application/atom+xml;type=feed"
8171     rel="http://docs.oasis-open.org/ns/cmism/link/200908/policies"
8172     href="http://cmisexample.oasis-open.org/rep1/c7b5a83e-37b6-4f5a-b646-
8173     50892252a180/policies"/>
8174     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8175     open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
8176     open.org/rep1/c7b5a83e-37b6-4f5a-b646-50892252a180/acl"/>
8177     <cmisra:object>
8178         <cmis:properties>
8179             <cmis:propertyId localName="rep-cmis:objectId"
8180     propertyDefinitionId="cmis:objectId">
8181                 <cmis:value>c7b5a83e-37b6-4f5a-b646-
8182     50892252a180</cmis:value>
8183             </cmis:propertyId>
8184             <cmis:propertyId localName="rep-cmis:objectTypeId"
8185     propertyDefinitionId="cmis:objectTypeId">
8186                 <cmis:value>customer</cmis:value>
8187             </cmis:propertyId>
8188             <cmis:propertyString localName="rep-cmis:name"
8189     propertyDefinitionId="cmis:name">
8190                 <cmis:value>Customer Folder</cmis:value>
8191             </cmis:propertyString>
8192             <cmis:propertyDateTime localName="rep-cmis:creationDate"
8193     propertyDefinitionId="cmis:creationDate">
8194                 <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
8195             </cmis:propertyDateTime>
8196             <cmis:propertyDateTime localName="rep-
8197     cmis:lastModificationDate" propertyDefinitionId="cmis:lastModificationDate">
8198                 <cmis:value>2010-01-25T10:20:59.521-08:00</cmis:value>
8199             </cmis:propertyDateTime>
8200             <cmis:propertyId localName="rep-cmis:baseTypeId"
8201     propertyDefinitionId="cmis:baseTypeId">
8202                 <cmis:value>cmis:folder</cmis:value>
8203             </cmis:propertyId>
8204             <cmis:propertyString localName="rep-cmis:lastModifiedBy"
8205     propertyDefinitionId="cmis:lastModifiedBy">
8206                 <cmis:value>Al Brown</cmis:value>
8207             </cmis:propertyString>
8208             <cmis:propertyString localName="rep-cmis:createdBy"
8209     propertyDefinitionId="cmis:createdBy">
8210                 <cmis:value>Al Brown</cmis:value>
8211             </cmis:propertyString>
8212             <cmis:propertyId localName="rep-cmis:parentId"
8213     propertyDefinitionId="cmis:parentId">
8214                 <cmis:value>c7b5a83e-37b6-4f5a-b646-
8215     50892252a180up</cmis:value>
8216             </cmis:propertyId>
8217         </cmis:properties>
8218     </cmisra:object>
8219     <cmisra:pathSegment>customer</cmisra:pathSegment>
8220 </atom:entry>
8221 </atom:feed>

```

8222

8223 Please also see the example documents included with the schema.

8224 **3.9.4.1 GET**

8225 The following arguments may be supplied. Please see the domain model for more information:

- 8226 • filter
- 8227 • depth
- 8228 • includeAllowableActions
- 8229 • includeRelationships
- 8230 • renditionFilter

8231 **3.9.4.2 DELETE**

8232 This is the same as DELETE on Folder Descendants. Please see that section.

8233 **3.9.5 AllVersions Feed**

8234 This is a feed comprised of all the versions of the given document.

8235 CMIS Services:

8236 GET: getAllVersions

8237 DELETE: deleteAllVersions

8238 Media Type: application/atom+xml;type=feed

8239

8240 The feed SHOULD contain the newest versions at the beginning of the feed.

8241

8242 Link Relations:

- 8243 • service: Points to service document containing the CMIS repository. The service document  
8244 MUST contain only one workspace element.
  - 8245 ○ Media Type: application/atomsvc+xml
- 8246 • via: points to the atom entry of the resource generating this collection
- 8247 • paging link relations as appropriate: first, next, previous, last

8248

8249 This feed contains a set of atom entries for each version in the version series

- 8250 • cmisra:object inside atom:entry
- 8251 • cmisra:children inside atom:entry if atom:entry represents a CMIS Folder

8252

8253 Example:

```
8254 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8255 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
8256 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
8257 xmlns:atom="http://www.w3.org/2005/Atom"
8258 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8259 open.org/ns/cmisis/restatom/200908/">
8260   <atom:title type="text">AllVersions for Document e8abd7cd-b9ec-4dba-9eaa-
8261 1bce2ae9977f</atom:title>
8262   <atom:author>
8263     <atom:name>Al Brown</atom:name>
8264     <atom:uri>http://www.ibm.com/</atom:uri>
8265     <atom:email>albertcbrown@us.ibm.com</atom:email>
8266   </atom:author>
8267   <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
8268   <atom:id>urn:uuid:5dc3d1c1-3e85-4720-acf8-cf98c96a5830</atom:id>
```

```

8269     <atom:link type="application/atom+xml;type=feed" rel="self"
8270 href="http://cmisexample.oasis-open.org/rep1/e8abd7cd-b9ec-4dba-9eaa-
8271 1bce2ae9977f/3"/>
8272     <atom:link type="application/atomsvc+xml" rel="service"
8273 href="http://cmisexample.oasis-open.org/rep1//service"/>
8274     <atom:link type="application/atom+xml;type=entry" rel="via"
8275 href="http://cmisexample.oasis-open.org/rep1/e8abd7cd-b9ec-4dba-9eaa-
8276 1bce2ae9977f"/>
8277     <cmisra:numItems>1</cmisra:numItems>
8278     <atom:entry>
8279       <atom:author>
8280         <atom:name>Al Brown</atom:name>
8281         <atom:uri>http://www.ibm.com/</atom:uri>
8282         <atom:email>albertcbrown@us.ibm.com</atom:email>
8283       </atom:author>
8284       <atom:content src="http://cmisexample.oasis-open.org/rep1/197033f2-
8285 ac11-4911-b5a3-60781fa5c281"/>
8286       <atom:id>urn:uuid:197033f2-ac11-4911-b5a3-60781fa5c281</atom:id>
8287       <atom:title type="text">Invoice (Version1)</atom:title>
8288       <atom:updated>2010-01-25T10:20:58.896-08:00</atom:updated>
8289       <atom:link rel="self" href="http://cmisexample.oasis-
8290 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281"/>
8291       <atom:link rel="edit" href="http://cmisexample.oasis-
8292 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281"/>
8293       <atom:link type="application/cmisacl+xml;type=allowableActions"
8294 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/allowableactions"
8295 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8296 60781fa5c281/allowableactions"/>
8297       <atom:link type="application/atom+xml;type=entry" rel="describedby"
8298 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8299 60781fa5c281/type"/>
8300       <atom:link type="application/atomsvc+xml" rel="service"
8301 href="http://cmisexample.oasis-open.org/rep1//service"/>
8302       <atom:published>2010-01-25T10:20:58.896-08:00</atom:published>
8303       <atom:summary type="html">HTML summary of Entry 197033f2-ac11-4911-
8304 b5a3-60781fa5c281</atom:summary>
8305       <atom:link rel="edit-media" href="http://cmisexample.oasis-
8306 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/edit-media"/>
8307       <atom:link rel="alternate" href="http://cmisexample.oasis-
8308 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/alternate"/>
8309       <atom:link type="application/atom+xml;type=feed" rel="up"
8310 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8311 60781fa5c281/parents"/>
8312       <atom:link type="application/atom+xml;type=feed" rel="version-history"
8313 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8314 60781fa5c281/allversions"/>
8315       <atom:link type="application/atom+xml;type=entry" rel="current-
8316 version" href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8317 60781fa5c281/latest"/>
8318       <atom:link type="application/atom+xml;type=feed"
8319 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/relationships"
8320 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8321 60781fa5c281/relationships"/>
8322       <atom:link type="application/atom+xml;type=feed"
8323 rel="http://docs.oasis-open.org/ns/cmisacl/link/200908/policies"
8324 href="http://cmisexample.oasis-open.org/rep1/197033f2-ac11-4911-b5a3-
8325 60781fa5c281/policies"/>
8326       <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
8327 open.org/ns/cmisacl/link/200908/acl" href="http://cmisexample.oasis-
8328 open.org/rep1/197033f2-ac11-4911-b5a3-60781fa5c281/acl"/>
8329       <cmisra:object>
8330         <cmis:properties>
8331           <cmis:propertyId localName="rep-cmis:objectId"
8332 propertyDefinitionId="cmis:objectId">

```

```
8333         <cmis:value>197033f2-ac11-4911-b5a3-
8334 60781fa5c281</cmis:value>
8335         </cmis:propertyId>
8336     </cmis:properties>
8337 </cmisra:object>
8338 </atom:entry>
8339 </atom:feed>
```

8340

8341 Please also see the example documents included with the schema.

### 8342 **3.9.5.1 GET**

8343 The following arguments may be supplied. Please see the domain model for more information:

- 8344 • filter
- 8345 • includeAllowableActions

### 8346 **3.9.5.2 DELETE**

8347 This removes the entire version history of the document.

8348

8349 Success HTTP code: 204

## 8350 **3.9.6 Type Descendants Feed**

8351 This is a feed described in the service document that contains all the types under a specific type in the  
8352 repository to a specific depth. If no parent type is specified, then the base types and their descendants  
8353 are returned in the feed which is equivalent to all types in the repository if depth is infinite. ~~The link~~  
8354 ~~relation is <http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>.~~ ~~The link relation is~~  
8355 ~~<http://docs.oasis-open.org/ns/cmis/link/200908/typedescendants>.~~

8356

8357 Types are nested using the CMIS hierarchy extension. Please see section 3.4.3.2 Hierarchy Navigation  
8358 Internet Draft Link Relations.

8359

8360 CMIS Services:

8361 GET: getTypeDescendants

8362

8363 Media Type: application/atom+xml;type=feed

8364

8365 Link Relations:

- 8366 • service: Points to service document containing the CMIS repository. The service document  
8367 MUST contain only one workspace element.
  - 8368 ○ Media Type: application/atomsvc+xml
- 8369 • via: points to the type definition whose descendents represent this feed. This link is not present if  
8370 no parent type is specified.
- 8371 • down: points to the children feed for the same type
- 8372 • up: points to the parent type definition
  - 8373 ○ If this is a descendants feed for a base object type, this link is not present.

8374

8375 The following CMIS Atom extension element MAY be included inside the atom feed:

- 8376 • cmisra:numItems

8377

8378 Example:

```
8379 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
8380 <atom:feed xmlns:cmis="http://docs.oasis-open.org/ns/cmisis/core/200908/"
8381 xmlns:cmism="http://docs.oasis-open.org/ns/cmisis/messaging/200908/"
8382 xmlns:atom="http://www.w3.org/2005/Atom"
8383 xmlns:app="http://www.w3.org/2007/app" xmlns:cmisra="http://docs.oasis-
8384 open.org/ns/cmisis/restatom/200908/">
8385   <atom:title type="text">Base Types</atom:title>
8386   <atom:author>
8387     <atom:name>Al Brown</atom:name>
8388     <atom:uri>http://www.ibm.com/</atom:uri>
8389     <atom:email>albertcbrown@us.ibm.com</atom:email>
8390   </atom:author>
8391   <atom:updated>2010-01-25T10:20:59.911-08:00</atom:updated>
8392   <atom:id>urn:uuid:c5d3d357-33ec-47c1-8436-563e0d94d2e5</atom:id>
8393   <atom:link type="application/atom+xml;type=feed" rel="self"
8394 href="http://cmisexample.oasis-open.org/repl//3"/>
8395   <atom:link type="application/atomsvc+xml" rel="service"
8396 href="http://cmisexample.oasis-open.org/repl//service"/>
8397   <atom:link type="application/atom+xml;type=entry" rel="via"
8398 href="http://cmisexample.oasis-open.org/repl//"/>
8399   <atom:link type="application/atom+xml;type=feed" rel="down"
8400 href="http://cmisexample.oasis-open.org/repl//children"/>
8401   <cmisra:numItems>1</cmisra:numItems>
8402   <atom:entry>
8403     <atom:author>
8404       <atom:name>Al Brown</atom:name>
8405       <atom:uri>http://www.ibm.com/</atom:uri>
8406       <atom:email>albertcbrown@us.ibm.com</atom:email>
8407     </atom:author>
8408     <atom:content>Type Definition for cmis:document</atom:content>
8409     <atom:id>http://cmisexample.oasis-
8410 open.org/repl/type/cmisis:document</atom:id>
8411     <atom:link type="application/atom+xml;type=entry" rel="self"
8412 href="http://cmisexample.oasis-open.org/repl/type/cmisis:document"/>
8413     <atom:link type="application/atomsvc+xml" rel="service"
8414 href="http://cmisexample.oasis-open.org/repl/type/cmisis:document"/>
8415     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8416 href="http://cmisexample.oasis-open.org/repl/type/cmisis:document"/>
8417     <atom:link type="application/atom+xml;type=entry" rel="up"
8418 href="http://cmisexample.oasis-open.org/repl/type/cmisis:document/parent"/>
8419     <atom:link type="application/atom+xml;type=feed" rel="down"
8420 href="http://cmisexample.oasis-
8421 open.org/repl/type/cmisis:document/children/flat"/>
8422     <atom:link type="application/cmistree+xml" rel="down"
8423 href="http://cmisexample.oasis-
8424 open.org/repl/type/cmisis:document/children/tree"/>
8425     <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
8426     <atom:summary type="html">HTML summary of Type Definition
8427 cmis:document</atom:summary>
8428     <atom:title type="text">Type Definition - cmis:document</atom:title>
8429     <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8430     <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8431     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8432 xsi:type="cmisis:cmisisTypeDocumentDefinitionType" cmisra:id="cmisis:document">
8433       <cmisis:id>dtcmisis:document</cmisis:id>
8434       <cmisis:localName>myrepname-cmisis:document</cmisis:localName>
8435       <cmisis:localNamespace xsi:nil="true"/>
8436       <cmisis:displayName>cmisis:document</cmisis:displayName>
8437       <cmisis:queryName>cmisis:document</cmisis:queryName>
8438       <cmisis:description>Description for type definition
8439 cmisis:document</cmisis:description>
```

```

8440         <cmis:baseId>cmis:document</cmis:baseId>
8441         <cmis:parentId>parent</cmis:parentId>
8442         <cmis:creatable>true</cmis:creatable>
8443         <cmis:fileable>true</cmis:fileable>
8444         <cmis:queryable>false</cmis:queryable>
8445         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8446
8447     <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8448         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8449         <cmis:controllableACL>true</cmis:controllableACL>
8450         <cmis:versionable>true</cmis:versionable>
8451         <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8452     </cmisra:type>
8453     <cmisra:children>
8454         <atom:feed>
8455             <atom:title type="text">Children for Document</atom:title>
8456             <atom:author>
8457                 <atom:name>Al Brown</atom:name>
8458                 <atom:uri>http://www.ibm.com/</atom:uri>
8459                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8460             </atom:author>
8461             <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8462             <atom:id>urn:uuid:6f1cdc44-bd89-41c0-8fad-
8463 89f3ad0f8f30</atom:id>
8464             <atom:link type="application/atom+xml;type=feed" rel="self"
8465 href="http://cmisexample.oasis-open.org/repl/cmis:document/3"/>
8466             <atom:link type="application/atomsvc+xml" rel="service"
8467 href="http://cmisexample.oasis-open.org/repl//service"/>
8468             <atom:link type="application/atom+xml;type=entry" rel="via"
8469 href="http://cmisexample.oasis-open.org/repl/cmis:document"/>
8470             <atom:link type="application/atom+xml;type=feed" rel="down"
8471 href="http://cmisexample.oasis-open.org/repl/cmis:document/children"/>
8472             <atom:link type="application/atom+xml;type=entry" rel="up"
8473 href="http://cmisexample.oasis-open.org/repl/document"/>
8474             <cmisra:numItems>1</cmisra:numItems>
8475             <atom:entry>
8476                 <atom:author>
8477                     <atom:name>Al Brown</atom:name>
8478                     <atom:uri>http://www.ibm.com/</atom:uri>
8479                     <atom:email>albertcbrown@us.ibm.com</atom:email>
8480                 </atom:author>
8481                 <atom:content>Type Definition for invoice-
8482 document</atom:content>
8483                 <atom:id>http://cmisexample.oasis-
8484 open.org/repl/type/invoice-document</atom:id>
8485                 <atom:link type="application/atom+xml;type=entry"
8486 rel="self" href="http://cmisexample.oasis-open.org/repl/type/invoice-
8487 document"/>
8488                 <atom:link type="application/atomsvc+xml" rel="service"
8489 href="http://cmisexample.oasis-open.org/repl/type/invoice-document"/>
8490                 <atom:link type="application/atom+xml;type=entry"
8491 rel="describedby" href="http://cmisexample.oasis-
8492 open.org/repl/type/cmis:document"/>
8493                 <atom:link type="application/atom+xml;type=entry" rel="up"
8494 href="http://cmisexample.oasis-open.org/repl/type/invoice-document/parent"/>
8495                 <atom:link type="application/atom+xml;type=feed"
8496 rel="down" href="http://cmisexample.oasis-open.org/repl/type/invoice-
8497 document/children/flat"/>
8498                 <atom:link type="application/cmistree+xml" rel="down"
8499 href="http://cmisexample.oasis-open.org/repl/type/invoice-
8500 document/children/tree"/>
8501                 <atom:published>2010-01-25T10:20:59.927-
8502 08:00</atom:published>

```

```

8503         <atom:summary type="html">HTML summary of Type Definition
8504 invoice-document</atom:summary>
8505         <atom:title type="text">Type Definition - invoice-
8506 document</atom:title>
8507         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8508         <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8509         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8510 instance" xsi:type="cmis:cmisTypeDocumentDefinitionType" cmisra:id="invoice-
8511 document">
8512             <cmis:id>dtinvoice-document</cmis:id>
8513             <cmis:localName>myrepname-invoice-
8514 document</cmis:localName>
8515             <cmis:localNamespace xsi:nil="true"/>
8516             <cmis:displayName>invoice-document</cmis:displayName>
8517             <cmis:queryName>invoice-document</cmis:queryName>
8518             <cmis:description>Description for type definition
8519 invoice-document</cmis:description>
8520             <cmis:baseId>cmis:document</cmis:baseId>
8521             <cmis:parentId>parent</cmis:parentId>
8522             <cmis:creatable>true</cmis:creatable>
8523             <cmis:fileable>true</cmis:fileable>
8524             <cmis:queryable>>false</cmis:queryable>
8525             <cmis:fulltextIndexed>>false</cmis:fulltextIndexed>
8526
8527 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8528
8529 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8530             <cmis:controllableACL>true</cmis:controllableACL>
8531             <cmis:versionable>true</cmis:versionable>
8532
8533 <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8534             </cmisra:type>
8535         </atom:entry>
8536     </atom:feed>
8537 </cmisra:children>
8538 </atom:entry>
8539 <atom:entry>
8540     <atom:author>
8541         <atom:name>Al Brown</atom:name>
8542         <atom:uri>http://www.ibm.com/</atom:uri>
8543         <atom:email>albertcbrown@us.ibm.com</atom:email>
8544     </atom:author>
8545     <atom:content>Type Definition for cmis:folder</atom:content>
8546     <atom:id>http://cmisexample.oasis-
8547 open.org/rep1/type/cmis:folder</atom:id>
8548     <atom:link type="application/atom+xml;type=entry" rel="self"
8549 href="http://cmisexample.oasis-open.org/rep1/type/cmis:folder"/>
8550     <atom:link type="application/atomsvc+xml" rel="service"
8551 href="http://cmisexample.oasis-open.org/rep1/type/cmis:folder"/>
8552     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8553 href="http://cmisexample.oasis-open.org/rep1/type/cmis:folder"/>
8554     <atom:link type="application/atom+xml;type=entry" rel="up"
8555 href="http://cmisexample.oasis-open.org/rep1/type/cmis:folder/parent"/>
8556     <atom:link type="application/atom+xml;type=feed" rel="down"
8557 href="http://cmisexample.oasis-open.org/rep1/type/cmis:folder/children/flat"/>
8558     <atom:link type="application/cmistree+xml" rel="down"
8559 href="http://cmisexample.oasis-open.org/rep1/type/cmis:folder/children/tree"/>
8560     <atom:published>2010-01-25T10:20:59.927-08:00</atom:published>
8561     <atom:summary type="html">HTML summary of Type Definition
8562 cmis:folder</atom:summary>
8563     <atom:title type="text">Type Definition - cmis:folder</atom:title>
8564     <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8565     <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>

```

```

8566     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8567 xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="cmis:folder">
8568     <cmis:id>dtcmis:folder</cmis:id>
8569     <cmis:localName>myrepname-cmis:folder</cmis:localName>
8570     <cmis:localNamespaces xsi:nil="true"/>
8571     <cmis:displayName>cmis:folder</cmis:displayName>
8572     <cmis:queryName>cmis:folder</cmis:queryName>
8573     <cmis:description>Description for type definition
8574 cmis:folder</cmis:description>
8575     <cmis:baseId>cmis:folder</cmis:baseId>
8576     <cmis:parentId>parent</cmis:parentId>
8577     <cmis:creatable>true</cmis:creatable>
8578     <cmis:fileable>true</cmis:fileable>
8579     <cmis:queryable>false</cmis:queryable>
8580     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8581
8582 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8583     <cmis:controllablePolicy>true</cmis:controllablePolicy>
8584     <cmis:controllableACL>true</cmis:controllableACL>
8585 </cmisra:type>
8586 <cmisra:children>
8587     <atom:feed>
8588         <atom:title type="text">Children for Folder</atom:title>
8589         <atom:author>
8590             <atom:name>Al Brown</atom:name>
8591             <atom:uri>http://www.ibm.com/</atom:uri>
8592             <atom:email>albertcbrown@us.ibm.com</atom:email>
8593         </atom:author>
8594         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8595         <atom:id>urn:uuid:361a3ac1-f7f7-47cb-b941-
8596 ae1200213fe0</atom:id>
8597         <atom:link type="application/atom+xml;type=feed" rel="self"
8598 href="http://cmisexample.oasis-open.org/repl/cmis:folder/3"/>
8599         <atom:link type="application/atomsvc+xml" rel="service"
8600 href="http://cmisexample.oasis-open.org/repl//service"/>
8601         <atom:link type="application/atom+xml;type=entry" rel="via"
8602 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8603         <atom:link type="application/atom+xml;type=feed" rel="down"
8604 href="http://cmisexample.oasis-open.org/repl/cmis:folder/children"/>
8605         <atom:link type="application/atom+xml;type=entry" rel="up"
8606 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8607         <cmisra:numItems>1</cmisra:numItems>
8608         <atom:entry>
8609             <atom:author>
8610                 <atom:name>Al Brown</atom:name>
8611                 <atom:uri>http://www.ibm.com/</atom:uri>
8612                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8613             </atom:author>
8614             <atom:content>Type Definition for customer-
8615 folder</atom:content>
8616             <atom:id>http://cmisexample.oasis-
8617 open.org/repl/type/customer-folder</atom:id>
8618             <atom:link type="application/atom+xml;type=entry"
8619 rel="self" href="http://cmisexample.oasis-open.org/repl/type/customer-
8620 folder"/>
8621             <atom:link type="application/atomsvc+xml" rel="service"
8622 href="http://cmisexample.oasis-open.org/repl/type/customer-folder"/>
8623             <atom:link type="application/atom+xml;type=entry"
8624 rel="describedby" href="http://cmisexample.oasis-
8625 open.org/repl/type/cmis:folder"/>
8626             <atom:link type="application/atom+xml;type=entry" rel="up"
8627 href="http://cmisexample.oasis-open.org/repl/type/customer-folder/parent"/>

```

```

8628         <atom:link type="application/atom+xml;type=feed"
8629 rel="down" href="http://cmisexample.oasis-open.org/repl/type/customer-
8630 folder/children/flat"/>
8631         <atom:link type="application/cmistree+xml" rel="down"
8632 href="http://cmisexample.oasis-open.org/repl/type/customer-
8633 folder/children/tree"/>
8634         <atom:published>2010-01-25T10:20:59.927-
8635 08:00</atom:published>
8636         <atom:summary type="html">HTML summary of Type Definition
8637 customer-folder</atom:summary>
8638         <atom:title type="text">Type Definition - customer-
8639 folder</atom:title>
8640         <atom:updated>2010-01-25T10:20:59.927-08:00</atom:updated>
8641         <app:edited>2010-01-25T10:20:59.927-08:00</app:edited>
8642         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8643 instance" xsi:type="cmis:cmisTypeFolderDefinitionType" cmisra:id="customer-
8644 folder">
8645             <cmis:id>dtcustomer-folder</cmis:id>
8646             <cmis:localName>myrepname-customer-
8647 folder</cmis:localName>
8648             <cmis:localNamespace xsi:nil="true"/>
8649             <cmis:displayName>customer-folder</cmis:displayName>
8650             <cmis:queryName>customer-folder</cmis:queryName>
8651             <cmis:description>Description for type definition
8652 customer-folder</cmis:description>
8653             <cmis:baseId>cmis:folder</cmis:baseId>
8654             <cmis:parentId>parent</cmis:parentId>
8655             <cmis:creatable>true</cmis:creatable>
8656             <cmis:fileable>true</cmis:fileable>
8657             <cmis:queryable>false</cmis:queryable>
8658             <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8659
8660 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8661
8662 <cmis:controllablePolicy>true</cmis:controllablePolicy>
8663         <cmis:controllableACL>true</cmis:controllableACL>
8664         </cmisra:type>
8665     </atom:entry>
8666 </atom:feed>
8667 </cmisra:children>
8668 </atom:entry>
8669 <atom:entry>
8670     <atom:author>
8671         <atom:name>Al Brown</atom:name>
8672         <atom:uri>http://www.ibm.com/</atom:uri>
8673         <atom:email>albertcbrown@us.ibm.com</atom:email>
8674     </atom:author>
8675     <atom:content>Type Definition for cmis:relationship</atom:content>
8676     <atom:id>http://cmisexample.oasis-
8677 open.org/repl/type/cmisis:relationship</atom:id>
8678     <atom:link type="application/atom+xml;type=entry" rel="self"
8679 href="http://cmisexample.oasis-open.org/repl/type/cmisis:relationship"/>
8680     <atom:link type="application/atomsvc+xml" rel="service"
8681 href="http://cmisexample.oasis-open.org/repl/type/cmisis:relationship"/>
8682     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8683 href="http://cmisexample.oasis-open.org/repl/type/cmisis:relationship"/>
8684     <atom:link type="application/atom+xml;type=entry" rel="up"
8685 href="http://cmisexample.oasis-open.org/repl/type/cmisis:relationship/parent"/>
8686     <atom:link type="application/atom+xml;type=feed" rel="down"
8687 href="http://cmisexample.oasis-
8688 open.org/repl/type/cmisis:relationship/children/flat"/>
8689     <atom:link type="application/cmistree+xml" rel="down"
8690 href="http://cmisexample.oasis-
8691 open.org/repl/type/cmisis:relationship/children/tree"/>

```

```

8692     <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
8693     <atom:summary type="html">HTML summary of Type Definition
8694 cmis:relationship</atom:summary>
8695     <atom:title type="text">Type Definition -
8696 cmis:relationship</atom:title>
8697     <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8698     <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8699     <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8700 xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8701 cmisra:id="cmis:relationship">
8702     <cmis:id>dtcmis:relationship</cmis:id>
8703     <cmis:localName>myrepname-cmis:relationship</cmis:localName>
8704     <cmis:localNamespace xsi:nil="true"/>
8705     <cmis:displayName>cmis:relationship</cmis:displayName>
8706     <cmis:queryName>cmis:relationship</cmis:queryName>
8707     <cmis:description>Description for type definition
8708 cmis:relationship</cmis:description>
8709     <cmis:baseId>cmis:relationship</cmis:baseId>
8710     <cmis:parentId>parent</cmis:parentId>
8711     <cmis:creatable>true</cmis:creatable>
8712     <cmis:fileable>false</cmis:fileable>
8713     <cmis:queryable>false</cmis:queryable>
8714     <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8715
8716 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8717     <cmis:controllablePolicy>true</cmis:controllablePolicy>
8718     <cmis:controllableACL>true</cmis:controllableACL>
8719     <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8720     <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8721     <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8722 </cmisra:type>
8723 <cmisra:children>
8724     <atom:feed>
8725         <atom:title type="text">Children for Relationship</atom:title>
8726         <atom:author>
8727             <atom:name>Al Brown</atom:name>
8728             <atom:uri>http://www.ibm.com/</atom:uri>
8729             <atom:email>albertcbrown@us.ibm.com</atom:email>
8730         </atom:author>
8731         <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8732         <atom:id>urn:uuid:9394ff3d-87c4-48c9-a951-
8733 ba725560faac</atom:id>
8734         <atom:link type="application/atom+xml;type=feed" rel="self"
8735 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/3"/>
8736         <atom:link type="application/atomsvc+xml" rel="service"
8737 href="http://cmisexample.oasis-open.org/repl//service"/>
8738         <atom:link type="application/atom+xml;type=entry" rel="via"
8739 href="http://cmisexample.oasis-open.org/repl/cmis:relationship"/>
8740         <atom:link type="application/atom+xml;type=feed" rel="down"
8741 href="http://cmisexample.oasis-open.org/repl/cmis:relationship/children"/>
8742         <atom:link type="application/atom+xml;type=entry" rel="up"
8743 href="http://cmisexample.oasis-open.org/repl/cmis:folder"/>
8744         <cmisra:numItems>1</cmisra:numItems>
8745         <atom:entry>
8746             <atom:author>
8747                 <atom:name>Al Brown</atom:name>
8748                 <atom:uri>http://www.ibm.com/</atom:uri>
8749                 <atom:email>albertcbrown@us.ibm.com</atom:email>
8750             </atom:author>
8751             <atom:content>Type Definition for customer-
8752 relationship</atom:content>
8753             <atom:id>http://cmisexample.oasis-
8754 open.org/repl/type/customer-relationship</atom:id>

```

```

8755         <atom:link type="application/atom+xml;type=entry"
8756 rel="self" href="http://cmisexample.oasis-open.org/repl/type/customer-
8757 relationship"/>
8758         <atom:link type="application/atomsvc+xml" rel="service"
8759 href="http://cmisexample.oasis-open.org/repl/type/customer-relationship"/>
8760         <atom:link type="application/atom+xml;type=entry"
8761 rel="describedby" href="http://cmisexample.oasis-
8762 open.org/repl/type/cmisa:relationship"/>
8763         <atom:link type="application/atom+xml;type=entry" rel="up"
8764 href="http://cmisexample.oasis-open.org/repl/type/customer-
8765 relationship/parent"/>
8766         <atom:link type="application/atom+xml;type=feed"
8767 rel="down" href="http://cmisexample.oasis-open.org/repl/type/customer-
8768 relationship/children/flat"/>
8769         <atom:link type="application/cmistree+xml" rel="down"
8770 href="http://cmisexample.oasis-open.org/repl/type/customer-
8771 relationship/children/tree"/>
8772         <atom:published>2010-01-25T10:20:59.943-
8773 08:00</atom:published>
8774         <atom:summary type="html">HTML summary of Type Definition
8775 customer-relationship</atom:summary>
8776         <atom:title type="text">Type Definition - customer-
8777 relationship</atom:title>
8778         <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8779         <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8780         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8781 instance" xsi:type="cmis:cmisTypeRelationshipDefinitionType"
8782 cmisra:id="customer-relationship">
8783             <cmis:id>dtcustomer-relationship</cmis:id>
8784             <cmis:localName>myrepname-customer-
8785 relationship</cmis:localName>
8786             <cmis:localNamespace xsi:nil="true"/>
8787             <cmis:displayName>customer-
8788 relationship</cmis:displayName>
8789             <cmis:queryName>customer-relationship</cmis:queryName>
8790             <cmis:description>Description for type definition
8791 customer-relationship</cmis:description>
8792             <cmis:baseId>cmis:relationship</cmis:baseId>
8793             <cmis:parentId>parent</cmis:parentId>
8794             <cmis:creatable>true</cmis:creatable>
8795             <cmis:fileable>false</cmis:fileable>
8796             <cmis:queryable>false</cmis:queryable>
8797             <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8798
8799             <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8800
8801             <cmis:controllablePolicy>true</cmis:controllablePolicy>
8802             <cmis:controllableACL>true</cmis:controllableACL>
8803
8804             <cmis:allowedSourceTypes>invoice</cmis:allowedSourceTypes>
8805
8806             <cmis:allowedSourceTypes>capitalinvoice</cmis:allowedSourceTypes>
8807
8808             <cmis:allowedTargetTypes>customer</cmis:allowedTargetTypes>
8809             </cmisra:type>
8810         </atom:entry>
8811     </atom:feed>
8812 </cmisra:children>
8813 </atom:entry>
8814 <atom:entry>
8815     <atom:author>
8816         <atom:name>Al Brown</atom:name>
8817         <atom:uri>http://www.ibm.com/</atom:uri>
8818         <atom:email>albertcbrown@us.ibm.com</atom:email>

```

```

8819         </atom:author>
8820         <atom:content>Type Definition for cmis:policy</atom:content>
8821         <atom:id>http://cmisexample.oasis-
8822 open.org/repl/type/cmis:policy</atom:id>
8823         <atom:link type="application/atom+xml;type=entry" rel="self"
8824 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8825         <atom:link type="application/atomsvc+xml" rel="service"
8826 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8827         <atom:link type="application/atom+xml;type=entry" rel="describedby"
8828 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy"/>
8829         <atom:link type="application/atom+xml;type=entry" rel="up"
8830 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/parent"/>
8831         <atom:link type="application/atom+xml;type=feed" rel="down"
8832 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/flat"/>
8833         <atom:link type="application/cmistree+xml" rel="down"
8834 href="http://cmisexample.oasis-open.org/repl/type/cmis:policy/children/tree"/>
8835         <atom:published>2010-01-25T10:20:59.943-08:00</atom:published>
8836         <atom:summary type="html">HTML summary of Type Definition
8837 cmis:policy</atom:summary>
8838         <atom:title type="text">Type Definition - cmis:policy</atom:title>
8839         <atom:updated>2010-01-25T10:20:59.943-08:00</atom:updated>
8840         <app:edited>2010-01-25T10:20:59.943-08:00</app:edited>
8841         <cmisra:type xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
8842 xsi:type="cmis:cmisTypePolicyDefinitionType" cmisra:id="cmis:policy">
8843         <cmis:id>dtcmis:policy</cmis:id>
8844         <cmis:localName>myrepname-cmis:policy</cmis:localName>
8845         <cmis:localNamespace xsi:nil="true"/>
8846         <cmis:displayName>cmis:policy</cmis:displayName>
8847         <cmis:queryName>cmis:policy</cmis:queryName>
8848         <cmis:description>Description for type definition
8849 cmis:policy</cmis:description>
8850         <cmis:baseId>cmis:policy</cmis:baseId>
8851         <cmis:parentId>parent</cmis:parentId>
8852         <cmis:creatable>true</cmis:creatable>
8853         <cmis:fileable>false</cmis:fileable>
8854         <cmis:queryable>false</cmis:queryable>
8855         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8856
8857 <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8858         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8859         <cmis:controllableACL>true</cmis:controllableACL>
8860     </cmisra:type>
8861 </atom:entry>
8862 </atom:feed>

```

8863

8864 Please also see the example documents included with the schema.

### 8865 **3.9.6.1 GET**

8866 The following arguments may be supplied. Please see the domain model for more information:

- 8867 • includePropertyDefinitions
- 8868 • depth

## 8869 **3.10 Resources**

8870 For any HTTP verb not specified on a resource, each implementation MAY chose to implement that HTTP  
8871 verb in a repository-specific manner.

### 8872 **3.10.1 Type Entry**

8873 This represents a type definition in the repository. This is enclosed as an atom entry

8874

8875 CMIS Services:

8876 GET: getTypeDefinition

8877 Media Type: application/atom+xml;type=entry

8878

8879 Link Relations:

- 8880 | • service: Points to service document containing the CMIS repository. The service document  
8881 | MUST contain only one workspace element.
  - 8882 | ○ Media Type: application/atomsvc+xml
- 8883 | • up: Points to the parent type as atom entry if applicable
- 8884 | • down: Points to the children of this type as atom feed if applicable
  - 8885 | ○ (Children) Media Type: application/atom+xml;type=feed points to the atom feed  
8886 | document representing the children feed for this same type
  - 8887 | ○ (Descendants) Media Type: application/cmistree+xml points to the atom feed document  
8888 | representing the descendants feed for this same type
- 8889 | • describedby: Points to the type definition atom entry of the base type of this type definition.

8890

8891 The following CMIS Atom extension element MUST be included inside the atom entry:

- 8892 | • cmisra:type

### 8893 3.10.1.1 GET

8894 There are no optional arguments for this resource.

8895

8896 Request:

```
8897 GET /obj/5070f89a-6f00-4acf-84e9-d8836a6c7d23 HTTP/1.1  
8898 Host: example.org  
8899
```

8900

8901 Response:

```
8902 HTTP/1.1 200 Ok  
8903 Date: Mon, 25 Jan 2010 10:21:00 -0800  
8904 Content-Length: 2995  
8905 Content-Type: application/atom+xml;type=entry  
8906 Location: http://cmisexample.oasis-open.org/repl/cmis:document  
8907  
8908  
8909 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
8910 <atom:entry xmlns:app="http://www.w3.org/2007/app"  
8911 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
8912 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-  
8913 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-  
8914 open.org/ns/cmis/restatom/200908/">  
8915 <atom:author>  
8916 <atom:name>Al Brown</atom:name>  
8917 <atom:uri>http://www.ibm.com/</atom:uri>  
8918 <atom:email>albertcbrown@us.ibm.com</atom:email>  
8919 </atom:author>  
8920 <atom:content>Type Definition for cmis:document</atom:content>  
8921 <atom:id>http://cmisexample.oasis-  
8922 open.org/repl/type/cmis:document</atom:id>  
8923 <atom:link type="application/atom+xml;type=entry" rel="self"  
8924 href="http://cmisexample.oasis-open.org/repl/type/cmis:document"/>
```

```

8925     <atom:link type="application/atomsvc+xml" rel="service"
8926 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document"/>
8927     <atom:link type="application/atom+xml;type=entry" rel="describedby"
8928 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document"/>
8929     <atom:link type="application/atom+xml;type=entry" rel="up"
8930 href="http://cmisexample.oasis-open.org/rep1/type/cmisis:document/parent"/>
8931     <atom:link type="application/atom+xml;type=feed" rel="down"
8932 href="http://cmisexample.oasis-
8933 open.org/rep1/type/cmisis:document/children/flat"/>
8934     <atom:link type="application/cmistree+xml" rel="down"
8935 href="http://cmisexample.oasis-
8936 open.org/rep1/type/cmisis:document/children/tree"/>
8937     <atom:published>2010-01-25T10:21:00.380-08:00</atom:published>
8938     <atom:summary type="html">HTML summary of Type Definition
8939 cmisis:document</atom:summary>
8940     <atom:title type="text">Type Definition - cmisis:document</atom:title>
8941     <atom:updated>2010-01-25T10:21:00.380-08:00</atom:updated>
8942     <app:edited>2010-01-25T10:21:00.380-08:00</app:edited>
8943     <cmisra:type xsi:type="cmis:cmisTypeDocumentDefinitionType"
8944 cmisra:id="cmisis:document" xmlns:xsi="http://www.w3.org/2001/XMLSchema-
8945 instance">
8946         <cmis:id>dtcmisis:document</cmis:id>
8947         <cmis:localName>myrepname-cmisis:document</cmis:localName>
8948         <cmis:localNamespace xsi:nil="true"/>
8949         <cmis:displayName>cmisis:document</cmis:displayName>
8950         <cmis:queryName>cmisis:document</cmis:queryName>
8951         <cmis:description>Description for type definition
8952 cmisis:document</cmis:description>
8953         <cmis:baseId>cmisis:document</cmis:baseId>
8954         <cmis:parentId>parent</cmis:parentId>
8955         <cmis:creatable>true</cmis:creatable>
8956         <cmis:fileable>true</cmis:fileable>
8957         <cmis:queryable>false</cmis:queryable>
8958         <cmis:fulltextIndexed>false</cmis:fulltextIndexed>
8959         <cmis:includedInSupertypeQuery>true</cmis:includedInSupertypeQuery>
8960         <cmis:controllablePolicy>true</cmis:controllablePolicy>
8961         <cmis:controllableACL>true</cmis:controllableACL>
8962         <cmis:versionable>true</cmis:versionable>
8963         <cmis:contentStreamAllowed>allowed</cmis:contentStreamAllowed>
8964     </cmisra:type>
8965 </atom:entry>
8966

```

8967

8968 Please also see the example documents included with the schema.

8969

### 8970 **3.10.2 Document Entry**

8971 This is a CMIS Document instance.

8972

8973 CMIS Services:

8974 GET: getObject, getObjectOfLatestVersion (getObject)

8975 PUT: updateProperties

8976 DELETE: deleteObject

8977 Media Type: application/atom+xml;type=entry

8978

8979 Link Relations:

- 8980 • self: Points to an URI that returns the atom entry for this document. Please see Atom for more  
8981 information.
- 8982 • edit: Points to an URI that accepts PUT of atom entry. Please see AtomPub for more information.
- 8983 • service: Points to service document containing the CMIS repository. The service document  
8984 MUST contain only one workspace element.
  - 8985 ○ Media Type: application/atomsvc+xml
- 8986 • up: Points to the atom feed containing the set of parents. If there is only one parent, the  
8987 repository MAY point this link relation directly to the atom entry of the parent.
- 8988 • version-history: Points to atom feed containing the versions of this document
  - 8989 ○ If the document is not versionable, this link relation may not be on the resource
- 8990 • current-version: Points to the latest version of the document
  - 8991 ○ Uses query parameter `"returnVersion"` and enumReturnVersion
  - 8992 ○ If this version is the current-version, this link relation may not be on the resource
- 8993 • edit-media:
  - 8994 ○ Same as setContentStream. Allows updating the content stream on this document
  - 8995 ○ Please see AtomPub for more information
- 8996 • working-copy: Points to the private working copy if it exists.
- 8997 • describedby: Points to the type definition as an atom entry for the type of this document entry.
- 8998 • alternate: this is used to identify the renditions available for the specified object. Please see the  
8999 Renditions section.
- 9000 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>~~: [http://docs.oasis-  
9002 open.org/ns/cmisis/link/200908/allowableactions](http://docs.oasis-<br/>9001 open.org/ns/cmisis/link/200908/allowableactions): Points to the allowable actions document for this  
object.
- 9003 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>~~: [http://docs.oasis-  
open.org/ns/cmisis/link/200908/relationships](http://docs.oasis-<br/>9004 open.org/ns/cmisis/link/200908/relationships): Points to the relationships feed for this object
- 9005 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/policies>~~: [http://docs.oasis-  
open.org/ns/cmisis/link/200908/policies](http://docs.oasis-<br/>9006 open.org/ns/cmisis/link/200908/policies): Points to the policy feed for this object.
- 9007 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/acl>~~: [http://docs.oasis-  
open.org/ns/cmisis/link/200908/acl](http://docs.oasis-<br/>9008 open.org/ns/cmisis/link/200908/acl): Points to ACL document for this object

9010 The following CMIS Atom extension element MUST be included inside the atom entry:

- 9011 • cmisra:object

### 9013 3.10.2.1 GET

9014 The following arguments may be supplied. Please see the domain model for more information:

- 9015 • returnVersion
  - 9016 ○ Used to differentiate between getObject() and getObjectOfLatestVersion().
  - 9017 ○ valid values are described by the schema element cmisra:enumReturnVersion
  - 9018 ○ If not specified, return the version specified by the URI
- 9019 • includeAllowableActions
- 9020 • includeRelationships
- 9021 • includePolicyIds
- 9022 • includeACL

- 9023 • filter
- 9024 • renditionFilter
  - 9025 ○ If not specified, renditions will not be included.

9026  
9027

9028 Request:

```
9029 GET /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId HTTP/1.1
9030 Host: example.org
9031
```

9032  
9033 Response:

```
9034 HTTP/1.1 200 Ok
9035 Date: Mon, 25 Jan 2010 10:21:00 -0800
9036 Content-Length: 3403
9037 Content-Type: application/atom+xml;type=entry
9038 Location: /obj/7c088887-5991-4b3a-9ad3-16379127e647?filter=cmis:objectId
9039
9040
9041 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9042 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9043 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9044 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9045 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9046 open.org/ns/cmisis/restatom/200908/">
9047   <atom:author>
9048     <atom:name>Al Brown</atom:name>
9049     <atom:uri>http://www.ibm.com/</atom:uri>
9050     <atom:email>albertcbrown@us.ibm.com</atom:email>
9051   </atom:author>
9052   <atom:content src="http://cmisexample.oasis-open.org/repl/7c088887-5991-
9053 4b3a-9ad3-16379127e647"/>
9054   <atom:id>urn:uuid:7c088887-5991-4b3a-9ad3-16379127e647</atom:id>
9055   <atom:title type="text">Invoice</atom:title>
9056   <atom:updated>2010-01-25T10:21:00.193-08:00</atom:updated>
9057   <atom:link rel="self" href="http://cmisexample.oasis-
9058 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647"/>
9059   <atom:link rel="edit" href="http://cmisexample.oasis-
9060 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647"/>
9061   <atom:link type="application/cmisis+xml;type=allowableActions"
9062 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9063 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9064 16379127e647/allowableactions"/>
9065   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9066 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9067 16379127e647/type"/>
9068   <atom:link type="application/atomsvc+xml" rel="service"
9069 href="http://cmisexample.oasis-open.org/repl//service"/>
9070   <atom:published>2010-01-25T10:21:00.193-08:00</atom:published>
9071   <atom:summary type="html">HTML summary of Entry 7c088887-5991-4b3a-9ad3-
9072 16379127e647</atom:summary>
9073   <atom:link rel="edit-media" href="http://cmisexample.oasis-
9074 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/edit-media"/>
9075   <atom:link rel="alternate" href="http://cmisexample.oasis-
9076 open.org/repl/7c088887-5991-4b3a-9ad3-16379127e647/alternate"/>
9077   <atom:link type="application/atom+xml;type=feed" rel="up"
9078 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9079 16379127e647/parents"/>
9080   <atom:link type="application/atom+xml;type=feed" rel="version-history"
9081 href="http://cmisexample.oasis-open.org/repl/7c088887-5991-4b3a-9ad3-
9082 16379127e647/allversions"/>
```

```

9083     <atom:link type="application/atom+xml;type=entry" rel="current-version"
9084 href="http://cmisexample.oasis-open.org/rep1/7c088887-5991-4b3a-9ad3-
9085 16379127e647/latest"/>
9086     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9087 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
9088 open.org/rep1/7c088887-5991-4b3a-9ad3-16379127e647/relationships"/>
9089     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9090 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
9091 open.org/rep1/7c088887-5991-4b3a-9ad3-16379127e647/policies"/>
9092     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
9093 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9094 open.org/rep1/7c088887-5991-4b3a-9ad3-16379127e647/acl"/>
9095     <cmisisra:object>
9096       <cmisisra:properties>
9097         <cmisisra:propertyId localName="rep-cmisisra:objectId"
9098 propertyDefinitionId="cmisisra:objectId">
9099           <cmisisra:value>7c088887-5991-4b3a-9ad3-16379127e647</cmisisra:value>
9100         </cmisisra:propertyId>
9101       </cmisisra:properties>
9102     </cmisisra:object>
9103 </atom:entry>
9104

```

9105

9106 Please also see the example documents included with the schema.

9107

### 9108 **3.10.2.2 PUT**

9109 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9110 properties are not included, the repository SHOULD NOT modify them.

9111

9112 The server SHOULD respond with:

- 9113 • HTTP Status Code 200
- 9114 • Response Body containing the updated atom entry

9115

### 9116 **3.10.2.3 DELETE**

9117 This removes the document.

9118 Success HTTP code: 204

## 9119 **3.10.3 Document Private Working Copy (PWC) Entry**

9120 This is the private working copy of the document (checkedout version of document)

9121 CMIS Services:

9122 GET: getObject

9123 PUT: updateProperties or checkIn

9124 DELETE: cancelCheckOut

9125 Media Type: application/atom+xml;type=entry

9126

9127 Link relations:

- 9128 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9129 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more  
9130 information.

- 9131 • service: Points to service document containing the CMIS repository. The service document  
9132 MUST contain only one workspace element.
- 9133 ○ Media Type: application/atomsvc+xml
- 9134 • up: Points to the atom feed containing the set of parents. If there is only one parent, the  
9135 repository MAY point this link relation directly to the atom entry of the parent.
- 9136 • version-history
- 9137 ○ Points to an URI that returns the feed associated with the version history
- 9138 • edit-media
- 9139 ○ Same as setContentStream. Allows updating the content stream on this document
- 9140 ○ Please see AtomPub for more information
- 9141 • via: atom entry that created this private working copy
- 9142 • describedby: Points to the type definition as an atom entry for the type of this PWC entry.
- 9143 • alternate: this is used to identify the renditions available for the specified object. Please see the  
9144 Renditions section.
- 9145 • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>:  
9146 <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions document for this  
9147 object.
- 9148 • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>:  
9149 <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for this object
- 9150 • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>:  
9151 <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 9152 • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>:  
9153 <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object

9154  
9155 The following element MUST be included inside the atom entry:

- 9156 • cmisra:object

9157

### 9158 3.10.3.1 GET

9159 The following arguments may be supplied. Please see the domain model for more information:

- 9160 • filter
- 9161 • includeAllowableActions
- 9162 • includeRelationships
- 9163 • renditionFilter
- 9164 ○ If not specified, renditions will not be included.

9165

9166 Request:

```
9167 GET /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId HTTP/1.1
9168 Host: example.org
9169
```

9170

9171 Response:

```
9172 HTTP/1.1 200 Ok
9173 Date: Mon, 25 Jan 2010 10:21:00 -0800
9174 Content-Length: 3564
9175 Content-Type: application/atom+xml;type=entry
```

9176 Location: /obj/3240a476-6de6-4ab2-978d-85ca2f4f3206?filter=cmis:objectId  
9177  
9178  
9179 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>  
9180 <atom:entry xmlns:app="http://www.w3.org/2007/app"  
9181 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-  
9182 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-  
9183 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-  
9184 open.org/ns/cmisis/restatom/200908/">  
9185 <atom:author>  
9186 <atom:name>Al Brown</atom:name>  
9187 <atom:uri>http://www.ibm.com/</atom:uri>  
9188 <atom:email>albertcbrown@us.ibm.com</atom:email>  
9189 </atom:author>  
9190 <atom:content src="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-  
9191 4ab2-978d-85ca2f4f3206"/>  
9192 <atom:id>urn:uuid:3240a476-6de6-4ab2-978d-85ca2f4f3206</atom:id>  
9193 <atom:title type="text">Invoice</atom:title>  
9194 <atom:updated>2010-01-25T10:21:00.333-08:00</atom:updated>  
9195 <atom:link rel="self" href="http://cmisexample.oasis-  
9196 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>  
9197 <atom:link rel="edit" href="http://cmisexample.oasis-  
9198 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206"/>  
9199 <atom:link type="application/cmisis+xml;type=allowableActions"  
9200 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"  
9201 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-  
9202 85ca2f4f3206/allowableactions"/>  
9203 <atom:link type="application/atom+xml;type=entry" rel="describedby"  
9204 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-  
9205 85ca2f4f3206/type"/>  
9206 <atom:link type="application/atomsvc+xml" rel="service"  
9207 href="http://cmisexample.oasis-open.org/rep1//service"/>  
9208 <atom:published>2010-01-25T10:21:00.333-08:00</atom:published>  
9209 <atom:summary type="html">HTML summary of Entry 3240a476-6de6-4ab2-978d-  
9210 85ca2f4f3206</atom:summary>  
9211 <atom:link rel="edit-media" href="http://cmisexample.oasis-  
9212 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/edit-media"/>  
9213 <atom:link rel="alternate" href="http://cmisexample.oasis-  
9214 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/alternate"/>  
9215 <atom:link type="application/atom+xml;type=feed" rel="up"  
9216 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-  
9217 85ca2f4f3206/parents"/>  
9218 <atom:link type="application/atom+xml;type=feed" rel="version-history"  
9219 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-  
9220 85ca2f4f3206/allversions"/>  
9221 <atom:link type="application/atom+xml;type=entry" rel="current-version"  
9222 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-  
9223 85ca2f4f3206/latest"/>  
9224 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-  
9225 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-  
9226 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/relationships"/>  
9227 <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-  
9228 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-  
9229 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/policies"/>  
9230 <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-  
9231 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-  
9232 open.org/rep1/3240a476-6de6-4ab2-978d-85ca2f4f3206/acl"/>  
9233 <atom:link type="application/atom+xml;type=feed" rel="working-copy"  
9234 href="http://cmisexample.oasis-open.org/rep1/3240a476-6de6-4ab2-978d-  
9235 85ca2f4f3206/pwc"/>  
9236 <cmisra:object>  
9237 <cmis:properties>  
9238 <cmis:propertyId localName="rep-cmisis:objectId"  
9239 propertyDefinitionId="cmisis:objectId">

```
9240         <cmis:value>3240a476-6de6-4ab2-978d-85ca2f4f3206</cmis:value>
9241         </cmis:propertyId>
9242     </cmis:properties>
9243 </cmisra:object>
9244 </atom:entry>
9245
```

9246  
9247 Please also see the example documents included with the schema.  
9248

### 9249 **3.10.3.2 PUT**

9250 This does a replacement of the atom entry with the atom entry document specified. If modifiable  
9251 properties (whencheckedout or readwrite) are not included, the repository SHOULD NOT modify them.

9252  
9253 The following arguments may be supplied. Please see the domain model for more information:

- 9254 • checkinComment
- 9255 • major
- 9256 • checkin
  - 9257 ○ Used to differentiate between updateProperties() or checkin() services. If TRUE, execute
  - 9258 checkin service.
  - 9259 ○ If this argument is specified as TRUE, then the body to PUT MAY be omitted if there are
  - 9260 no modifications to be made during checkin

9261  
9262 The server SHOULD respond with:

- 9263 • HTTP Status Code 200
- 9264 • Location header of the resource (if changed via checkin)
- 9265 • Response Body containing the updated atom entry

### 9266 **3.10.3.3 DELETE**

9267 This removes the document entry, in this case, cancels the check out. The PWC will be removed.

9268  
9269 Success HTTP code: 204

### 9270 **3.10.4 Folder Entry**

9271 This is a CMIS Folder instance. The properties of a folder map onto the feed tag.

9272 CMIS Services:

- 9273 GET: getObject
- 9274 PUT: updateProperties
- 9275 DELETE: deleteObject (this is deletion of the folder only and not any contained objects)

9276 Media Type: application/atom+xml;type=entry

9277  
9278 Link Relations:

- 9279 • self: Points to the URI to retrieve this atom entry. Please see Atom for more informationedit:
- 9280 Points to the URI to update this atom entry via POST. Please see AtomPub for more information.
- 9281 • service: Points to service document containing the CMIS repository. The service document
- 9282 MUST contain only one workspace element.

- 9283           ○ Media Type: application/atomsvc+xml
- 9284           • describedby: Points to the type definition as an atom entry for the type of this folder entry.
- 9285           • down: Points to the children of this folder
  - 9286           ○ application/atom+xml : Points to the atom feed document representing the children feed
  - 9287           for this same folder
  - 9288           ○ application/cmistree+xml: Points to the descendants feed of the same folder
- 9289           • up: Points to the atom entry for the parent
  - 9290           ○ If the root folder, this link will not be present
- 9291           • alternate: this is used to identify the renditions available for the specified object. Please see the
- 9292           Renditions section.
- 9293           • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions document for this
- 9294           object.
- 9295
- 9296           • <http://docs.oasis-open.org/ns/cmisis/link/200908/relationships>: Points to the relationships feed for this object
- 9297
- 9298           • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.
- 9299
- 9300           • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object
- 9301
- 9302           • <http://docs.oasis-open.org/ns/cmisis/link/200908/foldertree>: Points to the folder tree for this folder
- 9303

9304

The following CMIS Atom extension element MUST be included inside the atom entry:

- 9306           • cmisra:object

9307

### 9308   **3.10.4.1 GET**

9309   The following arguments may be supplied. Please see the domain model for more information:

- 9310           • filter
- 9311           • includeAllowableActions
- 9312           • includeRelationships
- 9313           • renditionFilter
  - 9314           ○ If not specified, renditions will not be included.

9315

9316   Request:

```
9317   GET /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId HTTP/1.1
9318   Host: example.org
```

9319

9320

9321   Response:

```
9322   HTTP/1.1 200 Ok
9323   Date: Mon, 25 Jan 2010 10:21:00 -0800
9324   Content-Length: 3332
9325   Content-Type: application/atom+xml;type=entry
9326   Location: /obj/cfc03a28-8240-471d-b4ba-6d8756cd5093?filter=cmis:objectId
```

9327  
9328

```

9329 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9330 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9331 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9332 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9333 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9334 open.org/ns/cmisis/restatom/200908/">
9335   <atom:author>
9336     <atom:name>Al Brown</atom:name>
9337     <atom:uri>http://www.ibm.com/</atom:uri>
9338     <atom:email>albertcbrown@us.ibm.com</atom:email>
9339   </atom:author>
9340   <atom:content src="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-
9341 471d-b4ba-6d8756cd5093"/>
9342   <atom:id>urn:uuid:cfc03a28-8240-471d-b4ba-6d8756cd5093</atom:id>
9343   <atom:title type="text">Customer Folder</atom:title>
9344   <atom:updated>2010-01-25T10:21:00.208-08:00</atom:updated>
9345   <atom:link rel="self" href="http://cmisexample.oasis-
9346 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
9347   <atom:link rel="edit" href="http://cmisexample.oasis-
9348 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093"/>
9349   <atom:link type="application/cmisis+xml;type=allowableActions"
9350 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9351 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9352 6d8756cd5093/allowableactions"/>
9353   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9354 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9355 6d8756cd5093/type"/>
9356   <atom:link type="application/atomsvc+xml" rel="service"
9357 href="http://cmisexample.oasis-open.org/repl//service"/>
9358   <atom:published>2010-01-25T10:21:00.208-08:00</atom:published>
9359   <atom:summary type="html">HTML summary of Entry cfc03a28-8240-471d-b4ba-
9360 6d8756cd5093</atom:summary>
9361   <atom:link type="application/atom+xml;type=entry" rel="up"
9362 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9363 6d8756cd5093/up"/>
9364   <atom:link type="application/atom+xml;type=feed" rel="down"
9365 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9366 6d8756cd5093/children"/>
9367   <atom:link type="application/cmisis+xml" rel="down"
9368 href="http://cmisexample.oasis-open.org/repl/cfc03a28-8240-471d-b4ba-
9369 6d8756cd5093/tree"/>
9370   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9371 open.org/ns/cmisis/link/200908/foldertree" href="http://cmisexample.oasis-
9372 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093/foldertree"/>
9373   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9374 open.org/ns/cmisis/link/200908/relationships" href="http://cmisexample.oasis-
9375 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093/relationships"/>
9376   <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9377 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
9378 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093/policies"/>
9379   <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
9380 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9381 open.org/repl/cfc03a28-8240-471d-b4ba-6d8756cd5093/acl"/>
9382   <cmisra:object>
9383     <cmis:properties>
9384       <cmis:propertyId localName="rep-cmisis:objectId"
9385 propertyDefinitionId="cmisis:objectId">
9386         <cmis:value>cfc03a28-8240-471d-b4ba-6d8756cd5093</cmis:value>
9387       </cmis:propertyId>
9388     </cmis:properties>
9389   </cmisra:object>
9390 </atom:entry>
9391

```

9392

9393 Please also see the example documents included with the schema.

9394

### 9395 **3.10.4.2 PUT**

9396 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9397 properties are not included, the repository SHOULD NOT modify them.

9398

9399 The server SHOULD respond with:

- 9400 • HTTP Status Code 200
- 9401 • Response Body containing the updated atom entry

9402

### 9403 **3.10.4.3 DELETE**

9404 This removes the object (folder) from the repository.

9405 Success HTTP code: 204

### 9406 **3.10.5 Relationship Entry**

9407 This is a CMIS relationship instance. These objects are exposed via `“relationships”` link type.

9408 CMIS Services:

- 9409 GET: getObject
- 9410 PUT: updateProperties
- 9411 DELETE: deleteObject

9412 Media Type: application/atom+xml;type=entry

9413

9414 Link Relations:

- 9415 • self: Points to the URI to retrieve this atom entry. Please see Atom for more information
- 9416 • edit: Points to the URI to update this atom entry via POST. Please see AtomPub for more  
9417 information.
- 9418 • service: Points to service document containing the CMIS repository. The service document  
9419 MUST contain only one workspace element.
  - 9420 ○ Media Type: application/atomsvc+xml
- 9421 • describedby: Points to the type definition as an atom entry for the type of this relationship entry.
- 9422 • <http://docs.oasis-open.org/ns/cmisis/link/200908/target>
- 9423 • <http://docs.oasis-open.org/ns/cmisis/link/200908/source>
- 9424 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>~~: [http://docs.oasis-](http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions)  
9425 [open.org/ns/cmisis/link/200908/allowableactions](http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions): Points to the allowable actions document for this  
9426 object.
- 9427 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/policies>~~: [open.org/ns/cmisis/link/200908/policies](http://docs.oasis-</a><br/>9428 <a href=): Points to the policy feed for this object.
- 9429 • ~~<http://docs.oasis-open.org/ns/cmisis/link/200908/acl>~~: [open.org/ns/cmisis/link/200908/acl](http://docs.oasis-</a><br/>9430 <a href=): Points to ACL document for this object

9431

9432 The following element MUST be included inside the atom entry:

- 9433 • cmisra:object

9434

### 9435 3.10.5.1 GET

9436 The following arguments may be supplied. Please see the domain model for more information:

- 9437 • filter
- 9438 • includeAllowableActions

9439

9440 Request:

```
9441 GET /obj/ad443afd-aala-4071-9735-1a49aac4e439?filter=cmis:objectId HTTP/1.1
9442 Host: example.org
9443
```

9444

9445 Response:

```
9446 HTTP/1.1 200 Ok
9447 Date: Mon, 25 Jan 2010 10:21:00 -0800
9448 Content-Length: 2861
9449 Content-Type: application/atom+xml;type=entry
9450 Location: /obj/ad443afd-aala-4071-9735-1a49aac4e439?filter=cmis:objectId
9451
9452
9453 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9454 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9455 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9456 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9457 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9458 open.org/ns/cmisis/restatom/200908/">
9459   <atom:author>
9460     <atom:name>Al Brown</atom:name>
9461     <atom:uri>http://www.ibm.com/</atom:uri>
9462     <atom:email>albertcbrown@us.ibm.com</atom:email>
9463   </atom:author>
9464   <atom:content src="http://cmisexample.oasis-open.org/repl/ad443afd-aala-
9465 4071-9735-1a49aac4e439"/>
9466   <atom:id>urn:uuid:ad443afd-aala-4071-9735-1a49aac4e439</atom:id>
9467   <atom:title type="text">Customer Relationship</atom:title>
9468   <atom:updated>2010-01-25T10:21:00.349-08:00</atom:updated>
9469   <atom:link rel="self" href="http://cmisexample.oasis-
9470 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439"/>
9471   <atom:link rel="edit" href="http://cmisexample.oasis-
9472 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439"/>
9473   <atom:link type="application/cmisis+xml;type=allowableActions"
9474 rel="http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions"
9475 href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9476 1a49aac4e439/allowableactions"/>
9477   <atom:link type="application/atom+xml;type=entry" rel="describedby"
9478 href="http://cmisexample.oasis-open.org/repl/ad443afd-aala-4071-9735-
9479 1a49aac4e439/type"/>
9480   <atom:link type="application/atomsvc+xml" rel="service"
9481 href="http://cmisexample.oasis-open.org/repl//service"/>
9482   <atom:published>2010-01-25T10:21:00.365-08:00</atom:published>
9483   <atom:summary type="html">HTML summary of Entry ad443afd-aala-4071-9735-
9484 1a49aac4e439</atom:summary>
9485   <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9486 open.org/ns/cmisis/link/200908/source" href="http://cmisexample.oasis-
9487 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/source"/>
9488   <atom:link type="application/atom+xml;type=entry" rel="http://docs.oasis-
9489 open.org/ns/cmisis/link/200908/target" href="http://cmisexample.oasis-
9490 open.org/repl/ad443afd-aala-4071-9735-1a49aac4e439/target"/>
```

```

9491     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9492 open.org/ns/cmisis/link/200908/policies" href="http://cmisexample.oasis-
9493 open.org/rep1/ad443afd-aa1a-4071-9735-1a49aac4e439/policies"/>
9494     <atom:link type="application/cmisisacl+xml" rel="http://docs.oasis-
9495 open.org/ns/cmisis/link/200908/acl" href="http://cmisexample.oasis-
9496 open.org/rep1/ad443afd-aa1a-4071-9735-1a49aac4e439/acl"/>
9497     <cmisra:object>
9498         <cmis:properties>
9499             <cmis:propertyId localName="rep-cmis:objectId"
9500 propertyDefinitionId="cmis:objectId">
9501                 <cmis:value>ad443afd-aa1a-4071-9735-1a49aac4e439</cmis:value>
9502             </cmis:propertyId>
9503         </cmis:properties>
9504     </cmisra:object>
9505 </atom:entry>
9506

```

9507

9508 Please also see the example documents included with the schema.

9509

### 9510 **3.10.5.2 PUT**

9511 This does a replacement of the atom entry with the atom entry document specified. If readwrite  
9512 properties are not included, the repository SHOULD NOT modify them.

9513

9514 The server SHOULD respond with:

- 9515 • HTTP Status Code 200
- 9516 • Response Body containing the updated atom entry

9517

### 9518 **3.10.5.3 DELETE**

9519 This removes the relationship entry.

9520 Successful HTTP code: 204

### 9521 **3.10.6 Policy Entry**

9522 This is a CMIS policy instance.

9523 CMIS Services:

- 9524 GET: getObject
- 9525 PUT: updateProperties
- 9526 DELETE: deleteObject or removePolicy

9527 Media Type: application/atom+xml;type=entry

9528

9529 Link Relations:

- 9530 • self
- 9531 • edit
- 9532 • service: Points to service document containing the CMIS repository. The service document  
9533 MUST contain only one workspace element.
  - 9534 ○ Media Type: application/atomsvc+xml
- 9535 • describedby: Points to the type definition as an atom entry for the type of this policy entry.

- 9536 | • alternate: this is used to identify the renditions available for the specified object. Please see the  
9537 Renditions section.
- 9538 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/allowableactions>: Points to the allowable actions document for this  
9539 object.
- 9541 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/policies>: Points to the policy feed for this object.  
9542
- 9543 | • <http://docs.oasis-open.org/ns/cmisis/link/200908/acl>: Points to ACL document for this object  
9544

9545

9546 The following element MUST be included inside the atom entry:

- 9547 | • `cmisra:object`

9548

### 9549 3.10.6.1 GET

9550 The following arguments may be supplied. Please see the domain model for more information:

- 9551 | • filter
- 9552 | • includeAllowableActions
- 9553 | • includeRelationships
- 9554 | • renditionFilter
- 9555 |     ○ If not specified, renditions will not be included.

9556

9557 Request:

```
9558 GET /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId HTTP/1.1
9559 Host: example.org
9560
```

9561

9562 Response:

```
9563 HTTP/1.1 200 Ok
9564 Date: Mon, 25 Jan 2010 10:21:00 -0800
9565 Content-Length: 2608
9566 Content-Type: application/atom+xml;type=entry
9567 Location: /obj/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0?filter=cmis:objectId
9568
9569
9570 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9571 <atom:entry xmlns:app="http://www.w3.org/2007/app"
9572 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9573 open.org/ns/cmisis/core/200908/" xmlns:cmism="http://docs.oasis-
9574 open.org/ns/cmisis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9575 open.org/ns/cmisis/restatom/200908/">
9576   <atom:author>
9577     <atom:name>Al Brown</atom:name>
9578     <atom:uri>http://www.ibm.com/</atom:uri>
9579     <atom:email>albertcbrown@us.ibm.com</atom:email>
9580   </atom:author>
9581   <atom:content src="http://cmisexample.oasis-open.org/repl/a09ed524-5f1b-
9582 4940-b2f0-4e4cd4631bf0"/>
9583   <atom:id>urn:uuid:a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</atom:id>
9584   <atom:title type="text">Security Policy</atom:title>
9585   <atom:updated>2010-01-25T10:21:00.318-08:00</atom:updated>
9586   <atom:link rel="self" href="http://cmisexample.oasis-
9587 open.org/repl/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
```

```

9588     <atom:link rel="edit" href="http://cmisexample.oasis-
9589 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0"/>
9590     <atom:link type="application/cmism+xml;type=allowableActions"
9591 rel="http://docs.oasis-open.org/ns/cmism/link/200908/allowableactions"
9592 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9593 4e4cd4631bf0/allowableactions"/>
9594     <atom:link type="application/atom+xml;type=entry" rel="describedby"
9595 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9596 4e4cd4631bf0/type"/>
9597     <atom:link type="application/atomsvc+xml" rel="service"
9598 href="http://cmisexample.oasis-open.org/rep1//service"/>
9599     <atom:published>2010-01-25T10:21:00.318-08:00</atom:published>
9600     <atom:summary type="html">HTML summary of Entry a09ed524-5f1b-4940-b2f0-
9601 4e4cd4631bf0</atom:summary>
9602     <atom:link type="application/atom+xml;type=feed" rel="up"
9603 href="http://cmisexample.oasis-open.org/rep1/a09ed524-5f1b-4940-b2f0-
9604 4e4cd4631bf0/parents"/>
9605     <atom:link type="application/atom+xml;type=feed" rel="http://docs.oasis-
9606 open.org/ns/cmism/link/200908/relationships" href="http://cmisexample.oasis-
9607 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0/relationships"/>
9608     <atom:link type="application/cmisacl+xml" rel="http://docs.oasis-
9609 open.org/ns/cmism/link/200908/acl" href="http://cmisexample.oasis-
9610 open.org/rep1/a09ed524-5f1b-4940-b2f0-4e4cd4631bf0/acl"/>
9611     <cmisra:object>
9612       <cmis:properties>
9613         <cmis:propertyId localName="rep-cmism:objectId"
9614 propertyDefinitionId="cmism:objectId">
9615           <cmis:value>a09ed524-5f1b-4940-b2f0-4e4cd4631bf0</cmis:value>
9616         </cmis:propertyId>
9617       </cmis:properties>
9618     </cmisra:object>
9619 </atom:entry>
9620

```

9621

9622 Please also see the example documents included with the schema.

9623

### 9624 **3.10.6.2 PUT**

9625 This does a replacement of the atom entry with the atom entry document specified. If read/write  
9626 properties are not included, the repository SHOULD NOT modify them.

9627

9628 The server SHOULD respond with:

- 9629 • HTTP Status Code 200
- 9630 • Response Body containing the updated atom entry

9631

### 9632 **3.10.6.3 DELETE**

9633 This removes the policy entry. If this policy entry was discovered through a policy collection on an object,  
9634 then removePolicy() is performed rather than deleteObject() on the policy itself.

9635

9636 Success HTTP code: 204

### 9637 **3.10.7 Content Stream**

9638 This is the content stream portion of the document object.

9639 CMIS Services:  
9640 GET: getContentStream  
9641 PUT: setContentStream  
9642 DELETE: deleteContentStream  
9643 Media Type: Mime/Type of resource (mime type of content stream on document)

### 9644 **3.10.7.1 GET**

9645 This returns the content stream.

9646

9647 It is RECOMMENDED that HTTP Range requests are supported on this resource. It is RECOMMENDED  
9648 that HTTP compression is also supported.

9649

9650 Please see RFC2616 for more information on HTTP Range requests.

### 9651 **3.10.7.2 PUT**

9652 This does a replacement of the content stream.

9653

9654 The following optional arguments may be supplied. Please see the domain model for more information:

- 9655 • overwriteFlag.
  - 9656 ○ If not specified, this defaults to `“true”` in this binding and behaves consistent with  
9657 AtomPub.

9658

9659 Success HTTP code: 200 (with content), 204 (without content) or 201 if a new resource is created.  
9660 Please see the HTTP specification for more information.

9661

9662 Returns headers:

- 9663 • Content-Location: URI for content stream
- 9664 • Location: URI for content stream

### 9665 **3.10.7.3 DELETE**

9666 This removes the content stream.

## 9667 **3.10.8 ACL Resource**

9668 CMIS Services:

9669 GET: getACL  
9670 PUT: applyACL

9671

9672 Media Type: application/cmisacl+xml

9673

### 9674 **3.10.8.1 GET**

9675 This returns the CMIS ACL for a specified object. The client will follow the link on the atom entry to get  
9676 the CMIS ACL for that object.

9677

9678 Request:

```
9679 GET /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef HTTP/1.1
9680 Host: example.org
9681
```

9682

9683 Response:

```
9684 HTTP/1.1 200 Ok
9685 Date: Mon, 25 Jan 2010 10:21:00 -0800
9686 Content-Length: 758
9687 Content-Type: application/cmisacl+xml
9688 Location: /objacl/fd79b7bd-2579-4ad1-aea2-eda89527fbef
9689
9690
9691 <?xml version="1.0" encoding="UTF-8" standalone="yes"?>
9692 <cmis:acl xmlns:app="http://www.w3.org/2007/app"
9693 xmlns:atom="http://www.w3.org/2005/Atom" xmlns:cmis="http://docs.oasis-
9694 open.org/ns/cmis/core/200908/" xmlns:cmism="http://docs.oasis-
9695 open.org/ns/cmis/messaging/200908/" xmlns:cmisra="http://docs.oasis-
9696 open.org/ns/cmis/restatom/200908/">
9697   <cmis:permission>
9698     <cmis:principal>
9699       <cmis:principalId>Al Brown</cmis:principalId>
9700     </cmis:principal>
9701     <cmis:permission>cmis:read</cmis:permission>
9702     <cmis:permission>cmis:write</cmis:permission>
9703     <cmis:permission>cmis:all</cmis:permission>
9704     <cmis:permission>publish</cmis:permission>
9705     <cmis:direct>true</cmis:direct>
9706   </cmis:permission>
9707 </cmis:acl>
9708
```

9709

9710 Please also see the example documents included with the schema.

---

## 9711 **4 Web Services Binding**

### 9712 **4.1 Overview**

9713 All services and operations defined in the Domain Model are presented in the Web Services binding.

9714 The WSDL for these services reference two XSD documents. One defines elements for the primary data  
9715 types of documents, folders, relationships and policies as well as collections of these types of objects.

9716 The second XSD defines the message formats for each of the CMIS services; the messages often refer  
9717 to the data types defined in the first XSD schema. The WSDL presents exactly the abstract services  
9718 defined in the Services section.

9719 The normative CMIS Web Services binding is defined by the WSDL and XSD as well as the details given  
9720 here in this part of the CMIS specification except the examples.

#### 9721 **4.1.1 WS-I**

9722 A CMIS Web Services binding **MUST** comply with WS-I Basic Profile 1.1 and Basic Security Profile 1.0.

#### 9723 **4.1.2 Authentication**

9724 A CMIS Web Services binding **SHOULD** support WS-Security 1.1 for Username Token Profile 1.1 and  
9725 **MAY** also support other authentication mechanisms. A CMIS repository **MAY** grant access to all or a  
9726 subset of the CMIS services to unauthenticated clients.

#### 9727 **4.1.3 Content Transfer**

9728 All endpoints of the Web Services binding **MUST** be MTOM enabled.

#### 9729 **4.1.4 Reporting Errors**

9730 Services **MUST** report errors via SOAP faults. The CMIS-Messaging.xsd defines a basic fault structure  
9731 that includes an error code and an error message and the WSDL for each service defines specific  
9732 messages that have the basic fault format.

## 9733 **4.2 Web Services Binding Mapping**

9734 The Domain Model defines all services, operations, parameters and objects of CMIS. The Web Services  
9735 binding is an exact one-to-one mapping of this definition with small exceptions that are explained in the  
9736 next section. Operations and parameters are named exactly after their counterparts in the Services  
9737 section. All rules and exceptions defined there apply to the Web Services binding. Optional parameters  
9738 and optional return values are not set if they are missing or their value is NULL.

## 9739 **4.3 Additions to the Services section**

### 9740 **4.3.1 updateProperties and checkIn Semantics**

9741 This binding supports partial properties updates. All properties passed to updateProperties or checkIn will  
9742 be updated to their new values. Properties that are passed without a value will be set to their default  
9743 value or un-set if no default value is defined. All others property values remain untouched.

### 9744 **4.3.2 Content Ranges**

9745 This binding supports the retrieval of content ranges. The operation getContentStream accepts two  
9746 optional parameters:

- 9747
- **Integer offset:** The first byte of the content to retrieve. Default value is 0.
- 9748
- **Integer length:** The length of the range in bytes. Default value is the size of the content minus the offset.
- 9749

9750

9751 If the **offset** value is greater than the size of the content the repository SHOULD throw a `constraint`  
9752 exception.

9753 If **offset + length** is greater than the size of the content the repository should deliver the content from the  
9754 offset to the end of the content.

9755

### 9756 **4.3.3 Extensions**

9757 On all input messages and some output messages exists an element called extension. This element is  
9758 used to provide vendor or repository-specific information between client and server.

9759 All of the types referenced by the schema also support `xs:any` for vendor or repository-specific  
9760 information.

### 9761 **4.3.4 Web Services Specific Structures**

9762 This binding requires specific structures that are not part of the general CMIS schema.

9763 Please also see the example request and response documents included with the schema.

#### 9764 **4.3.4.1 cmisFaultType and cmisFault**

9765 `cmisFaultType` and `cmisFault` SHOULD be used to generate SOAP faults. See 0

9766 Reporting Errors.

#### 9767 **4.3.4.2 cmisRepositoryEntryType**

9768 `cmisRepositoryEntryType` is the return structure of `getRepositories`. It contains the id and the name  
9769 of a repository.

#### 9770 **4.3.4.3 cmisTypeContainer**

9771 `cmisTypeContainer` is the return structure of `getTypeDescendants`. It holds a type hierarchy.

#### 9772 **4.3.4.4 cmisTypeDefinitionListType**

9773 `cmisTypeDefinitionListType` is the return structure of `getTypeChildren`. It contains a list of types,  
9774 the `hasMoreItems` flag and the `numItem` element.

#### 9775 **4.3.4.5 cmisObjectInFolderType, cmisObjectParentsType and** 9776 **cmisObjectInFolderContainerType**

9777 `cmisObjectInFolderType` holds, in addition to a `cmisObjectType` object, a path segment string. It  
9778 is used in all operations that support the `includePathSegments` parameter.

9779 `cmisObjectParentsType` is similar but has a relative path segment string instead of a path segment.  
9780 For details about path segments and relative path segments see section 2.1.5.3 Paths.

9781 `cmisObjectInFolderContainerType` contains a folder hierarchy.

9782 **4.3.4.6 cmisObjectListType and cmisObjectInFolderListType**

9783 `cmisObjectListType` and `cmisObjectInFolderListType` hold lists of `cmisObjectType` and  
 9784 `cmisObjectInFolderType` structures. They also contain the `hasMoreItems` flag and the `numItems`  
 9785 element that are returned by operations that return these lists.

9786 **4.3.4.7 cmisContentStreamType**

9787 `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

9788

9789 **4.3.4.8 cmisACLType**

9790 `cmisACLType` is the return structure of `getACL` and `applyACL`. It contains the current Access Control List  
 9791 (ACL) of the object and the `exact` flag that indicates if the ACL fully describes the permission of this  
 9792 object.

9793 **4.3.4.9 cmisExtensionType**

9794 `cmisExtensionType` is a placeholder for extensions. See 4.3.3 Extensions.

9795

---

## 9796 5 IANA Considerations

### 9797 5.1 Content-Type Registration

#### 9798 5.1.1 CMIS Query

9799 A CMIS Query Document, when serialized as XML 1.0, can be identified with the following media type:

9800

9801 MIME media type name: application

9802 MIME subtype name: cmisquery+xml

9803 Mandatory parameters: None

9804 Optional parameters:

9805 "charset": This parameter has semantics identical to the charset parameter of the  
9806 "application/xml" media type as specified in [RFC3023].

9807 Encoding considerations:

9808 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.

9809 Security considerations: As defined in this specification.

9810 In addition, as this media type uses the "+xml" convention, it shares the same security  
9811 considerations as described in [RFC3023], Section 10.

9812 Interoperability considerations:

9813 There are no known interoperability issues.

9814 Published specification: This specification.

9815 Applications that use this media type:

9816 No known applications currently use this media type.

9817 Additional information:

9818 Magic number(s):

9819 As specified for "application/xml" in [RFC3023], Section 3.2.

9820 File extension: .cmisquery

9821 Fragment identifiers:

9822 As specified for "application/xml" in [RFC3023], Section 5.

9823 Base URI:

9824 As specified in [RFC3023], Section 6.

9825 Macintosh File Type code: TEXT

9826 Person and email address to contact for further information:

9827 | ~~OASIS~~[OASIS](http://www.oasis-open.org) CMIS TC <cmis@lists.oasis-open.org>

9828 Intended usage: COMMON

9829 Author/Change controller: IESG

#### 9830 5.1.2 CMIS AllowableActions

9831 A CMIS Allowable Actions Document, when serialized as XML 1.0, can be identified with the following  
9832 media type:

9833

9834 MIME media type name: application  
9835 MIME subtype name: cmisallowableactions +xml  
9836 Mandatory parameters: None.  
9837 Optional parameters:  
9838 "charset": This parameter has semantics identical to the charset parameter of the  
9839 "application/xml" media type as specified in [RFC3023].  
9840 Encoding considerations:  
9841 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9842 Security considerations: As defined in this specification.  
9843 In addition, as this media type uses the "+xml" convention, it shares the same security  
9844 considerations as described in [RFC3023], Section 10.  
9845 Interoperability considerations:  
9846 There are no known interoperability issues.  
9847 Published specification: This specification.  
9848 Applications that use this media type:  
9849 No known applications currently use this media type.  
9850 Additional information:  
9851 Magic number(s):  
9852 As specified for "application/xml" in [RFC3023], Section 3.2.  
9853 File extension: .cmisallowableactions  
9854 Fragment identifiers:  
9855 As specified for "application/xml" in [RFC3023], Section 5.  
9856 Base URI:  
9857 As specified in [RFC3023], Section 6.  
9858 Macintosh File Type code: TEXT  
9859 Person and email address to contact for further information:  
9860 | [OASIS OASIS CMIS TC <cmis@lists.oasis-open.org>](mailto:cmis@lists.oasis-open.org)  
9861 Intended usage: COMMON  
9862 Author/Change controller: IESG  
9863

### 9864 **5.1.3 CMIS Tree**

9865 A CMIS Tree Document, when serialized as XML 1.0, can be identified with the following media type:

9866  
9867 MIME media type name: application  
9868 MIME subtype name: cmistree +xml  
9869 Mandatory parameters: None.  
9870 Optional parameters:  
9871 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9872 type as specified in [RFC3023].  
9873 Encoding considerations:  
9874 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9875 Security considerations: As defined in this specification.

9876 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9877 described in [RFC3023], Section 10.

9878 Interoperability considerations:  
9879 There are no known interoperability issues.

9880 Published specification: This specification.

9881 Applications that use this media type:  
9882 No known applications currently use this media type.

9883 Additional information:  
9884 Magic number(s):  
9885 As specified for "application/xml" in [RFC3023], Section 3.2.  
9886 File extension: .cmistree  
9887 Fragment identifiers:  
9888 As specified for "application/xml" in [RFC3023], Section 5.  
9889 Base URI:  
9890 As specified in [RFC3023], Section 6.  
9891 Macintosh File Type code: TEXT  
9892 Person and email address to contact for further information:  
9893 | ~~OASIS~~OASIS CMIS TC <cmis@lists.oasis-open.org>  
9894 Intended usage: COMMON  
9895 Author/Change controller: IESG  
9896

#### 9897 **5.1.4 CMIS Atom**

9898 A CMIS Atom Document, when serialized as XML 1.0, can be identified with the following media type:  
9899

9900 MIME media type name: application  
9901 MIME subtype name: cmisatom+xml  
9902 Mandatory parameters: None.  
9903 Optional parameters:  
9904 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9905 type as specified in [RFC3023].  
9906 | ~~"type="~~: This parameter has semantics identical to the type parameter of the ~~"application/atom+xml"~~ as  
9907 specified in [RFC4287]  
9908 Encoding considerations:  
9909 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9910 Security considerations: As defined in this specification.  
9911 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9912 described in [RFC3023], Section 10.

9913 Interoperability considerations:  
9914 There are no known interoperability issues.

9915 Published specification: This specification.

9916 Applications that use this media type:  
9917 No known applications currently use this media type.

9918 Additional information:

9919 Magic number(s):  
9920 As specified for "application/xml" in [RFC3023], Section 3.2.  
9921 File extension: .cmisatom  
9922 Fragment identifiers:  
9923 As specified for "application/xml" in [RFC3023], Section 5.  
9924 Base URI:  
9925 As specified in [RFC3023], Section 6.  
9926 Macintosh File Type code: TEXT  
9927 Person and email address to contact for further information:  
9928 | [OASISOASIS](#) CMIS TC <cmis@lists.oasis-open.org>  
9929 Intended usage: COMMON  
9930 Author/Change controller: IESG  
9931  
9932 Please see section 3.1.1 on why this media type is needed above the Atom Media Type.

### 9933 **5.1.5 CMIS ACL**

9934 A CMIS ACL Document, when serialized as XML 1.0, can be identified with the following media type:

9935  
9936 MIME media type name: application  
9937 MIME subtype name: cmisacl+xml  
9938 Mandatory parameters: None.  
9939 Optional parameters:  
9940 "charset": This parameter has semantics identical to the charset parameter of the "application/xml" media  
9941 type as specified in [RFC3023].  
9942 Encoding considerations:  
9943 Identical to those of "application/xml" as described in [RFC3023], Section 3.2.  
9944 Security considerations: As defined in this specification.  
9945 In addition, as this media type uses the "+xml" convention, it shares the same security considerations as  
9946 described in [RFC3023], Section 10.  
9947 Interoperability considerations:  
9948 There are no known interoperability issues.  
9949 Published specification: This specification.  
9950 Applications that use this media type:  
9951 No known applications currently use this media type.  
9952 Additional information:  
9953 Magic number(s):  
9954 As specified for "application/xml" in [RFC3023], Section 3.2.  
9955 File extension: .cmisacl  
9956 Fragment identifiers:  
9957 As specified for "application/xml" in [RFC3023], Section 5.  
9958 Base URI:  
9959 As specified in [RFC3023], Section 6.  
9960 Macintosh File Type code: TEXT

9961 Person and email address to contact for further information:  
9962 | ~~OASIS~~OASIS CMIS TC <cmis@lists.oasis-open.org>  
9963 Intended usage: COMMON  
9964 Author/Change controller: IESG  
9965

---

## 6 Conformance

9966

9967 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level  
9968 requirements defined within this specification.

9969 Specification:

9970 This specification references a number of other specifications (see the table above). In order to  
9971 comply with this specification, an implementation MUST implement the portions of referenced  
9972 specifications necessary to comply with the required provisions of this specification. Additionally,  
9973 the implementation of the portions of the referenced specifications that are specifically cited in  
9974 this specification MUST comply with the rules for those portions as established in the referenced  
9975 specification.

9976

9977 An implementation conforms to this specification if it satisfies all of the MUST or REQUIRED level  
9978 requirements defined within this specification.

9979

9980

9981 Domain Model:

9982 Normative text within this specification takes precedence over the CMIS Core XML Schema.

9983 That is, the normative text in this specification further constrains the schemas and/or WSDL that  
9984 are part of this specification; and this specification contains further constraints on the elements  
9985 defined in referenced schemas.

9986

9987 Clients:

9988 Client implementations MAY implement either Restful AtomPub Binding or the Web  
9989 Services Binding.

9990

9991 Repositories:

9992 Repositories MUST implement the following CMIS protocol bindings:

9993 

- [i.](#) Restful AtomPub Binding

9994 

- [ii.](#) Web Services Binding

9995

9996 Rest Binding:

9997 This specification references a number of other specifications. In order to comply with this  
9998 specification, an implementation MUST implement the portions of referenced specifications  
9999 necessary to comply with the required provisions of this specification. Additionally, the  
10000 implementation of the portions of the referenced specifications that are specifically cited in this  
10001 specification MUST comply with the rules for those portions as established in the referenced  
10002 specification.

10003 Additionally normative text within this specification takes precedence over the CMIS RestAtom  
10004 XML Schema. That is, the normative text in this specification further constrains the schemas  
10005 and/or WSDL that are part of this specification; and this specification contains further constraints  
10006 on the elements defined in referenced schemas.

10007 The CMIS RestAtom XML takes precedence over any examples or non-normative outlines  
10008 included either in this document or as standalone examples.

10009

10010 Web Services Binding:

10011 Normative text within this specification takes precedence over the CMIS Messaging XML and  
10012 CMIS WSDL. That is, the normative text in this specification further constrains the schemas and  
10013 WSDL that are part of this specification; and this specification contains further constraints on the  
10014 elements defined in referenced schemas.

10015 The CMIS Messaging XML and CMIS WSDL takes precedence over any examples or non-  
10016 normative outlines included either in this document or as standalone examples.

10017

---

## A. Acknowledgements

10018 The following individuals have participated in the creation of this specification and are gratefully  
10019 acknowledged:

10020

10021 **Participants:**

10022 Philippe Allart, Adullact  
10023 Florian Bartels, fme AG  
10024 Fred Boiscuvier, Exalead, Inc.  
10025 Al Brown, IBM  
10026 Jay Brown, IBM  
10027 Mark Carlson, Sun Microsystems  
10028 Derek Carr, IBM  
10029 David Caruana, Alfresco Software  
10030 Eric Chan, Oracle Corporation  
10031 Sameer Charles, Magnolia International AG  
10032 Derek Chow, Genus Technologies, LLC  
10033 David Choy, EMC Corporation  
10034 Scott Conroy, Individual  
10035 Cornelia Davis, EMC Corporation  
10036 Doug Domeny, Ektron  
10037 Kevin Dorr, Flatirons Solutions Corporation  
10038 Jason Dubreuil, Fidelity Investments  
10039 Michael Duerig, Day Software  
10040 Randy Dufault, Genus Technologies, LLC  
10041 Will Ezell, dotCMS  
10042 Betsy Fanning, AIIM  
10043 Steffen Frederiksen, Content Technologies ApS  
10044 Stephan Friedl, Quark  
10045 Dustin Friesenhahn, Microsoft Corporation  
10046 Gary Gershon, Individual  
10047 Paul Goetz, SAP AG  
10048 Jens Goldhammer, fme AG  
10049 Gregory Grefenstette, Exalead, Inc.  
10050 Florent Guillaume, Nuxeo  
10051 Ethan Gur-esh, Microsoft Corporation  
10052 Alexander Haag, WeWebU Software AG  
10053 Dennis Hamilton, Individual  
10054 Martin Hermes, SAP AG  
10055 Jens Huebel, Open Text Corporation  
10056 David Izatt, Structured Software Systems Limited (3SL)  
10057 Gershon Janssen, Individual  
10058 Raphael Jean, Entropysoft  
10059 Volker John, Saperion AG  
10060 Shane Johnson, Citytech, Inc.  
10061 Christophe Kijewska, Adullact  
10062 Ijonas Kisselbach, Vamosa  
10063 Mark Klamerus, Individual  
10064 Stephan Klevenz, SAP AG  
10065 Boris Kraft, Magnolia International AG  
10066 Alison Macmillan, Oracle Corporation  
10067 Michael Marth, Day Software  
10068 Mary McRae, OASIS  
10069 Ryan McVeigh, Oracle Corporation

10070 Juerg Meier, fme AG  
10071 Gregory Melahn, IBM  
10072 Pat Miller, Microsoft Corporation  
10073 Florian Müller, Open Text Corporation  
10074 Thomas Mueller, Day Software  
10075 John Newton, Alfresco Software  
10076 David Nuescheler, Day Software  
10077 Conleth O'Connell, Vignette Corporation  
10078 Marc Pallot, ESoCE-NET  
10079 Rainer Pausch, WeWebU Software AG  
10080 Dominique Pfister, Day Software  
10081 Peeter Piegaze, Day Software  
10082 David Pitfield, Oracle Corporation  
10083 Thomas Pole, Harris Corp  
10084 Norrie Quinn, EMC Corporation  
10085 Craig Randall, Adobe Corporation  
10086 Julian Reschke, Greenbytes GmbH  
10087 Celso Rodriguez, ASG Software Solutions  
10088 Steve Roth, Oracle Corporation  
10089 Patrick Ryan, IBM  
10090 Angela Schreiber, Day Software  
10091 Spencer Shearer, Exalead, Inc.  
10092 Madi Solomon, Pearson PLC  
10093 Wojciech Specht, fme AG  
10094 Dmitri Tcherevik, FatWire  
10095 Jason Tesser, dotCMS  
10096 David Torres, dotCMS  
10097 Maik Uhlenberg, fme AG  
10098 Oliver Walthard, Day Software  
10099 Patrick Ward, Booz Allen Hamilton

10100  
10101 Original Authors of the initial contribution:

10102 Al Brown, IBM  
10103 David Choy, EMC  
10104 Cornelia Davis, EMC  
10105 Ethan Gur-Esh, Microsoft

10106  
10107 Original Acknowledgements of the initial contribution:

10108 Al Brown, IBM  
10109 David Caruana, Alfresco  
10110 Derek Carr, IBM  
10111 David Choy, EMC  
10112 Cornelia Davis, EMC  
10113 Paul Goetz, SAP  
10114 Ethan Gur-Esh, Microsoft  
10115 Martin Hermes, SAP  
10116 Jens Hubel, OpenText  
10117 Jay Brown, IBM  
10118 Ryan McVeigh, Oracle  
10119 Gregory Melahn, IBM  
10120 Florian Müller, OpenText  
10121 John Newton, Alfresco  
10122 Norrie Quinn, EMC  
10123 Steve Roth, Oracle  
10124 Craig Randall, EMC

10125

---

## **B. Non-Normative Text**

10126

10127

---

## C. Revision History

10128

Revision	Date	Editor	Changes Made
1.0	01/11/2010	Al Brown	First specification

10129